

**SISTEM PENGHITUNG WAKTU UNTUK
KEJUARAAN RENANG BERBASIS
MIKROKONTROLER AT89C51**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar sarjana teknik



Disusun oleh:

**PUDYASTI PUSPITARINI
NIM. 0001063345-63**

**DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
M A L A N G
2005**

**SISTEM PENGHITUNG WAKTU UNTUK
KEJUARAAN RENANG BERBASIS
MIKROKONTROLER AT89C51**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar sarjana teknik



Disusun oleh:

**PUDYASTI PUSPITARINI
NIM. 0001063345-63**

DOSEN PEMBIMBING:

Panca mudjirahardjo, ST., MT
NIP. 132 288 163

Ir. Nurussa'adah
NIP. 131 994 339

**SISTEM PENGHITUNG WAKTU UNTUK
KEJUARAAN RENANG BERBASIS
MIKROKONTROLER AT89C51**

Disusun oleh:

**PUDYASTI PUSPITARINI
NIM. 0001063345 – 63**

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 23 Juli 2005

DOSEN PENGUJI

Ir. Bambang Siswojo
NIP. 131 759 588

Moch. Rif'an ST, MT
NIP. 132 283 659

Ir. Primantara HT
NIP. 132 090 390

Ir. M. Julius St., MS
NIP. 131 124 635

Mengetahui
Ketua Jurusan Teknik Elektro

Ir. Purwanto, MT
NIP 131 574 847

PENGANTAR

Assalamualaikum Wr. Wb.

Segala puja dan puji hanya kami haturkan kepada Allah S.W.T yang telah memberikan karunia-Nya, sehingga penulisan Tugas Akhir ini dapat diselesaikan dengan baik.

Terselesaikannya penyusunan skripsi ini tidak lepas dari bimbingan, dukungan dan bantuan berbagai pihak. Dalam kesempatan ini penyusun menyampaikan terima kasih kepada:

1. Ayahanda Margo Yuwono, Ibunda MG. Murtinem dan kakak Agustin yang telah banyak dan senantiasa memberikan doa yang tiada putus-putusnya, nasehat, materiil serta kesabaran dan ketelatenan dalam mendidiku agar menjadi anak yang berguna.
2. Bapak Ir. Purwanto, MT selaku ketua jurusan Teknik Elektro yang telah memberikan dosen pembimbing demi kelancaran penulisan Tugas Akhir ini.
3. Bapak Ir. Ponco Siwindarto, MS selaku KKDK Elektronika yang telah menyetujui dilaksanakannya penulisan Tugas Akhir ini.
4. Bapak Panca Mudjirahardjo, ST. MT selaku dosen pembimbing yang senantiasa memberikan koreksi, masukan serta bimbingan hingga Tugas Akhir ini terselesaikan.
5. Ibu Ir. Nurussa'adah selaku dosen pembimbing yang telah banyak membantu dengan memberikan saran dan bimbingan hingga terselesaikannya Tugas Akhir ini.
6. Seluruh dosen pengajar dan Staf Jurusan Teknik Elektro
7. Inchi dan Mas Syarif yang sangat membantu terselesaikannya Tugas Akhir ini.
8. Rekan-rekan Elektro UB angkatan 1999, 2000, 2001, dan teman-teman dari alih jenjang

Dengan menyadari adanya keterbatasan waktu, pengetahuan, pengalaman dan pustaka, maka skripsi ini masih jauh dari sempurna. Akhir kata Penulis berharap semoga skripsi ini dapat bermanfaat dan menjadi tambahan pengetahuan bagi siapapun yang membutuhkan.

Wassalamualaikum, Wr. Wb

Malang, 11 Juli 2005

Penulis,

Dedicate my thanks to...

My greatest God, Allah SWT...for your blessed all d time in my life. I adore and praise You with all my soul and heart

Bapak & Ibu tersayang...yang telah memberiku kehidupan yang sangat luar biasa, kelak akan kubalas semua perjuangan kalian memperebutkan gelar kesarjanaanku...

My beloved Inchi...yang menciptakan kedamaian hatiku. Kukagumi segala kesetiaan dan seluruh pengorbananmu untukku...you are precious GODs gift in my life.

My big angel, Agustin...you are d best family I have...teruslah maju menghadapi kejamnya hidup...SEMANGAT!

My best fella I ever met, Wq, Didin, Ika, Asih, Trisna, Arvi...kaulah teman yang paling sabar yang pernah kupunya, yang slalu setia menemani dan mendengarkan keluhanku...

For all of my friend, especially Elektro 00, terima kasih buat semuanya dan maaf atas perkataanku dan perbuatanku selama kita bersama...hope well always be together

DAFTAR ISI

	Halaman
PENGANTAR	i
PERSEMBAHAN	ii
DAFTAR ISI	iii
DAFTAR TABEL	v
DAFTAR GAMBAR	vi
DAFTAR LAMPIRAN	vii
ABSTRAK	viii
BAB I : PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Batasan Masalah	2
1.3. Rumusan Masalah	2
1.4. Tujuan	2
1.5. Sistematika Penulisan	
BAB II : TINJAUAN PUSTAKA	4
2.1. Mikrokontroler AT89C51	4
2.1.1. Organisasi Memori	7
2.1.1.1. Memori Program	8
2.1.1.2. Memori Data	9
2.1.2. Pewaktuan CPU	9
2.1.3. Interupsi	10
2.1.3.1. <i>Interrupt Enable</i>	11
2.1.3.2. <i>Interrupt Priority</i>	12
2.1.4. <i>Special Function Register</i>	13
2.2. Basis Pewaktuan (<i>Real Time Clock</i>) IC DSI12887.....	15
2.3. Rangkaian Indikasi Suara (ISD)1400.....	18
2.4. LCD (<i>Liquid Crystal Display</i>) M1632	21
2.5. LED (<i>Light Emitting Diode</i>).....	22
2.6. DM74LS573 (<i>Latch</i> dengan Tiga Output)	22
2.7. Peraturan Kejuaraan Renang	23
2.7.1. Peraturan <i>Start</i>	23
2.7.2. Peraturan <i>Finish</i>	24
2.7.3. Pewaktuan	24
BAB III : METODOLOGI	26
3.1. Studi Literatur	26
3.2. Perancangan Peralatan	26
3.3. Perancangan Alat Secara Matematis.....	26
3.4. Pembuatan Alat	27
3.5. Pengujian Alat.....	27
BAB IV : PERANCANGAN ALAT	29
4.1. Spesifikasi Alat	29
4.2. Diagram Blok Rangkaian.....	29

4.3.	Prinsip Kerja Alat.....	30
4.4.	Perancangan Perangkat Keras	31
4.4.1.	Antarmuka Mikrokontroler AT89C51	31
4.4.2.	Antarmuka RTC (<i>Real Time Clock</i>) IC DSI12887	33
4.4.3.	Antarmuka IC ISD1420	34
4.4.4.	Antarmuka LCD (<i>Liquid Crystal Display</i>)	36
4.4.5.	Switch sebagai Penanda <i>Start</i> dan <i>Finish</i>	38
4.5.	Perancangan Perangkat Lunak	39
1)	BAB V : HASIL DAN PEMBAHASAN	41
5.1.	Pengujian Mikrokontroler AT89C51- LCD (<i>Liquid Crystal Display</i>)	41
5.2.	Pengujian RTC (<i>Real Time Clock</i>).....	42
5.3.	Pengujian Rangkaian Perekam/Pemutar Suara ISD1420.....	43
5.4.	Pengujian Saklar <i>Start</i> (<i>Switch</i>)	45
5.5.	Pengujian Sensor Sentuh.....	46
5.6.	Pengujian Keseluruhan Sistem.....	47
	BAB VI : PENUTUP	49
6.1	Kesimpulan	49
6.2	Saran.....	49

DAFTAR PUSTAKA

LAMPIRAN

Lampiran I	: Rangkaian Lengkap.....	L-1.1
	<i>Layout</i> PCB.....	L-1.2
Lampiran II	: Foto Alat	L-2.1
	Mekanik Alat.....	L-2.2
Lampiran III	: <i>Listing</i> Program.....	L-3
Lampiran IV	: Lembar Data Mikrokontroler AT89C51.....	L-4.1
	Lembar Data DS12887.....	L-4.2
	Lembar Data ISD 1400.....	L-4.3
	FINA Swimming Timing Rule.....	L-4.4
	LED (<i>Light Emitting Diode</i>).....	L-4.4

DAFTAR TABEL

	Halaman
Tabel 2.1. Alamat Layanan Rutin Interupsi.....	10
Tabel 2.2. <i>Interrupt Enable</i>	11
Tabel 2.3. <i>Interrupt Priority</i>	12
Tabel 2.4. <i>Special Function Register</i>	14
Tabel 2.5. Tabel Kebenaran IC <i>Latch</i> 74LS573	23
Tabel 4.1. Tabel Pemilihan Alamat Perekaman Suara.....	36
Tabel 5.1. Data Hasil Pengujian Rangkaian <i>Start</i>	46
Tabel 5.2. Data Hasil Pengujian Rangkaian <i>Finish</i>	47
Tabel 5.3. Tampilan Waktu Renang pada LCD.....	48

DAFTAR GAMBAR

	Halaman
Gambar 2.1. Blok Diagram MCS-51	4
Gambar 2.2. Susunan Kaki MCS-51	5
Gambar 2.3. Struktur Memori Mikrokontroler MCS-51	7
Gambar 2.4. Memori Program	8
Gambar 2.5. Memori Data Internal	9
Gambar 2.6. Rangkaian Osilator Internal	9
Gambar 2.7. Konfigurasi Penyemat IC DSI12887.....	15
Gambar 2.8. Peta Alamat RAM Internal <i>Real Time Clock</i> DSI12887.....	18
Gambar 2.9. Konfigurasi Penyemat IC ISD1400	18
Gambar 2.10. LCD (<i>Liquid Crystal Display</i>) dan Konfigurasi Pinnya.....	21
Gambar 2.11. LED (<i>Light Emitting Diode</i>)	22
Gambar 4.1. Blok Diagram Sistem.....	30
Gambar 4.2. Rangkaian Antarmuka Mikrokontroler AT89C51	31
Gambar 4.3. Antarmuka IC DS12887 dengan Mikrokontroler.....	33
Gambar 4.4. Antarmuka ISD1420 dengan Mikrokontroler	35
Gambar 4.5. Antarmuka LCD	37
Gambar 4.6. Switch <i>Start</i> dan <i>Finish</i>	38
Gambar 4.7. <i>Flowchart</i> Program	40
Gambar 5.1. Bagan Pengujian Mikrokontroler dan LCD	41
Gambar 5.2. <i>Flowchart</i> Sistem Pengujian Mikrokontroler dan LCD	42
Gambar 5.3. Tampilan pada LCD dari Hasil Pengujian	42
Gambar 5.4. Bagan Pengujian RTC.....	42
Gambar 5.5. <i>Flowchart</i> Sistem Pengujian RTC.....	43
Gambar 5.6. Tampilan pada LCD dari Hasil Pengujian RTC.....	43
Gambar 5.7. <i>Flowchart</i> Sistem Pengujian ISD1420	44
Gambar 5.8. Bagan Perekaman Suara pada IC ISD1420.....	44
Gambar 5.9. Bagan Pemutaran Ulang Rekaman Suara pada IC ISD1420.....	45
Gambar 5.10. Rangkaian <i>Start</i>	46
Gambar 5.11. Rangkaian Sensor Sentuh.....	47
Gambar 5.12. Bagan Pengujian Keseluruhan Sistem.....	48
Gambar L-1a Rangkaian Lengkap.....	L-1a
Gambar L-1b <i>Layout</i> PCB	L-1b
Gambar L-2a Foto Alat Terbuka Tampak Atas	L-2a
Gambar L-2b Foto Alat Tertutup Tampak Atas.....	L-2b
Gambar L-2c Mekanik Alat	L-2c

DAFTAR LAMPIRAN

	Halaman
Lampiran I : Rangkaian Lengkap.....	L1
<i>Layout</i> PCB.....	L1
Lampiran II : Foto Alat.....	L2
Mekanik Alat.....	L2
Lampiran III : <i>Listing</i> Program.....	L3
Lampiran IV : Lembar Data Mikrokontroler AT89C51.....	L-4.1
Lembar Data DS12887.....	L-4.2
Lembar Data ISD 1400.....	L-4.3
FINA <i>Swimming Timing Rule</i>	L-4.4
LED (<i>Light Emitting Diode</i>).....	L-4.4

ABSTRAK

Pudyasti Puspitarini, NIM.0001063345-63, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya Malang, 2005. “ *Sistem Penghitung Waktu untuk Kejuaraan Renang Berbasis Mikrokontroler AT89C51*”. Dosen pembimbing: 1. Panca Mudjirahardjo, ST. MT 2. Ir. Nurussa'adah.

Dalam pembuatan sistem penghitung waktu untuk kejuaraan renang ini ini utamanya adalah untuk menghitung waktu yang ditempuh oleh perenang dalam melaksanakan suatu perlombaan secara otomatis dan bersamaan. Pengontrol utama dari sistem ini menggunakan mikrokontroler AT89C51, yang dilengkapi dengan antarmuka dengan *Real Time Clock* sebagai basis pewaktuan. Hasil dari kejuaraan renang yang berupa waktu itu ditampilkan dalam suatu LCD (*Liquid Crystal Display*). Rangkaian perekam dan pemutar suara diaktifkan sebagai media informasi saat persiapan dan penanda ketika para perenang harus sudah meloncat ke air. Untuk mengetahui kondisi pada saat perenang harus melakukan persiapan di *platform* (LED merah nyala), perenang sedang melakukan putaran renang (LED hijau nyala) dan setelah menyentuh *finish* (sensor sentuh) adalah matinya LED hijau.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada saat ini dunia elektronik berkembang dengan pesat. Banyak peralatan diciptakan dengan perangkat elektronik sebagai pendukungnya. Baik dalam kehidupan sehari-hari maupun dalam dunia usaha. Dalam kehidupan kita, banyak peralatan mekanik yang beralih ke perangkat elektronik, misalnya jam penunjuk waktu. kebanyakan terdapat dalam bentuk mekanik. Dalam alat tersebut hanya mengandalkan perangkat mekanik saja. Sekarang sudah banyak terdapat jam penunjuk waktu yang dibuat dari perangkat elektronik secara digital.

Penggunaan jam penghitung waktu pada perlombaan renang masih jarang digunakan. Alat pencatat waktu saat ini kebanyakan masih menggunakan *stopwatch*, dimana kita hanya bisa mengoperasikan perangkat *stopwatch* tersebut secara manual. Hal ini sangat tidak efisien dan tidak akan mampu digunakan bila kita harus mencatat waktu dan menghitung kecepatan beberapa peserta renang, kecuali kita menggunakan beberapa *stopwatch* yang tentu saja membutuhkan beberapa orang untuk mengoperasikan secara manual. Hal inipun masih kurang efisien dalam pengoperasian, karena kita masih akan membandingkan waktu beberapa perenang untuk menentukan siapa yang paling cepat. Sistem Penghitung Waktu untuk Kejuaraan Renang Berbasis Mikrokontroler AT89C51 ini tidak menggunakan pemancar maupun penerima frekuensi. Alat ini tidak tergantung dengan kondisi lingkungan yang ada, karena pemancaran dan penerimaan frekuensi sangat tergantung dengan kondisi alam saat itu.

Melihat kondisi tersebut, maka perlu diciptakan sebuah perangkat elektronik yang bisa mengatasi kondisi tersebut. Dengan adanya alat ini maka waktu yang dihasilkan oleh beberapa perenang dalam satu kali lintasan bisa diketahui bersamaan sekaligus. Dalam perancangan alat ini kita akan membahas tentang perencanaan dan pembuatan sebuah alat pencatat waktu perlombaan renang yang diharapkan alat ini nanti bisa kita pakai dalam berbagai perlombaan renang.

Pembuatan SISTEM PENGHITUNG WAKTU UNTUK KEJUARAAN RENANG BERBASIS MIKROKONTROLER AT89C51 ini diharapkan dapat menjadi salah satu solusi untuk aplikasi di bidang olah raga dengan memadukan olah raga dengan teknologi.

1.2 Batasan Masalah

Dengan mengacu pada permasalahan yang telah dirumuskan, maka hal-hal yang berkaitan dengan permasalahan tersebut dibatasi sebagai berikut:

1. Sistem dibuat untuk penghitung waktu pada kejuaraan renang.
2. Tidak membahas catu daya.
3. Jumlah jalur adalah delapan jalur lintasan dan untuk delapan perenang.
4. Minimum penghitungan adalah dalam orde 1/100 detik.
5. Kelas perlombaan renang adalah 50m, 100m dan 200m.
6. Sistem dibuat dalam bentuk miniatur.

1.3 Rumusan Masalah

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

1. Bagaimana merancang dan membuat suatu sistem elektronika yang dapat digunakan untuk menghitung waktu pada suatu kejuaraan renang.
2. Bagaimana sistem antarmuka antara mikrokontroler dengan RTC (*Real Time Clock*), mikrokontroler dengan ISD (*Information Storage Devices*) dan mikrokontroler dengan LCD (*Liquid Crystal Display*).
3. Bagaimana mengolah data yang diterima dari sensor sentuh untuk ditampilkan sebagai data waktu di dalam LCD (*Liquid Crystal Display*).

1.4 Tujuan

Tujuan dari perancangan dan pembuatan alat ini adalah merancang dan membuat suatu alat yang digunakan untuk menghitung waktu yang dibutuhkan dalam perlombaan renang.

1.5 Sistematika Penulisan

Sistematika pembahasan dalam Laporan Tugas Akhir ini adalah sebagai berikut:

- **BAB I :** Menjelaskan pendahuluan yang berisi latar belakang, tujuan, batasan masalah dan sistematika pembahasan dari alat yang direncanakan.
- **BAB II :** Menjelaskan teori penunjang alat yang direncanakan antara lain mengenai metode penghitungan waktu, Mikrokontroler AT89C51, sensor sentuh, ISD (*Information Storage Devices*), LCD (*Liquid Crystal Display*), RTC (*Real Time Clock*) dan rangkaian penunjang sistem.
- **BAB III :** Menjelaskan tentang metodologi dalam perancangan alat.
- **BAB IV :** Menjelaskan tentang diagram blok, prinsip kerja rangkaian dan rancangan alat.
- **BAB V :** PENGUJIAN DAN ANALISIS, berisi pengujian yang dilakukan dalam perencanaan alat, dan data hasil pengujian dan analisis.
- **BAB VI :** Menjelaskan kesimpulan yang dapat diambil dan saran untuk penyempurnaan alat.
- **Lampiran:** Berisi tentang gambar rangkaian dan data komponen

BAB II

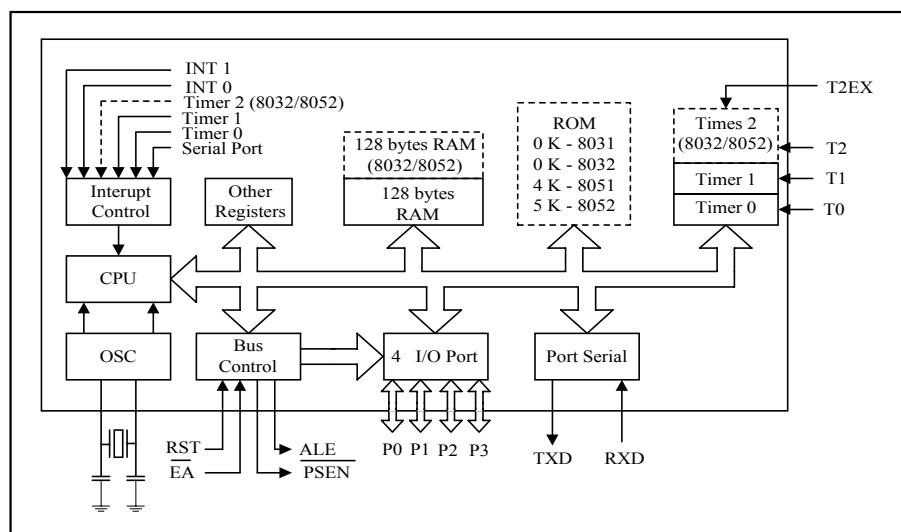
TINJAUAN PUSTAKA

Sebagai dasar perencanaan dan pembuatan Sistem Penghitung Waktu untuk Kejuaraan Renang Mikrokontroler AT89C51, diperlukan teori-teori yang berhubungan dengan aplikasi dan perencanaan rangkaian pendukung pada alat tersebut. Teori-teori pendukung yang akan dibahas pada bab ini adalah:

1. Mikrokontroler AT89C51
2. RTC (*Real Time Clock*)
3. ISD (*Information Storage Devices*)
4. LCD (*Liquid Crystal Display*)
5. LED (*Light emitting diode*)
6. DM74LS573 (Latch dengan tiga output)

2.1 Mikrokontroler AT89C51

AT89C51 adalah mikrokontroler yang mempunyai kapabilitas instruksi dan konfigurasi pin dengan mikrokontroler MCS-51. Blok diagram MCS-51 ditunjukkan dalam Gambar 2.1.

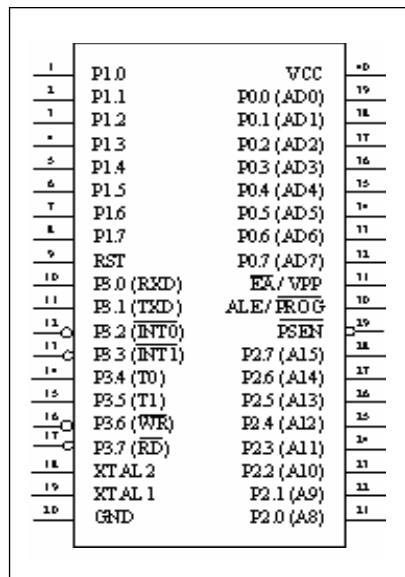


Gambar 2.1 Blok Diagram MCS-51

Sumber: Atmel AT89C51 *Data Sheets*, 1997: 2

AT89C51 mempunyai 4 kilobyte EEPROM (*Electrical Erasable and Programmable Read Only Memory*), 128 byte RAM (*Random Acces Memory*), 32 pin I/O (4 buah port I/O 8 bit) dan tiap pin dapat diprogram secara paralel dan tersendiri, mempunyai dua buah *timer counter* 16 bit, dan enam buah sumber interupsi.

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



Gambar 2.2 Susunan Kaki MCS-51

Sumber: Atmel AT89C51 *Data Sheets*, 1997: 1

Fungsi kaki-kaki AT89C51 adalah:

- Port 1 (Pin 1...8), berfungsi sebagai port I/O biasa.
- Pin 9 (RST), pulsa transisi dari rendah ke tinggi yang diumpankan ke pin RST akan mereset AT89C51. Pin ini dihubungkan dengan rangkaian *power on reset*.
- Port 3 (Pin 10...17), port paralel 8 bit dua arah yang memiliki fungsi pengganti. Fungsi pengganti meliputi TXD (*Transmit Data*) yang

digunakan sebagai port serial *output*, RXD (*Receive Data*) yang digunakan sebagai port serial *input*, INT0 (*Interrupt 0*) yang digunakan sebagai port eksternal *interrupt 0*, INT1 (*Interrupt 1*) yang digunakan sebagai port port eksternal *interrupt 1*, T0 (*Timer 0*) yang digunakan sebagai port eksternal *timer 0 input*, T1 (*Timer 1*) yang digunakan sebagai port eksternal *timer 1 input*, WR (*Write*) yang digunakan sebagai eksternal data *memory write strobe*, dan RD (*Read*) yang digunakan sebagai eksternal data *memory read strobe*. Apabila fungsi pengganti tidak digunakan, pin-pin ini dapat digunakan sebagai port I/O biasa.

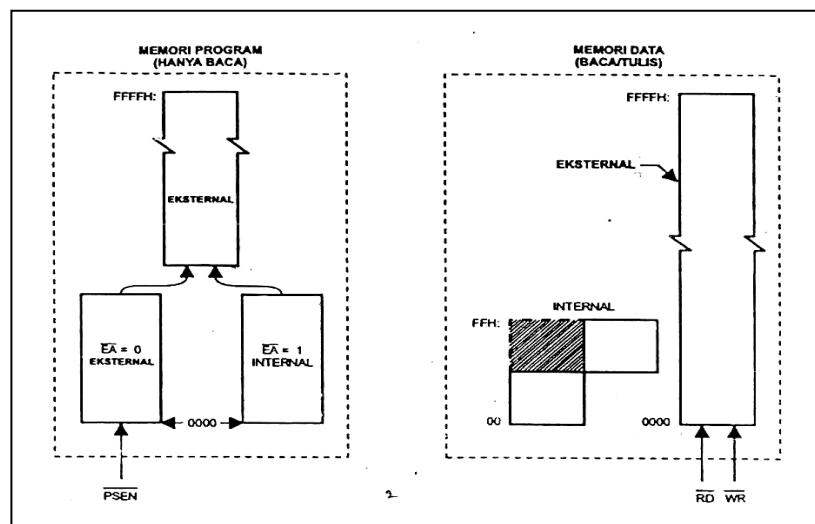
- Pin 18 (XTAL1), merupakan pin masukan ke rangkaian osilator internal. Osilator kristal dan sumber osilator luar dapat digunakan.
- Pin 19 (XTAL2), merupakan pin masukan ke rangkaian osilator internal. Pin ini dipakai apabila menggunakan osilator kristal.
- Pin 20 (*Ground*), dihubungkan ke VSS atau *ground*.
- Port 2 (Pin21...28), port paralel 8 bit dua arah, dapat digunakan sebagai port I/O 8 bit biasa, dan digunakan untuk mengirim *byte* alamat bila digunakan untuk mengakses memori eksternal.
- Pin 29 (PSEN/*Program Store Enable*), merupakan pengontrol yang digunakan untuk mengakses program memori eksternal masuk ke dalam bus selama proses pemberian/pengambilan instruksi.
- Pin 30 (ALE/*Address Latch Enable*), digunakan untuk menahan alamat memori eksternal selama pelaksanaan instruksi.
- Port 31 (EA/*External Access*), bila pin yang diberikan logika tinggi, maka mikrokontroler akan melaksanakan instruksi dari memori program luar.
- Port 0 (Pin 32...39), merupakan port paralel 8 bit *open drain* dua arah. Port 0 dapat digunakan sebagai port I/O biasa dan dapat juga digunakan untuk memultipleks alamat dengan data pada waktu mengakses memori eksternal.
- Pin 40 (VCC), dihubungkan dengan VCC (+5 volt).

2.1.1 Organisasi Memori

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

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

Sinyal yang memungkinkan pembacaan dari memori program eksternal adalah dari pin \overline{PSEN} (*Program Store Enable*). Memori data terletak pada ruang alamat terpisah dari memori program. RAM (*Random Acces Memory*) eksternal 64k dapat dialamati dalam ruang memori data eksternal. CPU (*Central Processing Unit*) menghasilkan sinyal *read* atau *write* selama menghubungkan memori data eksternal. Gambar pembagian ruang untuk program dan data pada mikrokontroler MCS_51 ditunjukkan dalam Gambar 2.3



Gambar 2.3 Struktur Memori Mikrokontroler MCS-51

Sumber: Agfianto Eko Putra, 2002 :3

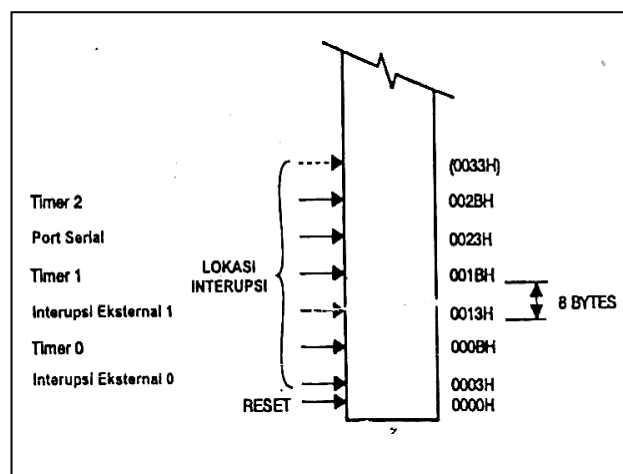
Mikrokontroler AT89C51 memiliki 5 buah ruang alamat, yaitu:

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

2.1.1.1 Memori Program

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

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

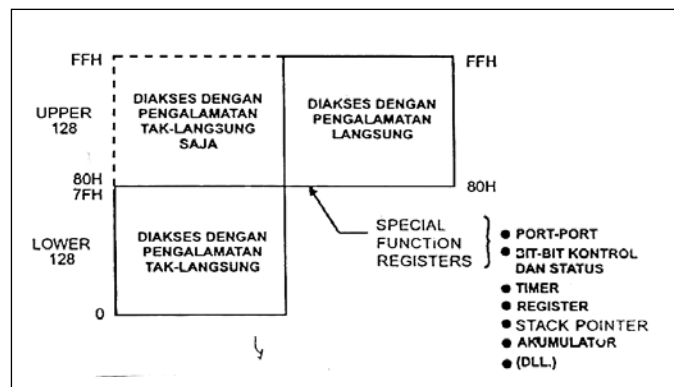


Gambar 2.4 Memori Program

Sumber: Agfianto Eko Putra, 2002 :4

2.1.1.2 Memori Data

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

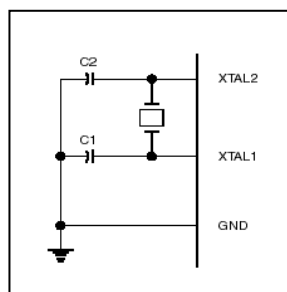


Gambar 2.5 Memori Data Internal

Sumber: Agfianto Eko Putra, 2002: 6.

2.1.2 Pewaktuan CPU

Mikrokontroler AT89C51 memiliki osilator internal (*on chip oscillator*) yang dapat digunakan sebagai sumber *clock* bagi CPU. Untuk menggunakan osilator internal diperlukan sebuah kristal antara pena XTAL1 dan XTAL2 serta dua buah kapasitor ke *ground* seperti ditunjukkan dalam Gambar 2.6. Untuk kristalnya dapat digunakan frekuensi dari 6 sampai 12MHz, sedangkan untuk kapasitor dapat bernilai antara 27pF sampai 33pF.



Gambar 2.6 Rangkaian Osilator Internal

Sumber: Agfianto Eko Putra, 2002 :22

2.1.3 Interupsi

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

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

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

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

Tabel 2.1 Alamat layanan rutin interupsi

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

Sumber: Agfianto Eko Putra, 2002: 157

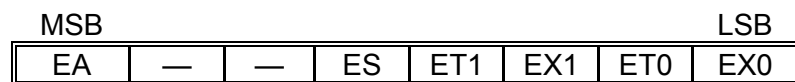
Mikrokontroler AT89C51 menyediakan 5 sumber interupsi: 2 interupsi eksternal, 2 interupsi *timer*, dan satu interupsi port serial. Interupsi eksternal \overline{INT}_0 dan \overline{INT}_1 masing-masing dapat diaktifkan berdasarkan level atau transisi, tergantung bit IT0 dan IT1 dalam TCON. Flag yang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

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

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

2.1.3.1 *Interrupt Enable*

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



Tabel 2.2 *Interrupt Enable*

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

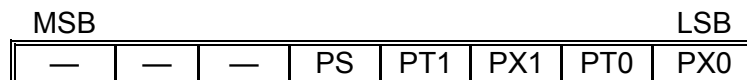
Misalnya jika akan mengaktifkan interupsi 0 ($\overline{\text{INT}}_0$), logika 1 diberikan pada EA dan EX0, sehingga keduanya menjadi aktif dan mengaktifkan $\overline{\text{INT}}_0$.

2.1.3.2 Interrupt Priority

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

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

Bit-bit pada IP adalah sebagai berikut:



Tabel 2.3 *Interrupt Priority*

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

2.1.4 *Special Function Register*

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

Beberapa fungsi SFR yang penting yaitu:

- *Accumulator* (ACC) merupakan register yang penting dalam operasi penambahan dan pengurangan,
- Register B merupakan register khusus yang berfungsi melayani operasi perkalian dan operasi pembagian.
- *Program Status Word* (PSW) berisi beberapa bit status yang menggambarkan keadaan CPU pada saat itu. PSW terdiri dari bit *carry*, *auxiliary carry*, dua bit pemilih bank, bendera *overflow*, bit paritas dan dua bendera yang dapat didefinisikan sendiri oleh pemakai.
- *Stack Pointer* (SP) merupakan register 8 bit. Register SP digunakan dalam proses simpan dan ambil dari atau ke *stack*, dan dapat diletakkan pada alamat manapun pada RAM internal. Isi register ini ditambah sebelum data disimpan, menggunakan instruksi *PUSH* dan *CALL*. Walaupun *stack* bisa menempati lokasi dimana saja dalam RAM (*Random Acces Memory*), namun setelah adanya *reset*, register SP akan selalu diinisialisasikan pada alamat 07h, sehingga *stack* akan dimulai pada lokasi 08h. Sedangkan untuk pengurangan isi register SP menggunakan instruksi *POP*, yaitu untuk mentransfer data dari areal *stack* yang ditunjukkan oleh *Stack Pointer* ke operand tujuan sehingga menyebabkan *Stack Pointer* berkurang. Selanjutnya instruksi *RET* akan mengembalikan proses pelaksanaan program ke program utama dengan cara mengambil isi *stack* yang disimpan pada saat instruksi *CALL*, dan selanjutnya dimasukkan ke dalam PC (*Program Counter*).

Tabel 2.4 *Special Function Register*

Simbol	Nama	Alamat	Power-on Reset
ACC	Akumulator	E0h	0000 0000
B	Register B	F0h	0000 0000
PSW	Program Status Word	D0h	0000 0000
SP	Stack Pointer	81h	0000 0111
DPTR	Data Pointer 16 bit		
DPL	Data Pointer byte rendah	82h	0000 0000
DPH	Data Pointer byte tinggi	83h	0000 0000
P0	Port 0	80h	1111 1111
P1	Port 1	90h	1111 1111
P2	Port 2	A0h	1111 1111
P3	Port 3	B0h	1111 1111
IP	Interrupt Priority Control	B8h	xxx0 0000
IE	Interrupt Enable Control	A8h	0xx0 0000
TMOD	Timer/counter Mode Control	89h	0000 0000
TCON	Timer/counter Control	88h	0000 0000
TH0	Timer/counter 0 high byte	8Ch	0000 0000
TL0	Timer/counter 0 low byte	8Ah	0000 0000
TH1	Timer/counter 1 high byte	8Dh	0000 0000
TL1	Timer/counter 0 low byte	8Bh	0000 0000
SCON	Serial Control	98h	0000 0000
SBUF	Serial Data Buffer	99h	xxxx xxxx
PCON	Power Control	87h	0xxx xxxx

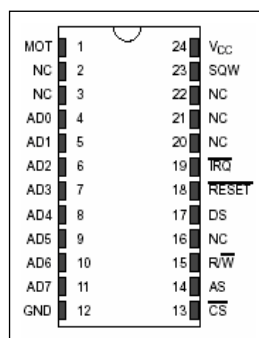
Sumber: Agfianto Eko Putra, 2002 : 189

- *Data pointer* (DTPR) terdiri dari dua register, yaitu untuk *byte* rendah (*Data Pointer Low/DPL*) dan untuk *byte* tinggi (*Data Pointer High/DPH*). Fungsinya adalah untuk menahan alamat 16 bit. DPTR dapat dimanipulasi sebagai register 16 bit atau sebagai dua buah register 8 bit.

- Port 0 sampai port 3 merupakan register yang berfungsi untuk membaca dan mengeluarkan data pada port 0, 1, 2, dan 3. Masing-masing register ini dapat dialamti secara per-bit maupun per-byte.
- *Control register*, terdiri atas register yang mempunyai fungsi kontrol untuk mengontrol sistem interupsi, terdapat dua register khusus, yaitu register IP (*Interrupt Priority*) dan register IE (*Interrupt Enable*). Untuk mengontrol pelayanan *timer/control* terdapat register khusus yaitu register TMOD (*Timer/Counter Mode Control*) dan register TCON (*Timer/counter Control*), serta untuk pelayanan port serial menggunakan register SCON (*serial port control*).

2.2 Basis Pewaktuan (*Real Time Clock*) IC DS12887

DS12887 adalah serpih yang dipakai sebagai basis pewaktuan buatan Dallas Semiconductor. Terdiri atas 24 penyemat dengan konfigurasi penyemat seperti ditunjukkan dalam Gambar 2.7



Gambar 2.7 Konfigurasi Penyemat IC DS12887

Sumber: Atmel, 1997: 4-29

Keistimewaan IC DS12887 antara lain adalah:

- Osilator internal dan *time base* internal.
- Menghitung detik, menit, jam dalam sehari-hari.
- Menghitung hari dalam setiap minggu, tanggal, bulan dan tahun.
- Seratus tahun kalender.
- Mempunyai catu daya *back-up*.

- Dapat tetap beroperasi selama lebih dari 10 tahun tanpa kehadiran catu daya eksternal (karena adanya baterai cadangan di dalam RTC).
- RAM internal sebesar 64 byte, terdiri atas 14 byte untuk *clock* dan register kontrol, sedangkan 50 byte lainnya dapat digunakan oleh pemakai.

Secara keseluruhan, fungsi penyemat-penyemat RTC DS12887 adalah sebagai berikut:

GND dan Vcc:

Merupakan penyemat catu daya. Vcc dihubungkan dengan catu daya +5 volt dan ground.

MOT — *Motel*:

Digunakan sebagai saklar pemilih mode diagram pewaktuan. Apabila dihubungkan dengan Vcc berarti sistem diagram pewaktuan Motorola yang dipakai, jika dihubungkan dengan GND berarti sistem diagram pewaktuan Intel yang dipakai.

AS — *Address Strobe Input*:

Merupakan masukan bagi sinyal yang digunakan untuk memisahkan bus data dan bus alamat (ALE). Tepi turun AS/ALE akan menyebabkan alamat ditahan dalam DS1287. Tepi naik berikutnya yang terjadi AS akan meniadakan alamat tersebut tanpa memperhatikan apakah penyemat \overline{CS} aktif atau tidak.

AD0–AD7 — *Multiplexed Bidirectional Address/Data Bus*:

Merupakan bus alamat/data dua arah yang termultipleks. Pengiriman data maupun alamat maupun ke RTC dilakukan melalui bus ini.

DS — *Data Strobe or Read Input*:

Penyemat DS dapat diartikan sama seperti sinyal *Output Enable* (\overline{OE}) dalam komponen memori. Dihubungkan dengan sinyal \overline{RD} yang berasal dari mikrokontroler untuk melakukan proses membaca data RAM internal RTC.

$\overline{R/W}$ — *Read/Write Input*:

Penyemat $\overline{R/W}$ dapat diartikan sama seperti sinyal *Write Enable* (\overline{WE}) yang biasa dipakai dalam komponen memori jenis RAM. Digunakan untuk proses penulisan data RAM internal RTC.

SQW — Square Wave Output:

Merupakan penyemat yang dapat dipilih untuk mengeluarkan satu dari 13 frekuensi keluaran yang tersedia. Besar frekuensi keluaran SQW dapat diubah dengan diprogram dalam Register A. Untuk mengaktifkan atau menonaktifkan sinyal SQW dipilih lewat bit SQWE dalam Register B.

 $\overline{\text{CS}}$ — Chip Select Input:

Merupakan masukan untuk mengaktifkan periferan RTC. Sinyal $\overline{\text{CS}}$ didapat dari dekoder alamat dengan alamat tertentu.

 $\overline{\text{IRQ}}$ — Interrupt Request Output:

Sinyal $\overline{\text{IRQ}}$ merupakan sinyal aktif rendah, yang dapat digunakan untuk menginterupsi mikrokontroler. Keluaran $\overline{\text{IRQ}}$ tetap rendah selama status bit yang menyebabkan interupsi ada. Untuk me-reset $\overline{\text{IRQ}}$, mikrokontroler memberikan program register C RTC. Saat tidak terdapat interupsi, penyemat ini dalam kondisi impedansi tinggi (*high-impedance*).

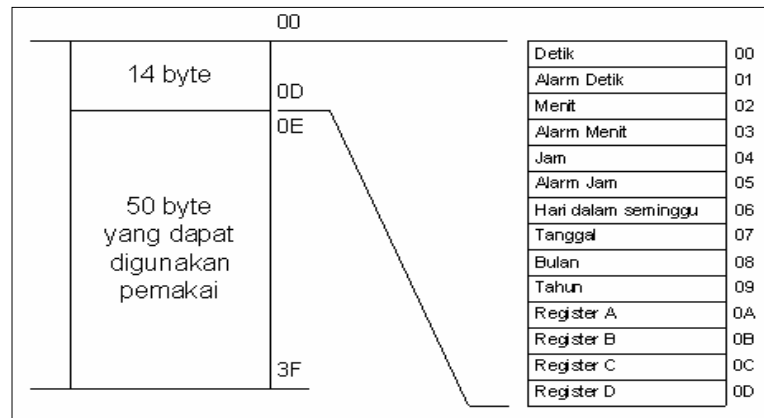
 $\overline{\text{RESET}}$ — Reset Input:

Sinyal $\overline{\text{RESET}}$ diberikan dengan memberikan logika rendah selama waktu yang dispesifikasikan, dan tidak berpengaruh dalam unjuk kerja *clock*, kalender, dan fungsi RAM. Namun sinyal $\overline{\text{RESET}}$ dapat mengakibatkan beberapa *flag* di-reset menjadi nol.

Untuk menginisialisasi RTC, yaitu men-set waktu yang ada di RAM RTC saat kali pertama RTC diaktifkan, digunakan sinyal $\overline{\text{WR}}$ (*write*) yang dihubungkan dengan penyemat $\overline{\text{R/W}}$, data yang diinginkan dalam alamat yang bersesuaian. Sedangkan untuk membaca data RAM internal RTC digunakan sinyal $\overline{\text{RD}}$ (*read*) yang dihubungkan dengan penyemat DS (*data strobe*). Operasi pembacaan dan penulisan RAM internal RTC sama seperti operasi baca dan tulis komponen memori jenis RAM.

RTC mempunyai RAM internal sebesar 64 byte, yang berisi data-data mengenai waktu yang sedang berjalan, seperti: detik, menit, jam, hari, tanggal, bulan, tahun, serta beberapa register. Secara otomatis RTC akan mengganti data dalam RAM internal sesuai dengan perhitungannya. Jika diinginkan mengambil

data waktu, maka dibaca dalam RAM internal sesuai dengan alamat yang dimaksud. Peta alamat RAM internal RTC DS12887 ditunjukkan dalam Gambar 2.8

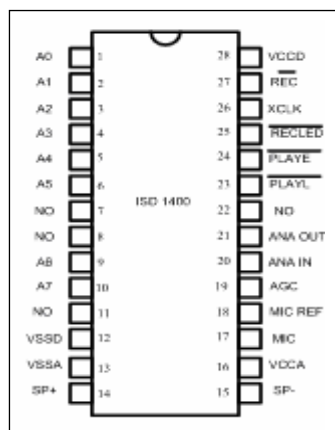


Gambar 2.8 Peta Alamat RAM Internal *Real Time Clock* DS12887

Sumber: *Dallas Semiconductor Data Sheets* 1995: 5

2.3 Rangkaian Indikasi Suara (ISD)1400

Media penyimpanan suara sebagai indikasi operasi sistem dapat menggunakan IC ISD. IC ISD 1400 ini tidak menggunakan catu daya eksternal, meskipun demikian rekaman suara yang tersimpan di dalamnya tidak akan hilang. Konfigurasi penyemat IC ISD1400 ditunjukkan dalam Gambar 2.9



Gambar 2.9 Konfigurasi Penyemat IC ISD1400

Sumber: *Data Sheet ISD 1400*, 1993: 2

Penjelasan dari masing-masing kaki adalah sebagai berikut:

- *Voltage Input* (V_{CCA} , V_{CCD}), untuk mengurangi *noise*, rangkaian analog dan digital dalam IC ISD 1400 digunakan *power supply* tersendiri dalam pak.
- *Ground Input* (V_{SSA} , V_{SSD}), pin ini harus dihubungkan dalam *power supply ground* dengan impedansi rendah.
- *Playback Edge Activated* (\overline{PLAYE}), ketika tidak sedang digunakan untuk merekam, atau menyalakan, pin \overline{PLAYE} harus diberi logika tinggi.
- *Playback Level Activated* (\overline{PLAYL}), pin ini harus diberi logika rendah untuk menjalankan semua operasi perekaman atau *playback*. Alamat *input* dan *record* (\overline{REC}) ditahan dengan adanya transisi turun dari \overline{PLAYL} .
- *Record* (\overline{REC}), input dari pin ini ditahan dengan adanya transisi turun dari dari pin \overline{PLAYL} . Logika tinggi akan memilih *playback cycle* dan logika rendah untuk memilih *record cycle*. Untuk *record cycle*, alamat *input* menetapkan mulainya alamat dan merekam secara kontinyu sampai \overline{PLAYE} atau \overline{PLAYL} ke transisi tinggi atau adanya *overflow*. Ketika *record cycle* berhenti dengan adanya transisi tinggi dari \overline{PLAYL} atau \overline{PLAYE} , sinyal *record led output* (\overline{RECLEd}) disimpan di alamat terakhir di memori. Untuk *playback cycle*, alamat *input* menetapkan mulainya alamat dan akan memutar suara perekaman sampai alamat \overline{RECLEd} ditemukan.
- *Reclad Led Output* (\overline{RECLEd}), sebuah alamat akan dimasukkan secara otomatis di akhir masing-masing perekaman. Pulsa *output* \overline{RECLEd} akan rendah untuk setiap periode dari T_{EOM} diakhir masing-masing perekaman.
- *Microphone Input* (MIC), digunakan untuk mentransfer sinyal suara ke *on chip preamplifier*.

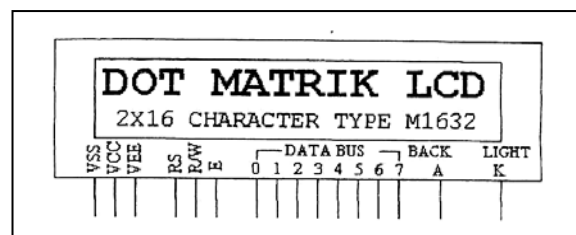
- *Microphone Reference Input (MIC REF)*, adalah *input* pembalik ke *microphone preamplifier*. Pin ini untuk memberikan *noise cancelling* atau *common mode rejection input* ke IC ini ketika dihubungkan ke sebuah *microphone* diferensial.
- *Automatic Gain Control Input (AGC)*, pin ini secara dinamik mengubah penguatan dari *preamplifier* untuk mengimbangi dari lebar jarak dari level *microphone input*. AGC memberikan jarak secara penuh dari suara rendah ke tinggi untuk direkam dengan distorsi minimal.
- *Analog output (ANA OUT)*, pin ini memberikan *preamplifier output* ke pengguna. Tegangan penguatan dari *preamplifier* ditentukan oleh level tegangan dari pin AGC.
- *Analog input (ANA IN)*, pin ini akan mentransfer sinyal ke dalam chip untuk perekaman. Untuk *microphone input*, pin ANA OUT harus dihubungkan ke kapasitor eksternal ke pin ANA IN. Jika permintaan *input* diperoleh dari sumber lain dari *microphone*, sinyal ini dapat langsung dikopel oleh kapasitor ke pin ANA IN.
- *External Clock Input (XCLK)*, untuk ISD1400 mempunyai sebuah internal *pull-down*. Jika *power supply* yang digunakan mempunyai presisi tinggi, maka alat ini dapat langsung di-*clock* pada pin XCLK. Besarnya frekuensi tiap *clock* yang masuk ke dalam IC dipakai hanya sebesar setengahnya dari besar frekuensi aslinya. Jika pin XCLK tidak digunakan, harus dihubungkan ke *ground*.
- *Speaker output (SP+/SP-)*, ISD 1400 telah mempunyai sebuah *driver on-chip* diferensial speaker, sanggup memikul beban 50 mW dalam 16 ohm dari AUX IN. *Speaker output* berada pada level V_{SSA} selama proses *record* dan *power down*.
- Peralatan penyimpanan suara, ISD1400 telah mempunyai kualitas bagus, dengan durasi penyimpanan dari 10 sampai 20 detik. ISD1400 adalah kompatibel dengan mikrokontroler. Alamat dan jalur kendali dapat dihubungkan dengan mikrokontroler, sehingga mengijinkan

penyimpanan dan pengalamatan yang kompleks. Perekaman disimpan dalam suatu *chip* yang tidak mudah berubah dalam *cell* memori.

2.4 LCD (*Liquid Crystal Display*) M1632

Penampil kristal cair ini terdiri atas tumpukan tipis atau sel dari dua lembar kaca yang sampingnya tertutup rapat. Antar dua lembar kaca tersebut diberi bahan kristal cair (*Liquid Crystal*) yang tembus cahaya. Permukaan luar dari masing-masing keping kaca mempunyai lapisan penghantar tembus cahaya. Sel mempunyai ketebalan sekitar 1×10^{-5} meter dan di isi dengan kristal cair.

Penampil kristal cair berupa dot matrik 5 x 7, tersusun sebanyak dua baris dan masing-masing baris terdiri atas 16 karakter. Dan catu daya yang diperlukan dari *power supply* adalah sebesar +5 volt. Konfigurasi pin pada LCD (*Liquid Crystal Display*) ditunjukkan dalam Gambar 2.10.



Gambar 2.10 LCD (*Liquid Crystal Display*) dan Konfigurasi Pinnya

Sumber: *El-Tech*, 1987: 5

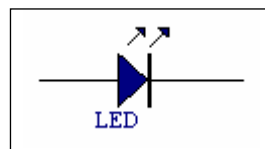
Fungsi dari masing-masing penyemat LCD (*Liquid Crystal Display*) M1632 ditunjukkan sebagai berikut:

- **D0...D7:** Merupakan saluran data, berisi perintah dan data yang akan ditampilkan pada LCD.
- **Enable (E):** Sinyal operasi awal. Sinyal ini akan mengaktifkan data tulis atau baca.
- **R/W:** Sinyal seleksi tulis dan baca: 0 = tulis 1 = baca
- **RS:** Sinyal pemilih register internal (0 = instruksi register (tulis); 1 = data register (tulis dan baca))
- **V_{EE}:** Untuk mengendalikan kecerahan LCD dengan mengubah-ubah nilai resistor variabel yang dihubungkan padanya.

- **Vcc:** Catu daya +5V
- **Vss:** Terminal *ground*.

2.5 *Light Emitting Diode (LED)*

Pada dioda dibias forward, elektron pita konduksi melewati junction dan jatuh ke dalam hole. Pada saat elektron-elektron jatuh dari pita konduksi ke pita valensi, mereka memancarkan energi. Pada dioda penyearah, energi ini keluar sebagai panas. Tetapi pada LED, energi dipancarkan sebagai cahaya. Simbol LED ditunjukkan dalam Gambar 2.11



Gambar 2.11 LED (*Light Emitting Diode*)

Sumber: Malvino 1984 : 99

Dengan menggunakan unsur-unsur seperti *phosphor*, dapat dibuat LED yang memancarkan infra merah (tak kelihatan). LED yang menghasilkan pancaran yang kelihatan dapat berguna pada display peralatan, mesin hitung, jam digital dan lain-lain. LED infra merah dapat digunakan dalam sistem tanda bahaya pencurian dan ruang lingkup lain yang membutuhkan pancaran yang tak kelihatan. Penggunaan LED yang perlu diperhatikan adalah arus yang melewatinya. Secara umum arus maksimal yang mampu dilewati oleh LED sebesar 35 mA, sehingga tegangan yang dipakai dapat bervariasi.

2.6 **DM74LS573 (Latch dengan tiga output)**

Register 8-bit ini memiliki sifat keluaran 3 status *totem-pole* yang dirancang khusus untuk menggerakkan beban yang berimpedansi rendah atau berkapasitas tinggi dan taraf logika tinggi yang melengkapi register-register ini berkemampuan untuk di koneksikan, menggerakkan langsung saluran-saluran bus di dalam sistem yang terorganisasi tanpa memerlukan komponen interface tambahan. IC 74LS573 ini sangat baik digunakan untuk register-register penyangga, gerbang I/O (I/O) port, penggerak bus dua arah dan register-register kerja. Kedelapan latch tersusun

dari flip-flop D. kalau enable (G) rendah, output akan berubah Dalam taraf sebelumnya. Ketika transisi positif, keluaran Q ditetapkan status logika yang sudah ditentukan Dalam masukan-masukan flip-flop D.

Kedelapan keluaran dapat ditempatkan dalam status logika normal (taraf logika rendah atau tinggi) atau status impedansi tinggi dengan buffer tersangga. Dalam status impedansi tinggi keluaran tidak membebani ataupun menggerakkan saluran-saluran bus, keluaran ini tidak mempengaruhi operasi intern flip-flop, artinya data lama dapat dipertahankan atau data baru dapat dimasukkan meskipun output sedang off. Tabel kebenaran IC 74LS573 ditunjukkan dalam Tabel 2.5.

Tabel 2.5 Tabel Kebenaran IC LATCH 74LS573

Output Kontrol	Enable		Output
	G	D	
L	H	H	H
L	H	L	L
L	L	X	Q ₀
H	X	X	Z

Sumber: *Datasheet Fairchild 2000:1*

Keterangan:

- L : Taraf Logika Rendah
- H : Taraf Logika Tinggi
- X : Tak relevan (sembarang jalan masuk, termasuk transisi)
- Q₀ : Taraf yang ada dalam Q sebelumnya
- Z : Taraf off (Impedansi tinggi) keluaran 3 status.z

2.7 Peraturan kejuaraan renang

2.7.1 Peraturan Start

- ✓ Garis *start* kelihatan jelas oleh peralatan tambahan atau peralatan yang dipindahkan dari permukaan air.
- ✓ Isyarat start seharusnya dapat didengar dan dilihat

- ✓ Jika ada perenang yang mengambil keuntungan secara tidak wajar atau curang maka kompetisi akan dihentikan dan dimulai kembali.
- ✓ Di pembukaan sebuah kejuaraan, wasit akan memberi isyarat kepada perenang dengan suatu siulan pendek untuk memberi tahu agar perenang memindahkan semua pakaian kecuali pakaian renang, dan diikuti oleh suatu siulan panjang agar mereka segera mengambil posisi pada permukaan platform. Manakala *official* dan perenang disiapkan untuk start, wasit akan memberi isyarat kepada stater dengan lengan direntangkan, menunjukkan bahwa perenang ada di bawah kendali stater itu. Lengan tangan yang terentang akan tetap pada posisi itu sampai start diberikan.

2.7.2 Peraturan Finish

- ✓ Daerah untuk yang memimpin seharusnya bersih dari peralatan penanda oleh pelampung yang membatasi penutup dengan dinding *finish*.
- ✓ Wasit finish dan pencatat waktu seharusnya ditempatkan agar mampu mengamati finish pada semua waktu.
- ✓ Ketika meninggalkan air beberapa perenang boleh menerima bantuan. Perenang hanya boleh disentuh atau ditangani jika mereka dengan jelas memang membutuhkan atau meminta suatu bantuan.

2.7.3 Pewaktuan

- Ketika peralatan otomatis digunakan, hasilnya akan direkam untuk 1/100 detik. Semua perenang mempunyai rekaman waktu yang sama yakni 1/100 detik pada waktu dan tempat yang sama. Dan hasil akhir yang direkam akan ditampilkan dalam suatu *display*.
- Wasit tetap melakukan *starter*, dalam hal ini adalah *start* pada hitungan ketiga yakni suara “dor” atau yang lainnya.
- Pengambilan nilai dengan arloji akan dilakukan oleh tiga wasit, dimana ketiga arloji tersebut mempunyai tingkat kepresisian yang tinggi dan sudah diakui oleh Negara yang melakukan kejuaraan renang tersebut. Jika

menggunakan peralatan yang otomatis maka ketentuannya adalah sebagai berikut:

- Jika dua dari tiga *stopwatch* merekam waktu yang sama dan yang ketiga berbeda, maka dua waktu yang sama seharusnya menjadi *official time*.
- Jika tiga *stopwatch* tidak sama, maka *stopwatch* yang merekam pertengahan waktu seharusnya menjadi *official time*.
- Jika hanya dua dari tiga *stopwatch* yang bekerja maka waktu rata-rata yang menjadi *official time*.

BAB III

METODOLOGI

Dalam metodologi akan diuraikan cara-cara yang dilakukan dalam perencanaan dan pembuatan alat penghitung waktu untuk kejuaraan renang serta penyusunan laporannya. Adapun metodologi penelitian skripsi yang digunakan adalah sebagai berikut:

3.1 Studi literatur

Mempelajari segala literatur dan dasar teori yang berhubungan dengan perancangan dan pembuatan alat penghitung waktu digital dalam perlombaan renang ini. Data-data yang diperlukan ialah karakteristik dari saklar, sensor sentuh, mikrokontroler, ISD (*Information Storage Devices*), RTC (*Real Time Clock*) dan LCD (*Liquid Crystal Display*)

3.2 Perancangan Peralatan

Berdasarkan studi literatur, tahap selanjutnya adalah perencanaan alat. Adapun hal-hal yang dilakukan dalam perencanaan yaitu penentuan spesifikasi sistem yang dirancang, penyusunan blok diagram sistem untuk mempermudah pemahaman mengenai alur kerja alat yang akan dibuat serta pembuatan skema rangkaian. Pemilihan komponen perangkat keras berdasarkan dalam komponen yang umum dan mudah didapatkan di pasaran lokal, sedangkan untuk perancangan perangkat lunak dimulai dengan membuat diagram alir. Perencanaan *software* meliputi pengambilan data dari RTC, proses penghitungan waktu, pembacaan data dari mikrokontroler untuk ditampilkan pada LCD.

3.3 Perancangan alat secara matematis

Rancangan ini disesuaikan dengan fungsi dari komponen-komponen yang akan digunakan sesuai dengan literatur yang dipelajari sehingga terbentuk

rangkaian elektronik yang siap direalisasikan. Dalam perancangan ini perlu dianalisis secara cermat, sehingga logika keluaran yang diinginkan akan terkondisi (*ter-setting*) dengan baik pada alat ini.

3.4 Pembuatan Alat

Dalam pembuatan alat dilakukan sesuai dengan perencanaan. Adapun tahap pembuatan alat dimulai dengan pembuatan unit rangkaian per blok pada PCB. Selanjutnya tiap blok rangkaian dilakukan pengujian awal. Kemudian dilakukan penggabungan tiap blok menjadi suatu blok rangkaian sesuai dengan perencanaan. Untuk proses pembuatan PCB dibantu dengan menggunakan perangkat lunak Protel PCB. Selanjutnya menyablon *layout* rangkaian pada PCB dan dilanjutkan dengan proses *pengetsaan* dengan tujuan untuk melarutkan tembaga yang tidak digunakan pada permukaan PCB. Setelah proses *pengetsaan* selesai maka jalur pada PCB yang sudah jadi dilapisi dengan larutan perak untuk memudahkan dalam penyolderan sekaligus sebagai pengaman pada jalur PCB. Dan diakhiri dengan melakukan pengeboran lubang tempat kaki-kaki komponen.

Sebagai tahap terakhir dari pembuatan alat adalah perakitan komponen elektronika pada PCB yang telah jadi. Untuk pemasangan komponen diawali dengan memasang komponen pasif seperti *jumper*, resistor, kapasitor, dan *socket* IC. Kemudian dilanjutkan dengan pemasangan komponen aktif seperti IC. Setelah pemasangan komponen tiap blok selesai, dilakukan penggabungan antara blok rangkaian PCB. Sedangkan pembuatan perangkat lunak dengan mengimplementasikan alur program yang telah direncanakan yaitu dengan menggunakan bahasa *assembly*.

3.4 Pengujian Alat

Adapun tujuan pengujian alat adalah untuk memastikan bahwa sistem yang dibuat telah berjalan, yaitu dengan menguji rangkaian perblok dan keseluruhan rangkaian. Metode pengujian alat adalah sebagai berikut:

- Membuat miniatur dari sistem.

- Pengujian sistem penghitung waktu untuk kejuaraan renang berbasis mikrokontroler AT89C51
Bertujuan untuk mengetahui apakah sistem penghitung waktu untuk kejuaraan renang ini bisa bekerja sesuai dengan perencanaan.
- Pengujian perangkat lunak
Bertujuan untuk mengetahui apakah program dapat berjalan sesuai dengan perencanaan. Metode pengujian dengan menggunakan *software* bahasa *assembly*.
- Pengujian keseluruhan sistem
Bertujuan untuk mengetahui apakah sistem dapat bekerja sesuai dengan perencanaan. Metode yang digunakan adalah dengan mengaplikasikan alat secara keseluruhan.

BAB IV

PERANCANGAN ALAT

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

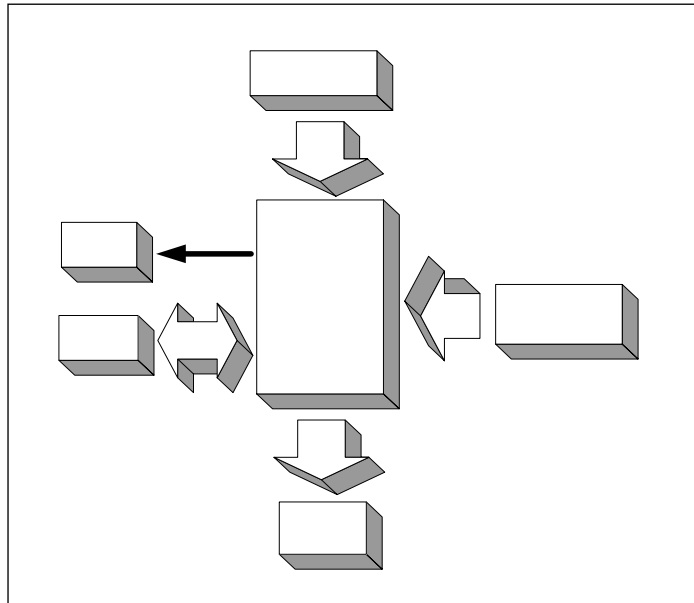
4.1 Spesifikasi Alat

Sistem penghitung waktu untuk kejuaraan renang yang akan dirancang mempunyai spesifikasi teknis sebagai berikut:

1. Menggunakan basis operasi mikrokontroler AT89C51.
2. Menggunakan RTC (*Real Time Clock*) Dallas 12887 sebagai basis waktu.
3. Pemberian informasi saat perenang melakukan persiapan *start*, nomor jalur perenang dilakukan dengan pemutar suara ISD1420.
4. Menggunakan LCD (*Liquid Crystal Display*) untuk menampilkan waktu, serta beberapa tampilan yang telah di-*set* sebelumnya (nomor jalur kolam renang).
5. Menggunakan saklar pada *start* renang dan sensor sentuh pada saat *finishnya*.
6. Menggunakan miniatur kolam renang yang terdiri dari papan *start* dan papan *finish*.

4.2 Diagram Blok Rangkaian

Blok diagram sistem yang dirancang ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Blok Diagram Sistem

4.3 Prinsip Kerja Alat

Pada tempat start (*platform*), dipasang switch atau saklar. Para perenang akan menginjak saklar ini sebelum melakukan perlombaan. Sebagai penanda para peserta telah benar-benar siap melakukan start adalah lampu LED yang akan menyala jika papan start (*platform*) telah diinjak oleh para peserta. ISD akan memberikan aba-aba ketika semua perenang sudah siap. Untuk bunyi aba-aba yang terakhir, misalnya bunyi “dor” tetap dilakukan secara manual oleh wasit, karena meninjau pada peraturan dasar suatu kejuaraan renang.

Kalau wasit belum membunyikan (memencet tombol) tanda “dor” sebagai tanda para peserta mulai berenang, ada salah seorang peserta mendahului start tersebut, mikrokontroler akan memprosesnya dalam *software*nya sehingga ISD akan mengumumkan bahwa peserta tersebut curang. Dan jika wasit sudah membunyikan suara “dor” maka para peserta renang bisa melompat dan meninggalkan tempat start dan secara otomatis sistem penghitung waktunya akan bekerja. RTC disini berfungsi menampilkan jam, menit, detik. Untuk embangkitkan milidetiknya oleh mikrokontroler Kemudian hasil penghitungan waktu tersebut ditampilkan dalam LCD.

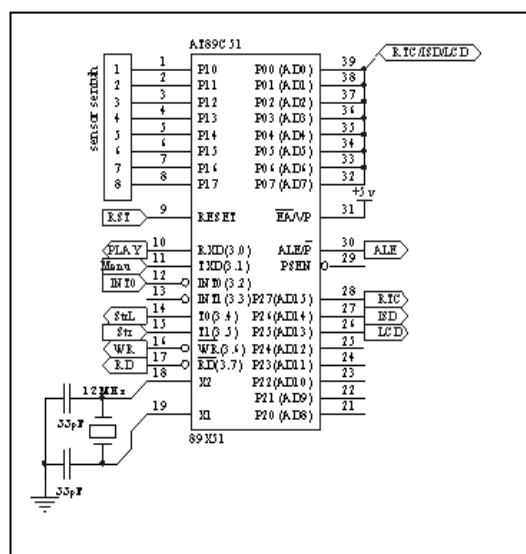
Untuk finishnya ditentukan dengan sensor sentuh. Jadi perenang akan langsung menyentuh sensor sentuh tersebut jika lintasan yang ditentukan dalam perlombaan sudah selesai dilakukannya. Sebagai tanda jika seorang perenang benar-benar telah menyentuh sensor sentuh tersebut adalah menyalaanya LED. Kalau pertandingan sudah selesai, LCD akan menampilkan jumlah waktu dan line yang digunakan perenang.

4.4 Perancangan Perangkat Keras

Perancangan perangkat keras merupakan penjabaran dari masing-masing blok diagram sistem. Perancangan perangkat keras terdiri dari perancangan rangkaian kontrol menggunakan AT89C51, perancangan rangkaian RTC (*Real Time Clock*), perancangan rangkaian perekam/pemutar suara ISD1420, perancangan rangkaian LCD (*Liquid Crystal Display*).

4.4.1 Antarmuka Mikrokontroler AT89C51

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



Gambar 4.2 Rangkaian Antarmuka Mikrokontroler AT89C51

Pin-pin mikrokontroler yang digunakan yaitu:

1. Port 0

P0.0-P0.7 dihubungkan secara paralel dengan ISD1420 (pin A0-A7), *Real Time Clock* (pin AD0-AD7), serta *Liquid Crystal Display* (pin D0-D7).

2. Port 1

P1.0-P1.7 dihubungkan switch untuk Start dan sensor sentuh untuk finish

3. Port 2

- P2.3 dihubungkan ke RS
- P2.5 dihubungkan ke LCD
- P2.6 dihubungkan ke ISD1420.
- P2.7 dihubungkan ke RTC

4. Port 3

- P3.0 dihubungkan ke *play* ISD.
- P3.1 Menu
- P3.4 untuk mengaktifkan *strL*.
- P3.5 untuk mengaktifkan *switch* start.

5. Pin 18 dan pin 19 digunakan sebagai *input* dari rangkaian osilator kristal, yang terdiri atas osilator kristal dengan frekuensi 12 Mhz, kapasitor kristal C1 dan C2 yang masing-masing sebesar 33pF (sesuai *data sheet*, 30 pF±10 pF untuk kapasitor kristal). Rangkaian ini akan membangkitkan pulsa *clock* yang akan menjadi penggerak bagi seluruh operasi internal mikrokontroler.

6. Rangkaian *Reset*

Untuk *mereset* mikrokontroler AT89C51, maka pin RST diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan sinyal *reset* kapasitor dihubungkan dengan Vcc dan sebuah resistor yang dihubungkan ke *ground*. Rangkaian *reset* ditunjukkan dalam Gambar 4.2. Karena kristal yang digunakan mempunyai frekuensi sebesar 12 MHz, maka satu periode membutuhkan waktu sebesar :

$$T = \frac{1}{f_{XTAL}} = \frac{1}{12 \text{ MHz}} \text{ s} = 8,3 \times 10^{-8} \text{ s}$$

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

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

$$= 8,3 \times 10^{-8} \times 24 = 1,992 \mu\text{s}$$

Jadi mikrokontroler membutuhkan waktu minimal 1,992 μs untuk mereset. Waktu minimal inilah yang dijadikan pedoman untuk menentukan nilai R dan C. Dengan menentukan nilai R = 8,2 k Ω , dan C = 10 μF serta V_o adalah tegangan logika nominal yang diijinkan oleh pin RST dimana $V_o = 0,7 \times V_{cc} = 0,7 \times 5 \text{ Volt} = 3,5 \text{ Volt}$, sehingga:

$$t = R \times C \left[\ln \frac{5}{V_o} \right]$$

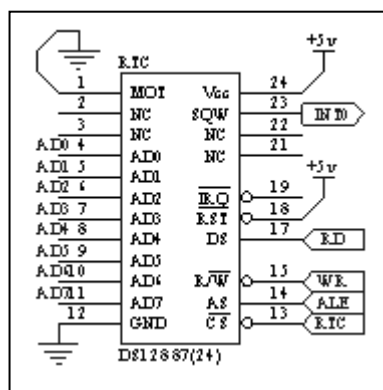
$$\Rightarrow t = 8,2 \text{ k}\Omega \times 10 \mu\text{F} \left[\ln \frac{5}{3,5 \text{ Volt}} \right]$$

$$\Rightarrow t = 29,192 \text{ ms}$$

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

4.4.2 Antarmuka RTC (*Real Time Clock*) IC DS12887

Perancangan rangkaian RTC (*Real Time Clock*) ditunjukkan dalam Gambar 4.3.



Gambar 4.3 Antarmuka IC DS12887 dengan Mikrokontroler

Pin-pin yang digunakan adalah sebagai berikut:

1. GND dan Vcc:

Merupakan penyemat catu daya. Vcc dihubungkan dengan catu daya +5 volt dan GND dihubungkan ke *ground*.

2. MOT — *Motel*:

Digunakan sebagai saklar pemilih mode diagram pewaktuan. Dalam perancangan ini dihubungkan dengan *ground* berarti sistem diagram pewaktuan Intel yang dipakai.

3. AS — *Address Strobe Input*:

Pin AS ini dihubungkan ke latch.

4. AD0–AD7 — *Multiplexed Bidirectional Address/Data Bus*:

Bus alamat/data ini dihubungkan dengan bus alamat pada pin P0.0–P0.7 mikrokontroler.

5. DS — *Data Strobe or Read Input*:

Pin DS dihubungkan ke RD.

6. $\overline{R/W}$ — *Read/Write Input*:

Pin ini dihubungkan ke \overline{WR} .

7. \overline{CS} — *Chip Select Input*:

Pada perancangan ini pin *chip select input* dihubungkan ke *ground*, sehingga RTC (*Real Time Clock*) dalam keadaan selalu aktif

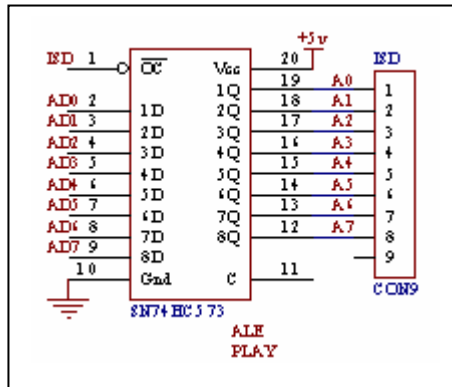
8. \overline{RI} — *Reset Input*:

Sinyal \overline{RESET} ini dihubungkan dengan *reset* pada mikrokontroler.

4.4.3 Antarmuka IC ISD1420 dengan Mikrokontroler

Rangkaian pemutar/perekam suara digunakan untuk memberikan informasi saat perenang sudah siap di papan *start*. IC yang digunakan adalah ISD seri 1420 yang mempunyai durasi penyimpanan suara selama 20 detik. Rangkaian perekam/pemutar suara ini dilengkapi dengan *speaker* sebagai *output* dari suara yang telah direkam sebelumnya. Dalam perancangan alat ini, *speaker* yang digunakan untuk simulasi sistem menggunakan *speaker* 16Ω.

Rangkaian antarmuka ISD1420 dengan mikrokontroler AT89C51 dan *speaker* ditunjukkan dalam Gambar 4.4.



Gambar 4.4 Antarmuka ISD1420 dengan Mikrokontroler

Pin alamat A0-A7 pada ISD1420 dihubungkan dengan Port 0 pada mikrokontroler. Untuk memutar suara informasi, mikrokontroler akan mengirimkan data lokasi alamat tempat penyimpanan suara informasi tersebut ke ISD1420 dan kemudian mengirimkan logika rendah pada pin \overline{PLAYL} . Alamat input dan *playback/record* ditahan dengan adanya transisi turun dari \overline{PLAYL} . Saat *playback cycle*, alamat input menetapkan mulainya alamat dan memutar suara secara kontinyu sampai alamat \overline{EOM} ditemukan (sampai \overline{EOM} berlogika rendah) atau adanya transisi tinggi dari \overline{PLAYL} .

Kata-kata yang direkam pada ISD1420 tidak boleh melebihi durasi waktu yang ditentukan yaitu selama 20 detik. Untuk ISD 20 detik, minimal perekaman hadala 125ms

$$\begin{aligned}
 \text{Jadi alamat untuk ISD 1420} &= \frac{20s}{125ms} \\
 &= 160 \text{ kali desimal} \\
 &= A0h
 \end{aligned}$$

Untuk lebih amannya pembagian alamat penyimpanan suara dirancang sedemikian rupa dengan memperhatikan selang waktu perekaman antara kalimat satu dengan kalimat yang lainnya, agar kalimat yang direkam tidak masuk ke lokasi alamat berikutnya.

Pemilihan lokasi alamat untuk perekaman suara pada sistem ini ditunjukkan dalam Tabel 4.1.

Tabel 4.1 Pemilihan alamat perekaman suara

Alamat Awal	Hexsa	Durasi (Detik)	Kata yang direkam
0000 0000	00	2	Persiapan
0000 1101	0D	1,5	Jalur satu
0001 1100	1C	1,5	Jalur dua
0010 1010	2A	1,5	Jalur tiga
0011 0111	37	1,5	Jalur empat
0100 0100	44	1,5	Jalur lima
0100 1111	4F	1,5	Jalur enam
0101 1010	5A	1,5	Jalur tujuh
0110 0111	67	2	Jalur delapan
0111 0100	74	1	Curang
0111 1101	7D	1,5	Siap-siap
1000 1000	88	1	Dor
		18 Detik	

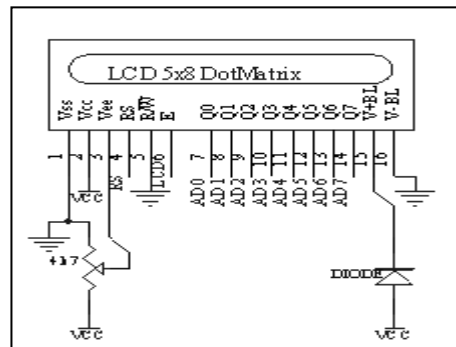
4.4.4 Antarmuka LCD (*Liquid Crystal Display*)

LCD yang digunakan dalam perancangan ini adalah tipe M1632, berfungsi untuk menampilkan informasi jam, menit dan detik yang berubah tiap detiknya serta sebagai tampilan saat pengesetan waktu untuk kejuaraan renang, dan menampilkan data yang telah diset sebelumnya, seperti nama, jurusan dan lain sebagainya.

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

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

Gambar antarmuka LCD (*Liquid Crystal Display*) ditunjukkan dalam Gambar 4.5.



Gambar 4.5 Antarmuka LCD

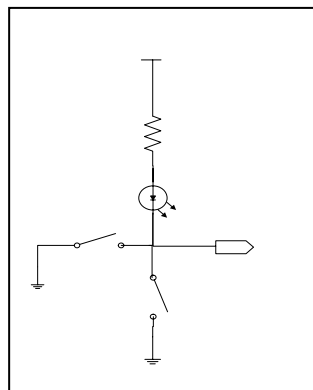
Pin-pin yang digunakan adalah:

- D0-D7
Pin ini dihubungkan pada port 0 mikrokontroler. Pin ini merupakan saluran data, berisi perintah dan data yang akan ditampilkan pada LCD (*Liquid Crystal Display*)
- *Enable* (E)
Pin ini adalah sinyal operasi awal yang akan mengaktifkan data tulis.
- R/\overline{W}
Pin ini adalah sinyal seleksi tulis (logika 0) dan baca (logika 1). Pin dihubungkan ke *ground*, karena dalam perancangan ini LCD (*Liquid Crystal Display*) dikhususkan untuk melakukan proses tulis saja.
- RS
Pin ini adalah sinyal pemilih register internal. Logika 0 berarti instruksi register (tulis), dan logika 1 berarti data register (tulis dan baca). Pin RS ini dihubungkan pada mikrokontroler.
- VEE
Pin ini untuk mengendalikan kecerahan LCD (*Liquid Crystal Display*) dengan mengubah-ubah nilai resistor variabel yang dihubungkan padanya.

- VCC
Pin ini dihubungkan dengan catu daya sebesar +5 volt.
- VSS
Pin ini merupakan terminal *ground*, dihubungkan ke *ground*.

4.4.5 *Switch* sebagai Penanda *Start* dan *Finish*

Switch (saklar) untuk *start* dan *finish* ditunjukkan dalam Gambar 4.6.



Gambar 4.6 *Switch start* dan *finish*

Pada saat saklar 1 (*foot*) diinjak maka LED akan menyala. Ini sebagai penanda pada saat para perenang sudah berdiri di *platform start*. Begitu juga dengan saklar 2 (*touch*). Ketika perenang menyentuh saklar ini, yakni sebagai penanda *finish*, maka LED juga akan menyala.

$$V_{cc} = 5 \text{ volt}$$

$$V_{LED} = 1,7 \text{ volt}$$

$$I_{LED} = 20 \text{ mA}$$

Jadi besarnya resistor (R) yang digunakan adalah:

$$\begin{aligned} R &= \frac{V_{cc} - V_{LED}}{I_{LED}} = \frac{5V - 1,7V}{20mA} \\ &= \frac{3,3V}{20mA} \\ &= 165\Omega \end{aligned}$$

Resistor yang digunakan sebesar 330Ω , yakni resistor yang disediakan di pasaran. Setelah menggunakan resistor 330Ω arus yang dihasilkan menjadi:

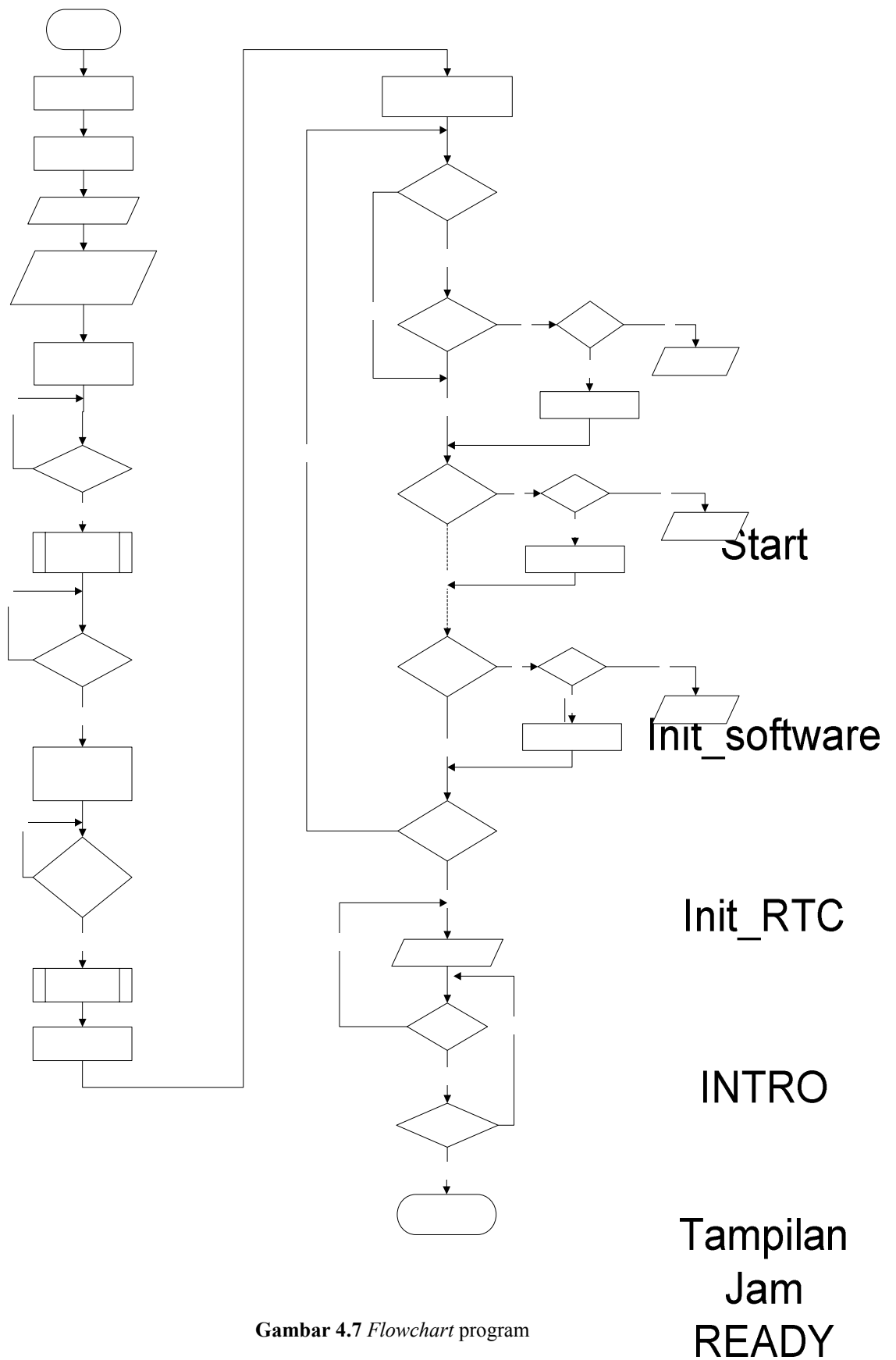
$$\begin{aligned} 330 &= \frac{V_{cc} - V_{LED}}{I_{LED}} = \frac{5V - 1,7V}{I_{LED}} \\ 330 &= \frac{3,3V}{I_{LED}} \\ I_{LED} &= \frac{3,3}{330} \\ &= 10\text{mA} \end{aligned}$$

Jadi arus yang mengalir adalah 10mA. Arus 10mA tersebut cukup untuk menyalakan LED merah.

4.5 Perancangan Perangkat Lunak

Perangkat lunak untuk mengendalikan sistem ini terdiri dari proses pengambilan data dari unit masukan, pengisian dan pengambilan data di RTC (*Real Time Clock*), proses pengambilan data suara di ISD1420 dan program *scanning display*.

Flowchart atau diagram alur sistem pada sistem penghitung waktu untuk kejuaraan renang ditunjukkan dalam Gambar 4.7..



Gambar 4.7 Flowchart program

BAB V

PENGUJIAN ALAT

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

1. Pengujian mikrokontroler AT89C51 – LCD (*Liquid Crystal Display*)
2. Pengujian RTC (*Real Time Clock*) DS12887
3. Pengujian rangkaian perekam/pemutar suara ISD1420
4. Pengujian saklar *start*
5. Pengujian sensor sentuh untuk *finish*
6. Pengujian keseluruhan sistem

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

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

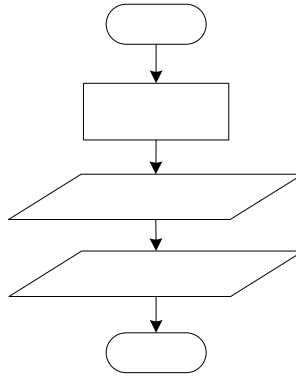
Pengujian terhadap mikrokontroler dan LCD ini ditujukan untuk mengetahui apakah mikrokontroler dan LCD dapat bekerja dengan baik atau tidak. Pengujian ini dilakukan dengan mengisi program inisialisasi LCD ke dalam mikrokontroler dengan menggunakan Flash PEROM (*Programmable and Erasable Read Only Memory*) *programmer*, yang berfungsi untuk mengambil program objek yang ada di PC dan memasukkannya ke dalam Flash PEROM (*Programmable and Erasable Read Only Memory*) AT89C51. Selanjutnya mikrokontroler diberi masukan beberapa karakter dan ditampilkan melalui LCD.

Bagan pengujian mikrokontroler dan LCD ditunjukkan dalam Gambar 5.1.



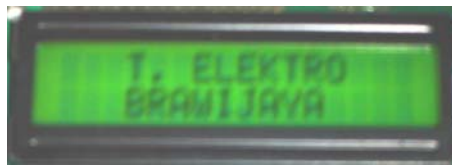
Gambar 5.1 Bagan Pengujian Mikrokontroler dan LCD

Flowchart sistem pengujian mikrokontroler dan LCD ditunjukkan dalam Gambar 5.2.



Gambar 5.2 *Flowchart* Sistem Pengujian Mikrokontroler dan LCD

Tampilan pada LCD dari hasil pengujian ditunjukkan dalam Gambar 5.3.

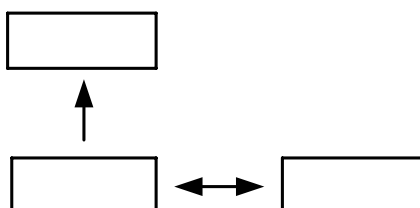


Gambar 5.3 Tampilan pada LCD dari Hasil Pengujian

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

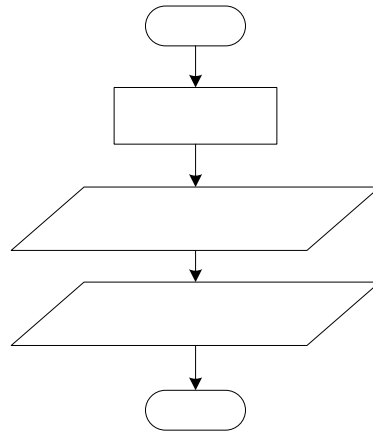
5.2 Pengujian RTC (*Real Time Clock*)

Pengujian terhadap RTC dilakukan dengan memberikan program ke dalam mikrokontroler untuk mengaktifkan RTC. Walaupun saat pengujian catu daya dimatikan, RTC akan tetap aktif karena mempunyai catu daya internal. Bagan pengujian ditunjukkan dalam Gambar 5.4.



Gambar 5.4 Bagan Pengujian RTC

Flowchart pengujian sistem ditunjukkan dalam Gambar 5.5.



Gambar 5.5 *Flowchart* Sistem Pengujian RTC

Tampilan pada LCD yang merupakan hasil dari pengujian RTC ditunjukkan dalam Gambar 5.6.



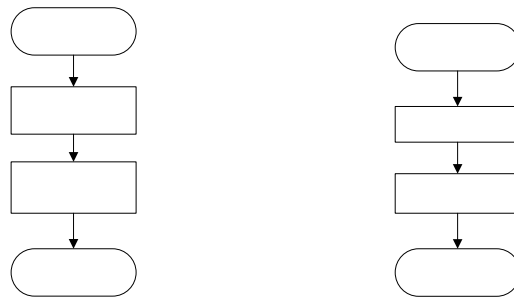
Gambar 5.6 Tampilan LCD hasil pengujian RTC

Dari hasil pengujian, data waktu berupa jam, menit dan detik sudah dapat ditampilkan pada LCD. Hal ini berarti rangkaian RTC telah dapat bekerja dengan baik.

5.3 Pengujian Rangkaian Perekam/Pemutar Suara ISD1420

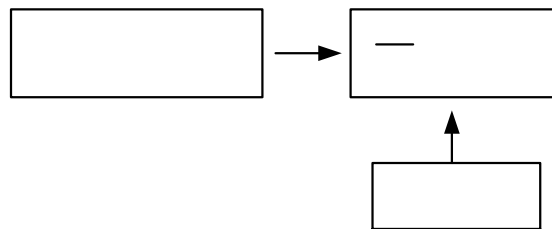
Pengujian terhadap rangkaian ini ditujukan apakah IC ISD1420 dapat dioperasikan untuk merekam suara, dan dapat dioperasikan untuk memutar kembali suara yang telah direkam. Modul ISD 1420 adalah rangkaian pendukung untuk melakukan perekaman pada IC ISD 1420.

Flowchart sistem pengujian ISD1420 ditunjukkan dalam Gambar 5.7.



Gambar 5.7 Flowchart Sistem Pengujian ISD1420

Pengujian pertama dilakukan dengan melakukan perekaman suara pada IC ISD 1420. Bagan perekaman suara ditunjukkan dalam Gambar 5.8.



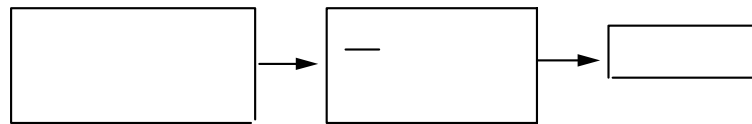
Gambar 5.8 Bagan Perekaman Suara pada IC ISD1420

Pemilihan alamat untuk perekaman suara harus dilakukan secara tepat sesuai dengan durasi waktu perekamannya. Lamanya perekaman akan menentukan banyaknya alamat yang dipakai oleh IC ISD1420. Semakin lama merekam, alamat yang dipakai akan makin banyak. Jika IC ISD1420 digunakan untuk merekam beberapa kata dalam durasi beberapa detik, lamanya perekaman tidak boleh melebihi durasi waktu yang telah dirancang atau masuk pada alamat perekaman berikutnya, karena suara yang telah direkam akan terdengar pada alamat suara berikutnya, dan tentu saja suara yang telah direkam akan terpotong.

Perekaman suara dilakukan dengan merangkai rangkaian seperti pada Gambar 5.8. Saat merekam kata-kata, tombol S3 (*RECORD*) pada modul ISD 1420 harus ditekan, dan tombol baru dilepaskan setelah satu sesi perekaman selesai dilakukan. Dengan menekan tombol S3 (*RECORD*), berarti memberi logika 0 pada pin \overline{REC} di IC ISD 1420.

Pengujian kedua adalah untuk mengetahui apakah IC ISD 1420 dapat dioperasikan untuk memutar kembali suara yang telah direkam, dan apakah suara

telah terekam dengan baik dan tidak terpotong oleh rekaman suara yang lain. Bagan pengujian untuk memutar rekaman suara ditunjukkan dalam Gambar 5.9.



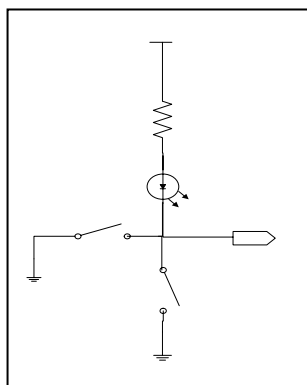
Gambar 5.9 Bagan Pemutaran Ulang Rekaman Suara pada IC ISD 1420

Pemutaran hasil perekaman suara dilakukan dengan merangkai rangkaian seperti Gambar 5.9. Untuk memutar rekaman suara dilakukan dengan men-*set* alamat awal perekaman seperti yang ditunjukkan dalam Tabel 5.1 dan selanjutnya tombol PLAYL ditekan. Rekaman suara akan diperdengarkan melalui *speaker*. Penekanan tombol PLAYL membuat pin \overline{PLAYL} menjadi berlogika 0. Setelah rekaman suara selesai diputar, secara otomatis pin \overline{PLAYL} akan menjadi berlogika 1.

Dari hasil pengujian, rekaman suara tidak terpotong oleh rekaman suara yang lain, dan sudah cukup terdengar dengan baik. Hal ini menunjukkan bahwa rangkaian perekam dan pemutar suara ISD 1420 telah bekerja dengan baik. Misalnya diberi alamat 0000 0000 maka suara yang keluar adalah persiapan.

5.4 Pengujian Saklar *Start (Switch)*

Tujuan dari pengujian rangkaian saklar start adalah untuk mengetahui besarnya tegangan yang diumpankan ke mikrokontroler saat *limit switch* dalam keadaan tertekan dan saat keadaan tidak tertekan. Saat terjadi penekanan terhadap *limit switch* berarti para perenang telah siap berdiri di papan start (*platform*), dan saat tidak terjadi penekanan terhadap *limit switch* berarti para perenang belum siap berdiri di papan start (*platform*) Pengujian ditunjukkan dalam Gambar 5.10



Gambar 5.10 Rangkaian Start

Data hasil pengujian ditunjukkan dalam Tabel 5.1

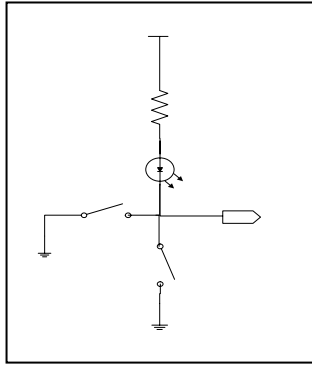
Tabel 5.1 Data Hasil pengujian Rangkaian

Kondisi	Logika
<i>Limit switch</i> tidak tertekan	1
<i>Limit switch</i> tertekan	0

Berdasarkan hasil pengujian didapatkan bahwa saat *limit switch* tertekan tegangan yang terukur sebesar 5 Volt, dalam hal ini berarti MCU berlogika 1. Dan saat *limit switch* tidak tertekan, tegangan yang terukur sebesar 0 volt, dalam hal ini berarti MCU berlogika 0. Dengan demikian rangkaian ini telah berkerja sesuai spesifikasi yang diinginkan.

5.5 Pengujian Sensor Sentuh

Tujuan dari pengujian rangkaian sensor sentuh adalah untuk mengetahui besarnya tegangan yang diumpankan ke mikrokontroler saat *limit switch* dalam keadaan tertekan dan saat keadaan tidak tertekan. Saat terjadi penekanan terhadap *limit switch* berarti para perenang telah selesai melakukan putaran perlombaan renang, dan saat tidak terjadi penekanan terhadap *limit switch* berarti para perenang belum selesai melakukan putaran perlombaan renang. Pengujian ditunjukkan dalam Gambar 5.11



Gambar 5.11 Rangkaian Sensor Sentuh

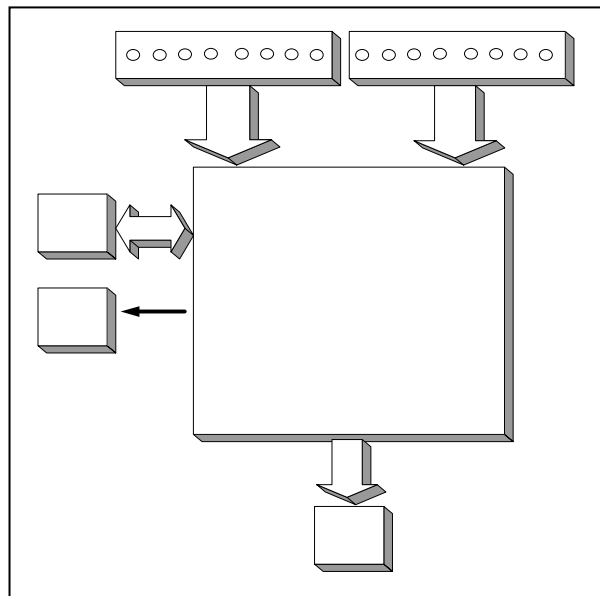
Data hasil pengujian ditunjukkan dalam Tabel 5.2.

Tabel 5.2 Data Hasil pengujian Rangkaian

Kondisi	Logika
<i>Limit switch</i> tidak tertekan	1
<i>Limit switch</i> tertekan	0

5.6 Pengujian Keseluruhan Sistem

Pengujian keseluruhan sistem ini ditujukan untuk mengetahui kinerja sistem secara keseluruhan, apakah telah dapat berhasil sesuai dengan spesifikasi yang diharapkan atau tidak. Pengujian dilakukan dengan merangkai keseluruhan blok rangkaian seperti ditunjukkan dalam Gambar 5.12.



Gambar 5.12 Bagan Pengujian Keseluruhan Sistem

Selanjutnya program yang telah diisikan ke dalam MCU AT89C51 digunakan untuk menguji keseluruhan sistem. Langkah-langkah pengujian dilakukan sebagai berikut:

Pengujian I

1. Pengecekan waktu dari RTC (jam, menit, detik).
2. Pengesetan waktu untuk mengaktifkan dan mematikan waktu (*timer*) pada perlombaan renang.
3. Pengecekan apakah pemutar suara ISD 1420 dan LCD bekerja sesuai dengan perencanaan.

Pengujian II

1. Menghidupkan catu daya.
2. Pilih menu perlombaan renang (untuk 50m, 100m, 200m), kemudian tekan *enter* untuk mengaktifkan kelas perlombaan renang tersebut.
3. ISD akan memberi informasi agar para perenang bersiap-siap.
4. Tekan semua saklar *start* (LED merah nyala).

5. Kalau LED sudah menyala semua, tekan *enter*. *Timer* aktif.

Pengujian III

1. Untuk kelas 50m, langsung tekan finish (tanpa balik), untuk kelas 100m balik satu kali kemudian tekan finish dan untuk kelas 200m balik dua kali kemudian tekan finish (sensor sentuh).
2. LCD dan ISD (suara) akan menampilkan waktu hasil perlombaan dari line satu sampai line delapan.
3. Tekan menu untuk melihat kembali tampilan dan tekan enter jika ingin mengakhiri.

Setelah dilakukan pengujian I, didapatkan bahwa rangkaian pemutar suara dan *timer* renang sesuai waktu hasil pengesetan. Selanjutnya setelah dilakukan pengujian II, didapatkan bahwa LED langsung nyala untuk persiapan perenang dan timer langsung bekerja ketika ditekan *enter* (dor), sedangkan rangkaian pemutar suara aktif. Setelah dilakukan pengujian III, didapatkan bahwa begitu sensor sentuh ditekan waktu akan berhenti dan langsung ditampilkan hasilnya oleh LCD.

Dari hasil pengujian yang telah dilakukan, didapatkan kesimpulan bahwa alat yang dibuat telah bekerja dengan baik sesuai spesifikasi yang telah direncanakan.

Hasil pengujian ditunjukkan dalam Tabel 5.3:

Tabel 5.3 Tampilan Waktu Renang pada LCD

No	Jalur	Waktu
1.	Jalur satu	00 : 15 : 28,45
2.	Jalur dua	00 : 17 : 36,10
3.	Jalur tiga	00 : 20 : 57,15
4.	Jalur empat	00 : 25 : 15,25
5.	Jalur lima	00 : 12 : 20,30
6.	Jalur enam	00 : 17 : 89,20
7.	Jalur tujuh	00 : 26 : 41,15
8.	Jalur delapan	00 : 13 : 36,40

BAB VI

PENUTUP

6.1 Kesimpulan

Dari hasil perencanaan dan pembuatan sistem penghitung waktu untuk kejuaraan renang, dapat diperoleh kesimpulan sebagai berikut:

1. Dalam pembuatan sistem ini, mikrokontroler secara kontinu senantiasa membaca serta mengambil data dari RTC sebagai basis pewaktuan, waktu pengambilan data dari line satu kembali ke line satu (satu periode) adalah $100\mu\text{s}$.
2. RTC berfungsi sebagai akurasi waktu untuk jam, menit, detik dalam suatu perlombaan renang.
3. Dengan menggunakan kabel jenis koaksial tembaga, saklar sentuh dapat bekerja dengan baik sesuai dengan pengujian, diletakkan sejauh 30m dari mikrokontroler.
4. RTC bisa menampilkan dalam orde detik dan sistem mampu menampilkan dalam orde milidetik kemudian data tersebut ditampilkan di LCD

6.2 Saran

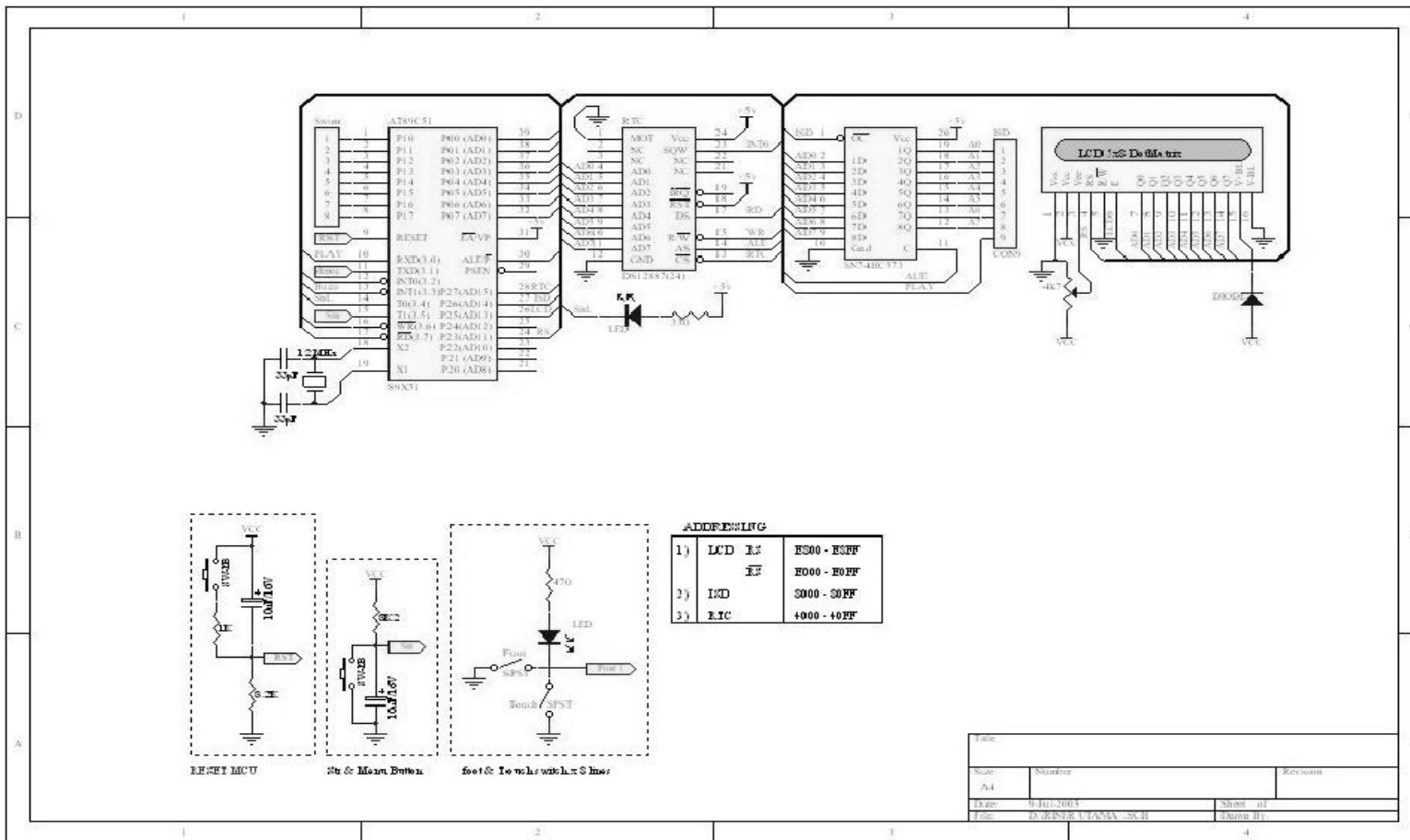
Untuk lebih menyempurnakan pembuatan sistem penghitung waktu untuk kejuaraan renang ini, disarankan untuk menambahkan hal-hal berikut:

1. Penambahan sistem interface yang ditujukan untuk menambah jumlah kategori dalam suatu kejuaraan, misalnya dengan penggunaan PPI (*Programmable Peripheral Interface*).
2. Pewaktuan lebih presisi, dalam arti bisa menampilkan tiga atau lebih angka di belakang koma.
3. Tampilan dapat dibuat modul *driver*. Misalnya pemakaian *dot matrix* agar lebih jelas tampilannya

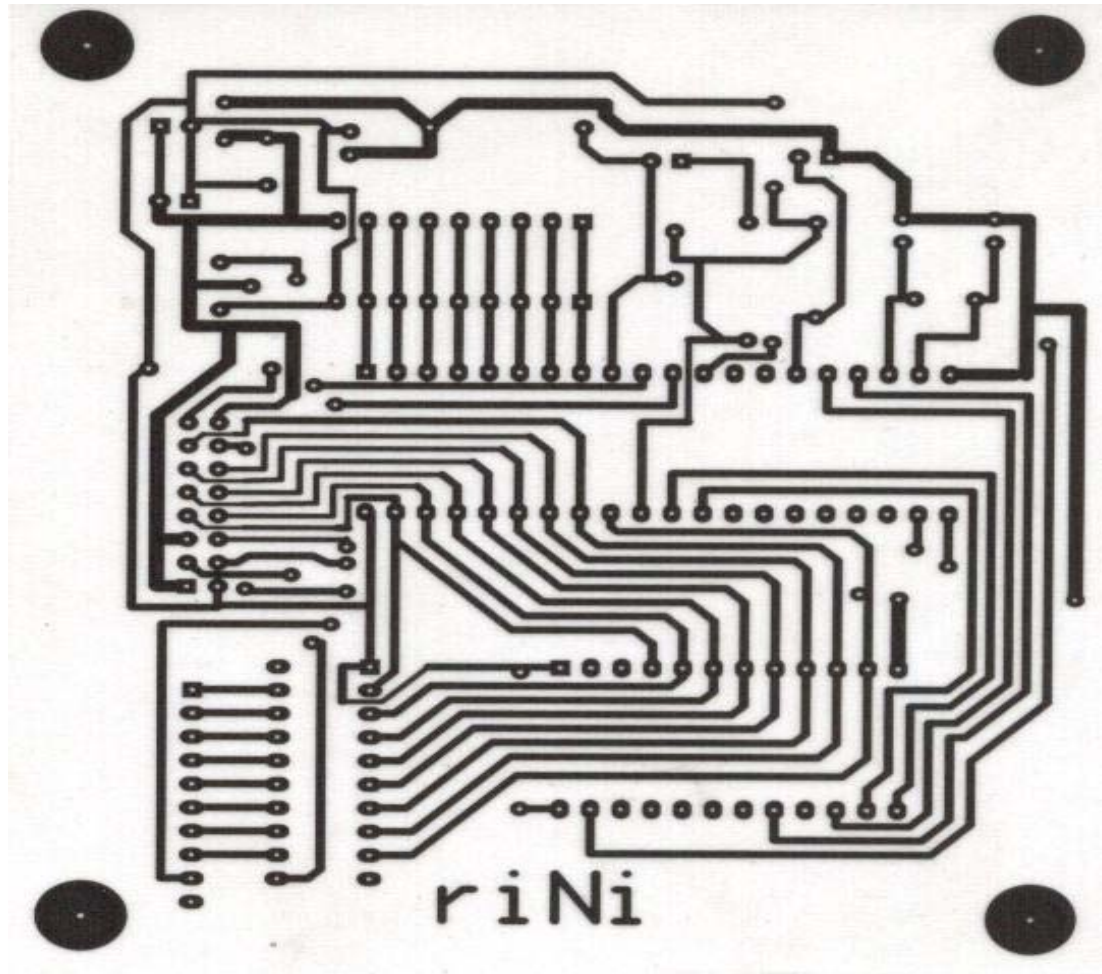
DAFTAR PUSTAKA

- Anonymous. 1987. *Liquid Crystal Display Module M1632 User Manual*. Surabaya: El-tech Electronics.
- Anonymous. 1996. *FINA Swimrules*. Tanggal akses: oktober 2002-2005.
- Anonymous. 1997. *8-Bit Microcontroller with 4K Bytes Flash*. Atmel corporation. Akses dari: www.atmel.com. Tanggal Akses: 28 September 2004.
- Anonymous. 1997. *ISD1400 Series Single-Chip Voice Record/Playback Devices 16- and 20- Second Durations*. Elfa Corporation. Akses dari: www.elfa.se. Tanggal akses: Oktober 2004.
- Anonymous. 1998. *CD4094BC 8-bit Shift Register/Latch 3-State Ouput*. Fairchild Semiconductors Corporation. Akses dari: www.National.com. Tanggal akses: 3 Desember 2004.
- Anonymous. 1998. *DS12887 Real Time Clock*. Dallas Semiconductor. Akses dari: www.dallas.com. Tanggal akses: oktober 2004.
- Hewes, J. 2005. *Electronics Club*. Kelsey Park School, www.kpsec.freeuk.com
- Malvino, Albert Paul. 1995. *Prinsip-Prinsip Elektronika Edisi Kedua*. Alih Bahasa: Hanapi Gunawan. Jakarta: Erlangga.
- Nalwan, Paulus Andi. 2002. *Panduan Praktis Teknik Antarmuka dan Pemrograman Mikrokontroler AT89C51*. Jakarta: PT. Elex Media Komputindo.
- Putro, Eko Agfianto. 2002. *Belajar Mikrokontroler AT89C51/52/55 (Teori dan Aplikasi)*. Yogyakarta: Gava Media.

LAMPIRAN L-1
RANGKAIAN LENGKAP
LAYOUT PCB



Gambar L-1a Rangkaian Lengkap



Gambar L-1b *Layout PCB*

LAMPIRAN L-2
FOTO ALAT
MEKANIK ALAT

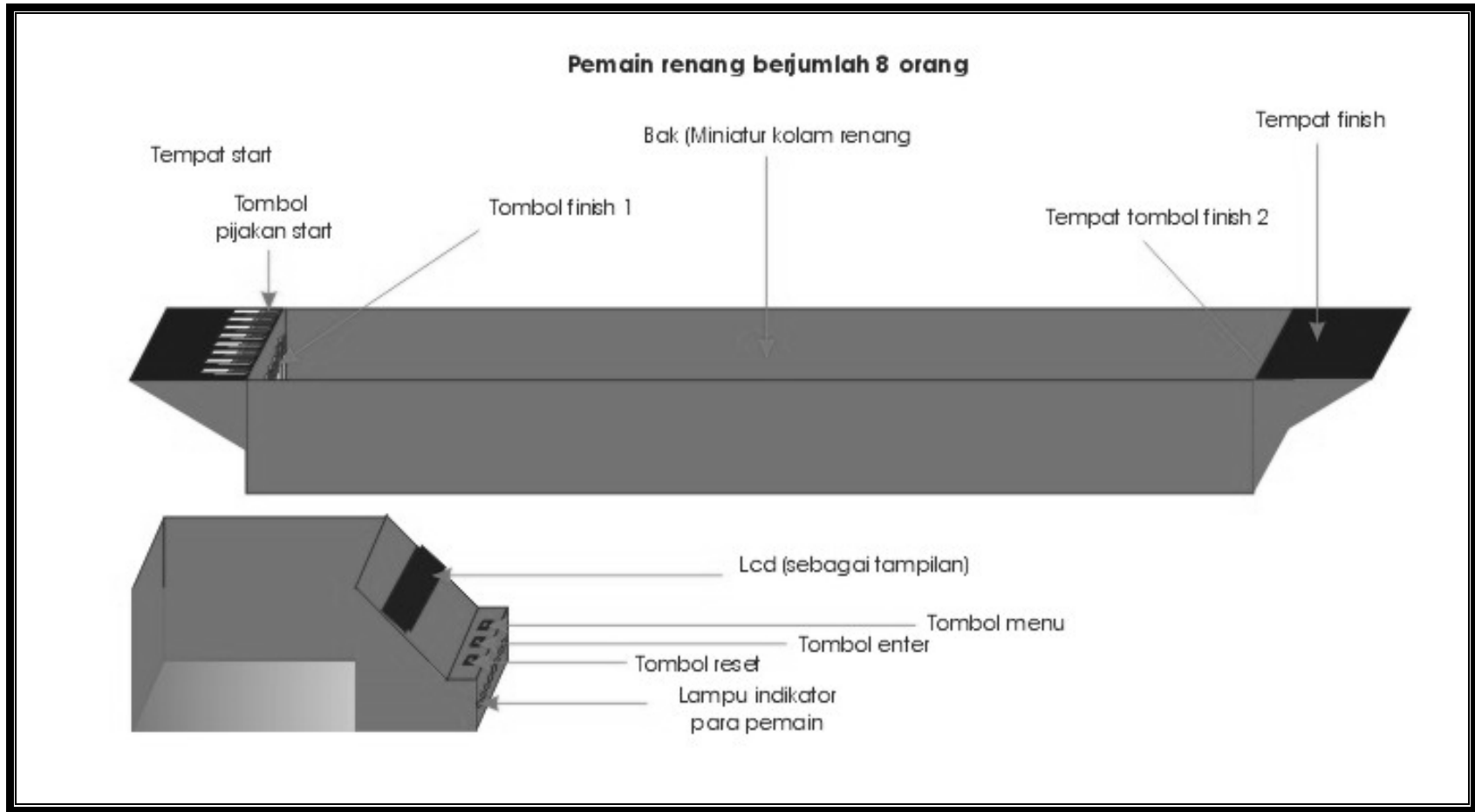
FOTO ALAT



Gambar L-2a Foto Alat Terbuka Tampak Atas



Gambar L-2b Foto Alat Tertutup Tampak Atas



Gambar L-2c Mekanik Alat

LAMPIRAN L-3
LISTING PROGRAM

;TIMER RENANG

```

;RINI 06-2005
;ELEKTRO-UNIBRAW
;----- EQUATION
    RTCc      Equ 4000h
    Line_1    Equ 080h
    Line_2    Equ 0c0h
    ISDd      Bit P2.6
    LCDd      Bit P2.5
    LCDRS     Bit P2.3
    Swim      Equ P1
    Play      Bit P3.0
    Menu      Bit P3.1
    StrL      Bit P3.4
    Strr      Bit P3.5
    Kelas     Equ 20h
    Finish    Equ 21h
    Balik     Equ 22h
    menit     Equ 25h
    detik     Equ 26h
    Line1     Equ 31h   ;32 33 34, kls
    Line2     Equ 35h   ;36 37 38, kls
    Line3     Equ 39h   ;3a 3b 3c, kls
    Line4     Equ 3dh   ;3e 3f 40, kls
    Line5     Equ 41h   ;42 43 44, kls
    Line6     Equ 45h   ;46 47 48, kls
    Line7     Equ 49h   ;4a 4b 4c, kls
    Line8     Equ 4dh   ;4e 4f 50, kls

;-----VEKTOR ROM
        org 00h
        ljmp Mulai

        org 03h
        ljmp Intx0

        org 0Bh
        ljmp IntT0

;-----INISIALISASI SOFTWARE
initSoft:  mov TMOD,#01h
           mov TL0,#05h
           mov TH0,#D7h      ;250x40=10ms
           mov IE,#83h ;EA,EX0,ET0
           setb IT0
           clr EX0
           mov r7,#02h ;rtc
           mov r1,#10h
inita:    mov @r1,#00h
           inc r1
           cjne r1,#55h,inita
           mov 11h,#00h      ;tmr10
           mov 12h,#00h      ;tmr100
           mov 25h,#00h
           mov 26h,#00h
           ret
;-----interrupt IntT0

```

```

IntT0:      clr EA
            push PSW
            push acc
            mov TL0,#05h
            mov TH0,#D7h
            inc 11h
            mov a,11h
            cjne a,#0Ah,intT0x
            mov 11h,#00h
            inc 12h
            mov a,12h
            cjne a,#0Ah,intT0x
            mov 12h,#00h
intT0x:     pop acc
            pop PSW
            setb EA
            reti

```

```

;-----interrupt INTX0

```

```

IntX0:      clr EA
            push dph
            push dpl
            push acc
            push PSW
            djnz r7,into
            lcall IncTime
            mov r7,#02h
into:       pop PSW
            pop acc
            pop dpl
            pop dph
            setb EA
            reti

```

```

;-----Prosedur LCD

```

```

wait1:      push 02h
            mov r2,#9fh
            djnz r2,$
            pop 02h
            ret
setLCD:     ;mov dptr,#0E000h
            setb LCDd
            clr LCDRS
            mov P0,a
            lcall wait1
            clr LCDd
            ret
wait2:      push 02h
            push 03h
            mov r2,#08h
waitms:     mov r3,#0fah
            djnz r3,$
            djnz r2,waitms
            pop 03h
            pop 02h
            ret

```

```

initLCD:    lcall wait2
            mov a,#38h          ;8bit Data Length
            lcall setLCD
            mov a,#06h         ;Mode inc
            lcall setLCD
            mov a,#0ch         ;turn on total display
            lcall setLCD
clearLCD:   mov a,#01h         ;display clear
            lcall setLCD
            lcall wait2
CrsrHome:  mov a,#02h         ;cursor home
            lcall setLCD
            lcall wait2
            ret
blinkON:    mov a,#0dh         ;Blink On
            lcall setLCD
            ret
blinkOFF:   mov a,#0ch         ;Blin Off
            lcall setLCD
            ret
dispTP:     ;mov dptr,#0E800h
            setb LCDd
            setb LCDRS
            mov P0,a
            lcall wait1
            clr LCDd
            clr LCDRS
            ret
dispMSG:    lcall setLCD
loopDISP:   mov a,#00h
            movc a,@a+dptr
            cjne a,#00h,loop2
            clr LCDd
            ret
loop2:      lcall dispTP
            inc dptr
            ljmp loopDISP
dispMSGi:   lcall setLCD
loopDISPi:  mov a,#00h
            mov a,@r0
            cjne a,#00h,loop2i
            clr LCDd
            ret
loop2i:     lcall dispTP
            inc r0
            ljmp loopDISPi

dispMSGiL:  lcall setLCD
            push 02h
            mov r2,#14h
loopDISPiL: mov a,@r0
            cjne a,#00h,loop2iL
loop3iL:    pop 02h
            clr LCDd
            ret
loop2iL:    lcall dispTP

```



```

        inc r0
        djnz r2,loopDISPiL
        ljmp loop3il

shiftA:  mov a,#16h           ;Righ Shift
         lcall setLCD
         djnz b,shiftA
         lcall blinkON
         ret

Write:   clr A
         movC A,@A+DPTR
         JZ WriteEnd
         call sendData
         inc DPTR
         sjmp Write

WriteEnd: ret

SendData: ;mov dptr,#0E800h
          setb LCDd
          setb LCDRS
          mov P0,a
          lcall wait1
          clr LCDd
          clr LCDRS
          ret

Shift:  mov a,#18h
         lcall SetLCD
         ret

;-----Rutin InitRTC
InitRTC: mov DPTR,#400Ah    ;addr reg A
         mov A,#2Fh        ;500 ms SQW
         movx @DPTR,A
         nop
         nop
         nop
         mov DPTR,#400Bh   ;reg B
         mov A,#0Ah       ;SQWE, BCD, 24
         movx @DPTR,A
         nop
         nop
         ret

;-----Rutin SetRTC
SetRTC:  mov A,#0AFh
         mov DPTR,#400ah
         movx @DPTR,A
         mov A,#8Ah
         mov DPTR,#400bh
         movx @DPTR,A
         mov A,#05h      ;Tahun
         mov DPTR,#4009h
         movx @DPTR,A
         mov A,#07h      ;Bulan
         mov DPTR,#4008h
         movx @DPTR,A
         mov A,#03h      ;Tanggal

```

```

Mov DPTR,#4007h
MovX @DPTR,A
Mov A,#01h           ;Hari
Mov DPTR,#4006h
MovX @DPTR,A
Mov A,#15h          ;Jam
Mov DPTR,#4004h
MovX @DPTR,A
Mov A,#05h         ;Menit
Mov DPTR,#4002h
MovX @DPTR,A
Mov A,#01h         ;Detik
Mov DPTR,#4000h
MovX @DPTR,A
Mov A,#2Fh
Mov DPTR,#400ah
MovX @DPTR,A
Mov A,#0Ah
Mov DPTR,#400bh
MovX @DPTR,A
ret

;-----tampilkan jam update dari rtc
IncTime:  push acc
          push b
          push 00h
          push 01h
          mov a,#84h           ;----xx:xx:xx----;
          lcall setLCD
          lcall delay_1
          mov dptr,#4004h
          movx a,@dptr
          lcall delay_1
          ;mov jam,a
          lcall writedataRTC
          mov a,#':'
          lcall sendData
          mov dptr,#4002h
          movx a,@dptr
          lcall delay_1
          ;mov menit,a
          lcall writedataRTC
          mov a,#':'
          lcall sendData
          mov dptr,#4000h
          movx a,@dptr
          lcall delay_1
          ;mov detik,a
          lcall writedataRTC
          pop 01h
          pop 00h
          pop b
          pop acc
          ret

WriteDataRtc:  push 06h
              mov r6,a

```

```

        anl a,#0f0h
        swap a
        add a,#30h
        lcall senddata
        mov a,r6
        anl a,#0fh
        add a,#30h
        lcall senddata
        pop 06h
        ret

;-----INTRO
Intro_1:      db 'PUDYASTI PUSPITARINI 0001063345-63      ',00h
Intro_2:      db 'TIMER RENANG Berbasis Mikrokontroler  ',00h
Intro_3:      db 'Elektro Unibraw      Elektro Unibraw  ',00h
Intro_4:      db ' Skripsi 2005      Skripsi 2005      ',00h

Intro:        lcall ClearLCD
              mov dptr,#Intro_1
              mov a,#line_1
              lcall dispMSG
              mov dptr,#Intro_3
              mov a,#line_2
              lcall dispMSG
              lcall del05sec
              lcall del05sec
              lcall del05sec
              lcall del05sec
              lcall dspShift
              lcall del05sec
              lcall del05sec
              lcall ClearLCD
              mov dptr,#Intro_2
              mov a,#line_1
              lcall dispMSG
              mov dptr,#Intro_4
              mov a,#line_2
              lcall dispMSG
              lcall del05sec
              lcall del05sec
              lcall del05sec
              lcall del05sec
              lcall dspShift
              lcall del05sec
              lcall del05sec
              ret

;-----DELAY
del05sec:     push 04h
              push 05h
              push 06h
              mov r4,#06h      ;05detik
del_a:        mov r5,#0a5h
del_b:        mov r6,#0ffh
              djnz r6,$
              djnz r5,del_b

```

```

        djnz r4,dela
        pop 06h
        pop 05h
        pop 04h
        ret

del025sec:  push 04h
            push 05h
            push 06h
            mov r4,#03h          ;025detik
delc:      mov r5,#0a3h
deld:      mov r6,#0ffh
            djnz r6,$
            djnz r5,deld
            djnz r4,delc
            pop 06h
            pop 05h
            pop 04h
            ret

del01sec:  push 05h          ;01detk
            push 06h
            mov r5,#0c3h
dele:      mov r6,#0ffh
            djnz r6,$
            djnz r5,dele
            pop 06h
            pop 05h
            ret

delay_2:   push 04h          ;10ms
            mov r4,#28h
delf:      lcall delay_1
            djnz r4,delf
            pop 04h
            ret

delay_1:   push 04h          ;250us
            mov r4,#0f9h
            djnz r4,$
            pop 04h
            ret

```

```

;-----PROGRAM RENANG

```

```

Persiap:   db 'Ke Papan Loncat ',00h
Siap22:    db '   Siap-siap   ',00h

```

```

Renang:    push 01h
            mov a,Kelas
            mov 34h,a
            mov 38h,a
            mov 3ch,a
            mov 40h,a
            mov 44h,a
            mov 48h,a
            mov 4ch,a
            mov 50h,a

```

```

renn00:    lcall del05sec
           lcall del05sec
           lcall del05sec
           clr EX0
           lcall clearLCD
           mov dptr,#Persiap
           mov a,#Line_2
           lcall dispMSG
           setb ISDd
           mov P0,#00h ;prsiapn
           clr ISDd
           setb EX0
           clr PLAY
           lcall del025sec
           setb PLAY

           mov P1,#0ffh
           push 03h
           mov r3,#0ch
ulgbb:    lcall del05sec
           djnz r3,ulgbb
           pop 03h
Renn1:    lcall del05sec
           mov a,P1
           cjne a,#00h,Renn1
           lcall del05sec
           lcall del05sec
           lcall del05sec
           lcall del05sec
           lcall del05sec
           lcall del05sec
           lcall del05sec
           clr EX0
           lcall clearLCD
           mov dptr,#Siap22
           mov a,#Line_2
           lcall dispMSG
           setb ISDd
           mov P0,#7dh ;siap2
           clr ISDd
           setb EX0
           clr PLAY
           lcall del025sec
           setb PLAY
           setb strr
           lcall del05sec
           lcall del05sec

           jb strr,$

           jnb P1.0,rennaa
           clr EX0
           setb ISDd
           mov P0,#0dh ;1
           clr ISDd
           setb EX0

```

```

        clr PLAY
        lcall del025sec
        setb PLAY
        lcall Curang
        ljmp Renn00

rennaa:  jnb P1.1,rennbb
        clr EX0
        setb ISDd
        mov P0,#1ch ;L2
        clr ISDd
        setb EX0
        clr PLAY
        lcall del025sec
        setb PLAY
        lcall Curang
        ljmp Renn00

rennbb:  jnb P1.2,renncc
        clr EX0
        setb ISDd
        mov P0,#2ah ;L3
        clr ISDd
        setb EX0
        clr PLAY
        lcall del025sec
        setb PLAY
        lcall Curang
        ljmp Renn00

renncc:  jnb P1.3,renndd
        clr EX0
        setb ISDd
        mov P0,#37h ;4
        clr ISDd
        setb EX0
        clr PLAY
        lcall del025sec
        setb PLAY
        lcall Curang
        ljmp Renn00

renndd:  jnb P1.4,rennee
        clr EX0
        setb ISDd
        mov P0,#44h ;L5
        clr ISDd
        setb EX0
        clr PLAY
        lcall del025sec
        setb PLAY
        lcall Curang
        ljmp Renn00

rennee:  jnb P1.5,rennff
        clr EX0
        setb ISDd

```

```

        mov P0,#4fh ;L6
        clr ISDd
        setb EX0
        clr PLAY
        lcall del025sec
        setb PLAY
        lcall Curang
        ljmp Renn00

rennff:  jnb P1.6,renngg
        clr EX0
        setb ISDd
        mov P0,#5ah ;L7
        clr ISDd
        setb EX0
        clr PLAY
        lcall del025sec
        setb PLAY
        lcall Curang
        ljmp Renn00

renngg:  jnb P1.7,rennhh
        clr EX0
        setb ISDd
        mov P0,#67h ;L8
        clr ISDd
        setb EX0
        clr PLAY
        lcall del025sec
        setb PLAY
        lcall Curang
        ljmp Renn00

ulang:   db ' d i u l a n g ',00h

Curang: lcall del05sec
        lcall del05sec
        lcall del05sec
        lcall del025sec
        lcall del01sec
        clr EX0
        lcall clearLCD
        mov dptr,#ulang
        mov a,#Line_2
        lcall dispMSG
        setb ISDd
        mov P0,#74h ;curang
        clr ISDd
        setb EX0
        clr PLAY
        lcall del025sec
        setb PLAY
        push 03h
        mov r3,#10h
ulгаа:  lcall del05sec
        djnz r3,ulгаа

```

```

        pop 03h
        ret

rennhh:    ;clr StrL
           setb ISDd
           mov P0,#88h ;doorr
           clr ISDd
           clr PLAY
           clr StrL
           lcall del025sec
           setb PLAY
           setb TR0
           setb ET0
           clr EX0
           mov 25h,#00h
           mov 26h,#00h
           lcall clearLCD
           mov dptr,#doorr
           mov a,#Line_2
           lcall dispMSG
           setb EX0
           lcall del05sec
           lcall del05sec
           lcall del05sec
           lcall del05sec
           lcall del05sec
           lcall del05sec
           ljmp Lomba

doorr:    db ' d i m u l a i ',00h

Lomba:    jb P1.0,lombaaa
           mov a,34h
           cjne a,#00h,lomba11
           jb 22h.0,lombaaa
           jb 21h.0,lombaaa
           mov R1,#Line1
           lcall Tangkap
           setb 21h.0      ;21 finish
           ljmp lombaaa

lomba11:  jb 22h.0,lombaaa ;22 balik
           dec 34h
           setb 22h.0

lombaaa:  jb P1.1,lombabb
           mov a,38h
           cjne a,#00h,lomba22
           jb 22h.1,lombabb
           jb 21h.1,lombabb
           mov R1,#Line2
           lcall Tangkap
           setb 21h.1
           ljmp lombabb

lomba22:  jb 22h.1,lombabb
           dec 38h
           setb 22h.1

```



```

Lombabb:    jb P1.2,lombacc
            lcall delay_2
            mov a,3ch
            cjne a,#00h,lomba33
            jb 22h.2,lombacc
            jb 21h.2,lombacc
            mov R1,#Line3
            lcall Tangkap
            setb 21h.2
            ljmp lombacc
lomba33:    jb 22h.2,lombacc
            dec 3ch
            ;djnz 3ch,lombacc
            setb 22h.2

lombacc:    jb P1.3,lombadd
            mov a,40h
            cjne a,#00h,lomba44
            jb 22h.3,lombadd
            jb 21h.3,lombadd
            mov R1,#Line4
            lcall Tangkap
            setb 21h.3
            ljmp lombadd
lomba44:    jb 22h.3,lombadd
            dec 40h
            ;djnz 40h,lombadd
            setb 22h.3

Lombadd:    jb P1.4,lombaee
            mov a,44h
            cjne a,#00h,lomba55
            jb 22h.4,lombaee
            jb 21h.4,lombaee
            mov R1,#Line5
            lcall Tangkap
            setb 21h.4
            ljmp lombaee
lomba55:    jb 22h.4,lombaee
            dec 44h
            ;djnz 44h,lombaee
            setb 22h.4

lombaee:    jb P1.5,lombaff
            mov a,48h
            cjne a,#00h,lomba66
            jb 22h.5,lombaff
            jb 21h.5,lombaff
            mov R1,#Line6
            lcall Tangkap
            setb 21h.5
            ljmp lombaff
lomba66:    jb 22h.5,lombaff
            dec 48h
            ;djnz 48h,lombaff
            setb 22h.5

```

```

Lombaff:    jb P1.6,lombagg
            mov a,4ch
            cjne a,#00h,lomba77
            jb 22h.6,lombagg
            jb 21h.6,lombagg
            mov R1,#Line7
            lcall Tangkap
            setb 21h.6
            ljmp lombagg
lomba77:    jb 22h.6,lombagg
            dec 4ch
            ;djnz 4ch,lombagg
            setb 22h.6

lombagg:    jb P1.7,lombahh
            mov a,50h
            cjne a,#00h,lomba88
            jb 22h.7,lombahh
            jb 21h.7,lombahh
            mov R1,#Line8
            lcall Tangkap
            setb 21h.7
            ljmp lombahh
lomba88:    jb 22h.7,lombahh
            dec 50h
            ;djnz 50h,lombahh
            setb 22h.7
            ljmp Lomba

lombahh:    mov a,22h
            cjne a,#0ffh,lombajj
            lcall del05sec
            lcall del05sec
            lcall del05sec
            lcall del05sec
            mov 22h,#00h
lombajj:    mov a,21h
            cjne a,#0ffh,Lombaii
            ljmp lombaxx
Lombaii:    ljmp Lomba
lombaxx:    setb StrL
            pop 01h
            ret

;-----Tangkap Waktu
Tangkap:   clr EX0
            ;mov dptr,#4002h
            ;movx a,@dptr
            mov @r1,25h
            inc r1
            ;mov dptr,#4000h
            ;movx a,@dptr
            mov @r1,26h
            inc r1
            mov 13h,11h

```

```

        mov 14h,12h
        mov a,13h
        cjne a,#01h,tangkbb
        mov 13h,#05h
tangkbb:  mov a,14h
        swap a
        clr c
        add a,13h
        mov @r1,a
        setb EX0
        ret

;-----HASIL LOMBA
HasilT00: db ' Catatan Waktu ',00h
HasilT11: db 'Lx= xx:xx:xx,xx ',00h

Hasil:   lcall clearLCD
        mov dptr,#HasilT00
        mov a,#line_1
        lcall dispMSG
        mov dptr,#HasilT11
        mov a,#line_2
        lcall dispMSG
        lcall del05sec
        lcall del05sec
        lcall del05sec
        lcall del05sec
        lcall del05sec
        lcall del05sec
        lcall del05sec

        setb ISDd
        mov P0,#0dh ;L1
        clr ISDd
        lcall delay_1
        clr PLAY
        lcall del025sec
        setb PLAY
        mov 15h,#31h
        mov 16h,#00h
        mov 17h,31h
        mov 18h,32h
        mov 19h,33h
        lcall rrubah
        lcall tampil
        lcall del05sec
        lcall del05sec
        lcall del05sec
        lcall del05sec
        lcall del05sec
        lcall del05sec
        lcall del05sec

        setb ISDd
        mov P0,#1ch ;L2
        clr ISDd

```

```
lcall delay_1
clr PLAY
lcall del025sec
setb PLAY
mov 15h,#32h
mov 16h,#00h
mov 17h,35h
mov 18h,36h
mov 19h,37h
lcall rrubah
lcall tampil
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
```

```
setb ISDd
mov P0,#2ah ;L3
clr ISDd
lcall delay_1
clr PLAY
lcall del025sec
setb PLAY
mov 15h,#33h
mov 16h,#00h
mov 17h,39h
mov 18h,3ah
mov 19h,3bh
lcall rrubah
lcall tampil
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
```

```
setb ISDd
mov P0,#37h ;L4
clr ISDd
lcall delay_1
clr PLAY
lcall del025sec
setb PLAY
mov 15h,#34h
mov 16h,#00h
mov 17h,3dh
mov 18h,3eh
mov 19h,3fh
lcall rrubah
lcall tampil
lcall del05sec
```

```
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
```

```
setb ISDd
mov P0,#44h ;L5
clr ISDd
lcall delay_1
clr PLAY
lcall del025sec
setb PLAY
mov 15h,#35h
mov 16h,#00h
mov 17h,41h
mov 18h,42h
mov 19h,43h
lcall rrubah
lcall tampil
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
```

```
setb ISDd
mov P0,#4fh ;L6
clr ISDd
lcall delay_1
clr PLAY
lcall del025sec
setb PLAY
mov 15h,#36h
mov 16h,#00h
mov 17h,45h
mov 18h,46h
mov 19h,47h
lcall rrubah
lcall tampil
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
```

```
setb ISDd
mov P0,#5ah ;L7
clr ISDd
lcall delay_1
clr PLAY
```

```
lcall del025sec
setb PLAY
mov 15h,#37h
mov 16h,#00h
mov 17h,49h
mov 18h,4ah
mov 19h,4bh
lcall rrubah
lcall tampil
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
```

```
setb ISDd
mov P0,#67h ;L8
clr ISDd
lcall delay_1
clr PLAY
lcall del025sec
setb PLAY
mov 15h,#38h
mov 16h,#00h
mov 17h,4dh
mov 18h,4eh
mov 19h,4fh
lcall rrubah
lcall tampil
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
lcall del05sec
ret
```

```
Tampil:  mov a,#0c1h
          lcall setLCD
          mov a,15h
          lcall senddata
          mov a,#0c4h
          lcall setLCD
          mov a,#30h
          lcall senddata
          lcall senddata
          mov a,#':'
          lcall senddata
          mov a,17h
          anl a,#0f0h
          swap a
          add a,#30h
          lcall senddata
```

```

        mov a,17h
        anl a,#0fh
        add a,#30h
        lcall senddata
        mov a,#':'
        lcall senddata
        mov a,18h
        anl a,#0f0h
        swap a
        add a,#30h
        lcall senddata
        mov a,18h
        anl a,#0fh
        add a,#30h
        lcall senddata
        mov a,#','
        lcall senddata
        mov a,19h
        anl a,#0f0h
        swap a
        add a,#30h
        lcall senddata
        mov a,19h
        anl a,#0fh
        add a,#30h
        lcall senddata
        ret

rrubah:  mov a,18h
        lcall HextoDec
        mov 18h,a
        ret

HextoDec:  clr c
        mov b,#0ah
        div ab
        swap a
        add a,b
        ret

; PROGRAM MULAI
Mulai:    mov SP,#59h
        lcall del05sec
        ;lcall initRTC

Mulai1:   lcall initSoft
        lcall initLCD
        ;lcall SetRTC
        lcall Intro
        lcall del05sec
        lcall del05sec

mulai2:   clr EX0
        lcall clearLCD
        mov dptr,#Ready
        mov a,#Line_2
        lcall dispMSG
        setb EX0

```

```

        lcall del01sec
        jb Menu,$
        clr EX0
        lcall clearLCD
        mov dptr,#Menu1      ;50m
        mov a,#Line_2
        lcall dispMSG
        setb EX0
mulaib: lcall del01sec
        jb Menu,mulaia
        ljmp mulaic
mulaia: jb Strr,mulaib
        mov Kelas,#00h      ;kls 50m
        ljmp mulaiLho
mulaic: lcall del01sec
        mov dptr,#Menu2      ;100m
        mov a,#Line_2
        lcall dispMSG
mulaid: lcall del01sec
        jb Menu,mulaie
        ljmp mulaif
mulaie: jb Strr,mulaid
        mov Kelas,#01h      ;kls 100m
        ljmp mulaiLho
mulaif: lcall del01sec
        mov dptr,#Menu3      ;200m
        mov a,#Line_2
        lcall dispMSG
mulaig: lcall del01sec
        jb Menu,mulaih
        ljmp mulai2
mulaih: jb Strr,mulaig
        mov Kelas,#03h      ;kls 200m

MulaiLho: lcall del05sec
        lcall del05sec
        jb STRr,$
        lcall Renang
        clr EX0
        clr EA
mulaii: lcall Hasil
mulaij: jb Menu,mulaik
        ljmp Mulaii
mulaik: jb Strr,mulaij
        lcall Initsoft
        ljmp Mulai2

Ready:   db '    READY    ',00h
Menu1:   db 'Kelas 50 meter ',00h
Menu2:   db 'Kelas 100 meter ',00h
Menu3:   db 'Kelas 200 meter ',00h

```

End

Data dan alamat dalam menginisialisasi program ke mikrokontroler:

Untuk menginisialisai 1/100 detik (sesuai dengan peraturan penilaian dalam suatu kejuaraan renang), atau sebesar 10ms.

TL0	TH0
250d	40d
↓	↓
FAh (FF-FA)	28h (FF-28)
05h	D7h

Kemudian alamat 05 dan D7 tersebut diinisialisasikan ke program mikrokontroler

LAMPIRAN L-4.1
MIKROKONTROLER
AT89C51
(DATA SHEET)

LAMPIRAN L-4.2

RTC DS12887

(DATA SHEET)

LAMPIRAN L-4.3

ISD 1400

(DATA SHEET)

LAMPIRAN L-4.4

FINA *SWIMMING TIMING*

RULE

LED (*Light Emitting Diode*)

(DATA SHEET)

FINA SWIMMING RULE

Timing

New Rules 2002-2005

SW 1 The operation of Automatic Officiating Equipment shall be under the supervision of appointed officials. Times recorded by Automatic Equipment shall be used to determine the winner, all placing and the time applicable to each lane. The placing and times so determined shall have precedence over the decisions of timekeepers. In the event that a break-down of the Automatic Equipment occurs or that it is clearly indicated that there has been a failure of the Equipment, or that a swimmer has failed to activate the Equipment, the recordings of the timekeepers shall be official (See SW 13.3).

SW 2 When Automatic Equipment is used, the results shall be recorded only to 1/100 of a second. When timing to 1/1000 of a second is available, the third digit shall not be recorded or used to determine time or placement. In the event of equal times, all swimmers who have recorded the same time at 1/100 of a second shall be accorded the same placing. Times displayed on the electronic scoreboard should show only to 1/100 of a second.

SW 3 Any timing device that is terminated by an official shall be considered a watch. Such manual times must be taken by three timekeepers appointed or approved by the Member in the country concerned. All watches shall be certified as accurate to the satisfaction of the governing body concerned. Manual timing shall be registered to 1/100 of a second. Where no Automatic Equipment is used, official manual times shall be determined as follows:

- a. If two of the three watches record the same time and the third disagrees, the two identical times shall be the official time.
- b. If all three watches disagree, the watch recording the intermediate time shall be the official time.
- c. With only two (2) out of three (3) watches working the average time shall be the official time.

Light Emitting Diodes (LEDs)

Calculating an LED resistor value

An LED must have a resistor connected in series to limit the current through the LED, otherwise it will burn out almost instantly.

The resistor value, R is given by:

$$R = (V_S - V_L) / I$$

V_S = supply voltage

V_L = LED voltage (usually 2V, but 4V for blue and white LEDs)

I = LED current (e.g. 20mA), this must be less than the maximum permitted

If the calculated value is not available choose the nearest standard resistor value which is **greater**, so that the current will be a little less than you chose. In fact you may wish to choose a greater resistor value to reduce the current (to increase battery life for example) but this will make the LED less bright.

Working out the LED resistor formula using Ohm's law

Ohm's law says that the resistance of the resistor,

$R = V/I$, where:

V = Voltage across the resistor (= $V_S - V_L$ in this case)

I = The current through the resistor

So $R = (V_S - V_L) / I$