

**PEMBUATAN PROTOTYPE PENGHITUNG BIS MASUK
BERBASIS MIKROKONTROLLER AT89S51**



Disusun Oleh :

Febriaji Primadeni

NIM. M3306046

**PROGRAM DIPLOMA III ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SEBELAS MARET
SURAKARTA**

2010

HALAMAN PENGESAHAN
PEMBUATAN PROTOTYPE PENGHITUNG BIS MASUK
BERBASIS MIKROKONTROLLER AT89S51

Disusun Oleh :

FEBRIAJI PRIMADENI

NIM. M3306046

Dibimbing oleh :

Pembimbing Utama

Mohtar Yuniarto, S.Si, M.Si

NIP. 19800630 200501 1 001

Tugas Akhir ini telah diterima dan disahkan oleh dewan penguji Tugas Akhir
Program Diploma III Ilmu Komputer
pada hari Kamis tanggal 11 Februari 2010

Dewan Penguji :

1. Penguji 1 **Mohtar Yuniarto, S.Si, M.Si** ()
)

NIP. 19700727 199702 1 001

2. Penguji 2 **Rudi Hartono S.Si** ()
)

NIP. 0626128402

3. Penguji 3 **Retno Wulandari S.Si** ()
)

NIP. 0604128402

Disahkan Oleh :

Dekan
Fakultas MIPA UNS

Ketua Program Studi
DIII Ilmu Komputer UNS

Prof. Drs. Sutarno, M.Sc, Ph.D

NIP.19600809 198612 1 001

Drs. YS. Palgunadi, M.Sc

NIP.19560407 198303 1 004

MOTTO

*Optimis ditambah enjoy menghasilkan rasa tidak takut
(Penulis)*

*Sesuatu yang besar tidak terjadi tanpa sesuatu yang kecil
(Penulis)*

*Keseriusan membuahkan hasil maksimal, ketidakseriusan
membuahkan hasil minimal
(Penulis)*

*Gagal merencanakan berarti merencanakan kegagalan.
(Penulis)*

PERSEMBAHAN

My Truly Saviour Jesus Christ

Keluarga kecilku Ayahku Suti Hantoro, Ibuku Sri Winarti sebagai saluran berkat buatku, Kakakku Bangkit Mahendra dan Adek-adekku Febriana Primadesi dan Yohana Nugrahaning Primadewi yang memberi keceriaan hari-hariku.

My Brother n Sister Sragen-Solo, Ary, Angga, Mas Budi, Mas Bayu yang berada di jauh, Heppy yang memberiku ngeprint gratis, Anita, Sekar, Cita, Mbak Niken, we will grow and multiple..

Teman-teman TeKomp 2006

Team DotA TeKomp 2006 Fajar Hero, Singgih Jontor (vacuum), Aat D Ace, Yayan Copal (vacuum), Agung Dwi Cupu (new user), Dedi (new user)

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan YME, dengan segala rahmat dan karuniaNya akhirnya pembuatan laporan Tugas Akhir dengan judul “PEMBUATAN PROTOTYPE PENGHITUNG BIS MASUK BERBASIS MIKROKONTROLER AT89S51” ini dapat diselesaikan.

Penulis menyadari adanya kelemahan dan keterbatasan yang penulis miliki, sehingga dalam penyelesaian laporan ini banyak dibantu dan didukung oleh berbagai pihak. Untuk itu sampaikan terima kasih kepada :

1. Bapak Drs. YS Palgunadi, M.Sc selaku ketua Program Studi D3 Ilmu Komputer UNS.
2. Bapak Muhtar Yuniato, M.Si selaku dosen Pembimbing akademik atas bimbingannya.
3. Teman-teman Teknik Komputer 2006 yang telah memberikan semangat kepada penulis.
4. Keluarga ku yang telah memberikan dukungan kepada penulis.
5. Pihak-pihak lain yang tak dapat penyusun sebutkan satu persatu yang telah membantu penyusunan dalam menyelesaikan laporan ini.

Menyadari bahwa suatu karya di bidang apapun tidak terlepas dari kekurangan, maka dihimbau kepada pembaca untuk memberikan saran penyempurnaan. Dengan jiwa dan moral keilmuan, buku pedoman ini akan terus disempurnakan sehinggampu menampung aspirasi bersama.

Surakarta, 14 Desember 2009

Penyusun

DAFTAR ISI

	Halaman
KATA PENGANTAR	ii
DAFTAR ISI	iii
DAFTAR TABEL.....	v
DAFTAR GAMBAR	vi
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Perumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan dan Manfaat	2
1.5. Metodologi Penelitian	2
1.6. Sistematika Penulisan	3
BAB II LANDASAN TEORI.....	5
2.1. Dasar Mikrokontroler.....	5
2.2. Mikrokontroler AT8S51.....	5
2.3. Arsitektur Mikrokontroler AT89S51	7
2.4. Organisasi Memori	12
2.5. Memori Program.....	13
2.6. Memori Data	13
2.7. Instruksi Mikrokontroler.....	15
2.8. Metode Pengalamatan.....	18
2.9. Sistem Interupsi	21
2.10. Seven Semen.....	22
2.11. Sensor Ultrasonik.....	24
BAB III DESAIN DAN PERANCANGAN.....	25
3.1. Identifikasi Kebutuhan	25
3.2. Analisis Kebutuhan	25
3.3. Pembuatan Alat	26
3.4. Perancangan Alat Keras	26
3.4.1. Bagian Pemancar Sensor Ultrasonik.....	29

3.4.2. Mikrokontroler AT89S51	30
3.4.3. Display	30
3.4.4. Rangkaian Catu Daya.....	31
3.5. Diagram Blok	32
3.6. Perancangan Software	33
3.6.1 Software Pendukung	33
3.6.2 Flowchart	34
BAB IV PENGUJIAN DAN ANALISIS	36
4.1. Pengujian Alat.....	36
4.1.1 Pengujian Sensor Ultrasonik.....	36
4.1.2 Pengujian Mikrokontroler	37
4.2. Pengujian Secara Keseluruhan.....	39
BAB V PENUTUP.....	42
5.1. Kesimpulan.....	42
5.2. Saran	42
DAFTAR PUSTAKA	43
LAMPIRAN	44

DAFTAR TABEL

	halaman
Tabel 2.1 Kapasitas Memory Mikrokontroler seri AT89X	6
Tabel 2.2 Mode Kerja Timer/ Counter	20
Tabel 4.1 Hasil Uji coba Mikrokontroller	38
Tabel 4.2 Hasil Uji Sistem Keseluruhan	39

DAFTAR GAMBAR

	halaman
Gambar 2.1 Mikrokontroler AT89S51	7
Gambar 2.2 Diagram blok AT89S51	8
Gambar 2.3 Struktur Memory Program Dan Memory Data	12
Gambar 2.4 Struktur Memori Program	13
Gambar 2.5 Struktur Memori Data Internal	14
Gambar 2.6 128 Byte Rendah RAM Internal	14
Gambar 2.7 128 Atas RAM Internal Dan Ruang SFR	15
Gambar 2.8 Seven Segment	23
Gambar 2.9 Sensor Ultrasonik	24
Gambar 3.1 Desain Alat Penghitung Jumlah Bis	28
Gambar 3.2 Bagian Pemancar Sensor Ultrasonik	29
Gambar 3.3 Bagian Penerima Sensor Ultrasonik	29
Gambar 3.4 Bagian Mikrokontroller	30
Gambar 3.5 Display Seven Segmen 3 Digit	30
Gambar 3.6 Bagian Catu Daya	31
Gambar 3.7 Diagram Blok	32
Gambar 3.8 Flow Chart	35
Gambar 4.1 Rangkaian Uji Sensor Ultrasonik	36
Gambar 4.2 Rangkaian Uji Mikrokontroller	37
Gambar 4.3 Alat sesudah dijalankan	41
Gambar 4.4 Prototipe Penghitung Bis	41

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Dalam masa sekarang ini semua hal dituntut mencari solusi yang yang mampu menyelesaikan permasalahan-permasalahan secara tepat dan cepat. Seiring perkembangan waktu, perkembangan teknologi terus bergerak cepat sehingga dihasilkan berbagai piranti yang mampu memberikan solusi terhadap kesulitan-kesulitan yang dihadapi manusia. Seperti halnya kegiatan-kegiatan yang terjadi di terminal termasuk penghitungan bis yang melihat kondisi manusia yang tidak memungkinkan melakukan pekerjaan tersebut 24 jam sehari.

Dewasa ini perkembangan teknologi elektronika digital telah berkembang dengan pesatnya, diberbagai bidang, baik sektor industri maupun peralatan kedokteran. Penggunaan *mikrokontroller* sangat luas, mulai dari permainan elektronika sampai dengan pesawat luar angkasa. Teknologi ini dirasakan masih cukup ampuh dalam menyelesaikan permasalahan elektronika digital, memiliki kemudahan dalam hal pemrograman untuk mengendalikan input-output dan biaya yang murah. Lahirnya *mikrokontroler* telah menimbulkan suatu revolusi dalam membuat sistem berbasis digital yang kompleks.

Hal ini juga berlaku dalam kehidupan sehari-hari, termasuk dalam kegiatan di terminal, untuk menunjang kegiatan di terminal supaya menjadi lebih efektif dan efisien terutama dalam penghitungan jumlah bis masuk ke dalam terminal yang selama ini lebih sering dilakukan penghitungan manual oleh petugas terminal maka penulis mencoba membuat prototipe alat penghitung bis masuk tersebut. Dan prototipe penghitung bis ini menggunakan *sensor ultrasonic* dikarenakan sensor tersebut lebih peka dan untuk mendapatkan hasil bisa lebih akurat daripada sensor yang lain walaupun harganya lebih mahal tetapi keakurasian data yang didapat lebih baik.

Untuk mengatasi hal tersebut mendorong penulis untuk merancang sistem penghitung jumlah bis masuk berbasis *Mikrokontroller*. Dimana alat tersebut secara otomatis menampilkan jumlah bis masuk ke dalam terminal sehingga memudahkan kegiatan pada terminal tersebut.

1.2 Perumusan Masalah

Sesuai dengan latar belakang judul di atas, maka yang menjadi permasalahan dalam Tugas Akhir ini adalah :

1. Bagaimana membuat *prototype* penghitung jumlah bis masuk ke terminal.
2. Bagaimana mengefektifkan *prototype* tersebut hanya untuk menghitung bis.

1.3 Batasan Masalah

Untuk mendapatkan informasi materi serta agar pembahasan tidak menyimpang dari tujuan maka dilakukan pembatasan masalah sebagai berikut.

1. Permasalahan dari proyek akhir ini dibatasi hanya untuk menghitung jumlah bis masuk.
2. *Mikrokontroller* yang digunakan adalah seri AT89S51 sebagai kontroler utama untuk sistem ini.
3. Bahasa pemrograman yang digunakan adalah bahasa pemrograman Assembler.
4. Pemanfaatan *sensor* ultrasonik sebagai sensor untuk mendeteksi sebuah benda (bis).
5. Objek yang terdeteksi oleh sensor dari jarak 50 cm kebawah.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Maksud dan tujuan pembuatan tugas akhir ini adalah.

1. Membuat *prototype* penghitung bis masuk ke dalam terminal.
2. Memudahkan kegiatan di terminal terkhusus penghitungan bis masuk yang selama ini masih dilakukan manual.

1.4.2 Manfaat

1. Nilai Ekonomis

Proyek akhir ini akan menghasilkan sebuah produk yaitu sebuah *prototype* untuk menghitung bis masuk yang dapat dikembangkan untuk penerapan pada bidang lain (parkir, antrian masuk, dll).

2. Nilai Strategis

Menambah wawasan pengetahuan terutama dalam bidang *Mikrokontroler* bagi mahasiswa.

1.5 Metodologi Penelitian

Untuk menyelesaikan proyek akhir pada bagian mikrokontroler ini, dilakukan langkah-langkah sebagai berikut.

1. Mempelajari prinsip kerja dasar dari sistem *mikrokontroler* AT89S51, Sensor ultrasonik dan sistem perangkat keras.
2. Menguji coba tiap subsistem perangkat keras, dilanjutkan dengan integrasi antar subsistem dan pengujian hasil integrasi, diteruskan dengan integrasi dan pengujian hasil integrasi antar sistem perangkat lunak dan perangkat keras.
3. Menganalisa dan menyimpulkan hasil-hasil simulasi, serta memberi saran bila proyek akhir ini dikembangkan lebih lanjut.
4. Menyusun buku laporan proyek akhir.

1.6 Sistematika Penulisan

Sistematika yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut.

Bab I Pendahuluan

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, tujuan dan manfaat penelitian, metodologi penelitian dan sistematika penulisan.

Bab II Landasan Teori

Berisi tentang dasar-dasar teori mengenai peralatan, tinjauan pustaka yang diperlukan untuk perancangan alat.

Bab III **Desain dan Perancangan**

Berisi mengenai tentang data-data yang diperlukan dalam membuat peralatan tersebut.

Bab IV **Hasil dan Pembahasan**

Berisi hasil perancangan alat dan pembahasan kinerja alat.

Bab V **Kesimpulan dan Saran**

Berisi tentang kesimpulan dan saran dari hasil penelitian.

BAB II LANDASAN TEORI

2.1 Dasar *Mikrokontroler*

Membuat projek atau tugas akhir yang menggunakan *Mikroprosesor* 8086/8088 tampaknya saat ini cukup sulit karena dibutuhkan biaya yang besar serta diperlukannya EPROM *Programmer*. Cara lain yang lebih gampang dan murah ialah mengembangkan aplikasi menggunakan kit *mikrokontroler* (*main board*) yang sudah ada di pasaran. Kit tersebut umumnya terdiri dari *mikrokontroler*, memori serta *interface* untuk koneksi ke lcd, ke PPI atau ke port serial komputer. Tinggal membuat program serta menyambungkan dengan board atau komponen tambahan saja. Membangun aplikasi elektronika berbasis *embedded system* merupakan topik yang sangat hangat saat ini. Apalagi *mikrokontroler* tertentu sudah mendukung aplikasi yang terhubung ke jaringan computer (*network microcontroller*).

Ada perbedaan yang cukup penting antara *Mikroprosesor* dan *Mikrokontroler*. Jika *Mikroprosesor* merupakan CPU (*Central Processing Unit*) tanpa memori dan I/O pendukung dari sebuah komputer, maka *Mikrokontroler* umumnya terdiri dari CPU, Memori , I/O tertentu dan unit pendukung, misalnya *Analog to Digital Converter* (ADC) yang sudah terintegrasi di dalam *mikrokontroler* tersebut. Kelebihan utama dari *Mikrokontroler* ialah telah tersedianya RAM dan peralatan I/O Pendukung sehingga ukuran board *mikrokontroler* menjadi sangat ringkas. Terdapat berbagai jenis *mikrokontroler* dari berbagai vendor yang digunakan secara luas di dunia. Diantaranya yang terkenal ialah dari Intel, Maxim, Motorola , dan ATMEL. Beberapa seri *mikrokontroler* yang digunakan secara luas ialah 8031, 68HC11, 6502 , 2051 dan 89S51 (www.toko-elektronika.com,2009).

Mikrokontroller merupakan suatu chip IC yang dapat berfungsi sebagai suatu pengontrol rangkaian elektronika yang umumnya pada chip IC tersebut dapat menyimpan suatu program didalamnya. Perbedaan antara *Mikroprosesor* dengan *mikrokontroler* sangatlah jelas. *Mikroprosesor* hanya

terdiri dari CPU (*Central Processing Unit*) tanpa memori dan I/O pendukung sebuah komputer, sedangkan pada *mikrokontroler* terdiri dari CPU, memori, I/O tertentu, serta memiliki pendukung tertentu yang telah terintegrasi didalamnya seperti Rangkaian *Clock, Counter, Timer, ADC (Analog to Digital Converter)* juga ROM (*Read Only Memory*) dan RAM (*Random Access Memory*).

Tabel 2.1 Kapasitas Memory *Mikrokontroler* seri AT89X

Type	RAM	Flash Memory	EEPROM
AT89C51/ AT89S51	8 X 128 <i>byte</i>	4 Kbyte	Tidak
AT89C52/ AT89S52	8 X 256 <i>byte</i>	8 Kbyte	Tidak
AT89C55	8 X 256 <i>byte</i>	20 Kbyte	Tidak
AT89S53	8 X 256 <i>byte</i>	12 Kbyte	Tidak
AT89S8252	8 X 256 <i>byte</i>	8 Kbyte	2 Kbyte

(Sumber : <http://www.toko-elektronika.com/tutorial/uc1.html>, 2009)

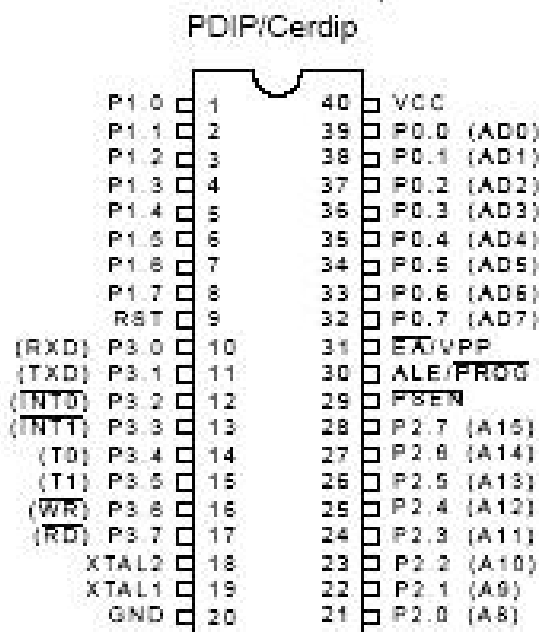
2.2 *Mikrokontroler* AT89S51

Mikrokontroler AT89S51 ialah mikrokomputer CMOS 8 bit dengan 4KB *Flash Programmable* dan *Erasable Read Only Memory (PEROM)*. *Mikrokontroler* berteknologi memori *non volatile* kerapatan tinggi dari Atmel ini kompatibel dengan *mikrokontroler* standar industri MCS-51 (seperti *mikrokontroler* 8031 yang terkenal dan banyak digunakan beberapa waktu lalu) baik pin kaki IC maupun set instruksinya serta harganya yang cukup murah.

AT89S51 mempunyai memori yang terdiri dari RAM internal sebesar 128 *byte* dengan alamat 00H-7FH dapat diakses menggunakan RAM address *register*. RAM Internal ini terdiri dari *Register Banks* dengan 8 buah *register* (R0-R7).Memori lain yaitu 21 buah *Special Function Register* dimulai dari alamat 80H-FFH. RAM ini beda lokasi dengan *Flash PEROM* dengan alamat 000H -7FFH.

Komponen utama yang digunakan dalam rangkaian sistem penghitung jumlah bis masuk adalah sebuah *mikrokontroller* AT89S51. Sebuah *mikrokontroller* tersusun atas beberapa bagian penting seperti *Central Prosesor Unit* (CPU), *Program Counter* (PC), *Stack Pointer* (SP), *Register*, *Untai Pewaktu*, *Read Only Memory* (ROM), *Random Access Memory* (RAM) dan dilengkapi dengan port input/output (PIO).

Memori Flash digunakan untuk menyimpan perintah (instruksi) berstandar MCS-51, dan fasilitas ROM yang dimiliki adalah bertipe *Reprogrammable Flash Memory* dimana ROM dapat diisi dan dihapus secara berulang-ulang.



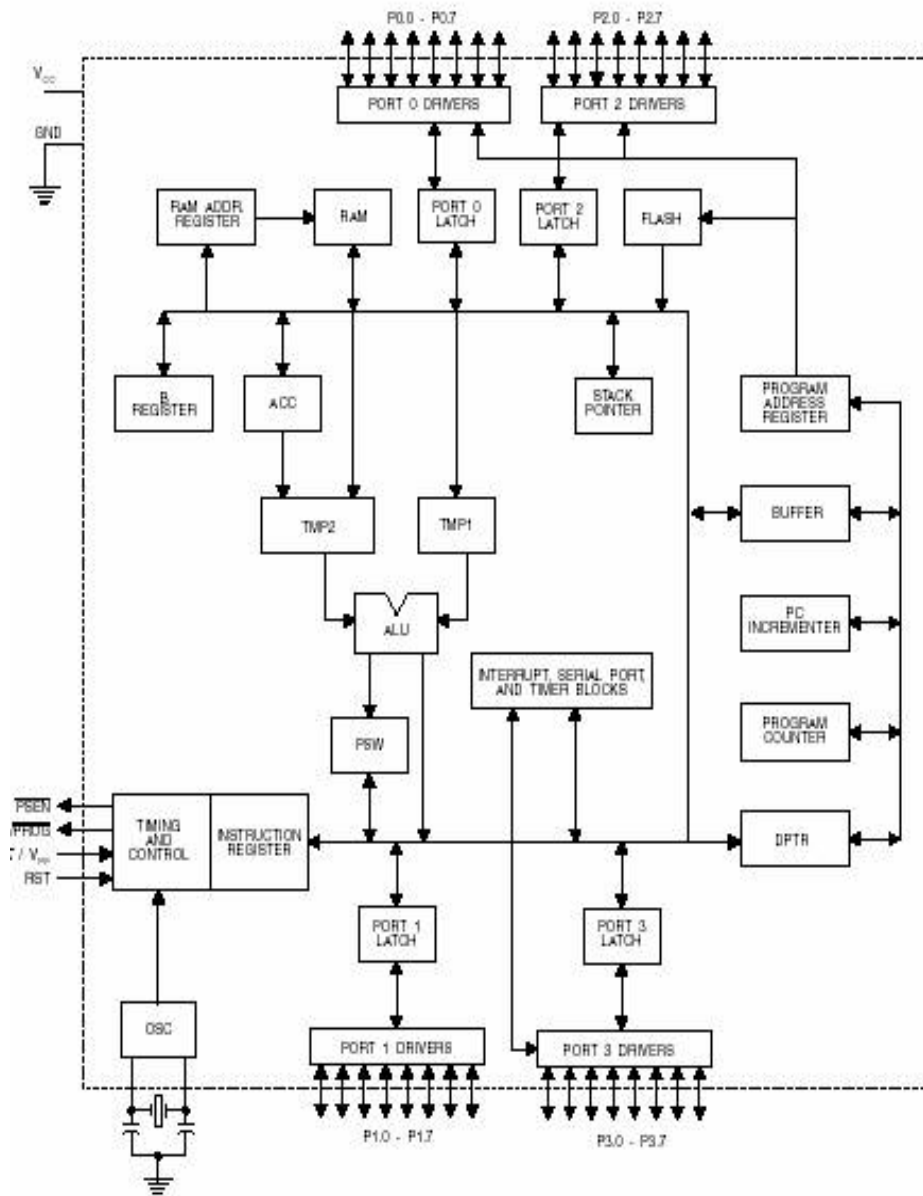
Gambar 2.1 *Mikrokontroler* AT89S51

(Sumber : <http://electronyclab.com>, 2009)

2.3 Arsitektur *Mikrokontroler* AT89S51

Mikrokontroler ini mempunyai empat port I/O, akumulator, *register*, RAM internal, stack pointer, *Arithmetic Logic Unit* (ALU), pengunci (*latch*),

dan rangkaian osilasi yang membuat *mikrokontroler* ini dapat beroperasi hanya dengan sekeping IC. Secara fisik, *mikrokontroler* AT89S51 mempunyai 40 pin, 32 pin diantaranya adalah pin untuk keperluan port masukan/keluaran. Satu port paralel terdiri dari 8 pin, dengan demikian 32 pin tersebut membentuk 4 buah port paralel, yang masing-masing dikenal dengan Port 0, Port 1, Port 2 dan Port 3.



Gambar 2.2 Diagram blok AT89S51

(Sumber : <http://www.toko-elektronika.com/tutorial/uc1.html>, 2009)

Berikut penjelasan dari masing-masing port:

a. PORT 0

Port 0 merupakan port I/O 8-bit yang bersifat dua arah, sebagai port keluaran setiap kaki mampu memberikan masukan arus kepada 8 buah TTL. Jika kaki-kaki port 0 diset 1, maka kaki-kaki tersebut dapat digunakan sebagai masukan. Port 0 dikonfigurasi sebagai bus alamat orde rendah termultipleks untuk mengakses program atau data pada memori eksternal. Selama pemrograman EPROM, port 0 mampu menerima kode *byte* dan mengeluarkan *byte* kode selama verifikasi program.

b. PORT 1

Port 1 adalah port I/O 8-bit dengan pull-up internal. *Buffer* keluaran port 1 dapat mensuplai empat buah masukan TTL. Jika kaki-kaki port 1 di set 1, maka kaki-kaki tersebut akan tertarik ke atas oleh pull-up internal, sehingga dapat digunakan sebagai masukan.

c. PORT 2

Port 2 juga termasuk port I/O 8-bit dengan *pull up* internal seperti halnya Port 1. *Buffer* keluaran port 2 dapat mensuplai empat buah masukan TTL. Jika kaki-kaki port 2 di set 1, maka kaki-kaki tersebut akan tertarik oleh pull-up internal, sehingga dapat digunakan sebagai masukan. Port 2 berfungsi sebagai alamat orde tinggi selama proses pengambilan intruksi dari memori eksternal dan selama mengakses memori eksternal.

d. PORT 3

Port 3 juga termasuk port I/O 8-bit dengan *pull up* internal seperti halnya Port 1, port 2. *Buffer* keluaran port 3 dapat mensuplai empat buah masukan TTL. Jika kaki-kaki port 3 di set 1, maka kaki-kaki tersebut akan tertarik oleh *pull up* internal.

Berikut penjelasan dari masing-masing pin:

a. Pin 1 sampai 8

Berfungsi sebagai: P1.0 _ P1.7 Pin 1 sampai 8 merupakan saluran I/O 8 bit yang bersifat dua arah, dengan internal pull-up yang dapat digunakan untuk berbagai keperluan seperti mengendalikan 4 input TTL. Port ini juga digunakan sebagai saluran alamat saat pemrograman dan verifikasi. Pada pin 6, 7, 8 terdapat port pin yang digunakan pada saat *download* program.

b. Pin 9

Berfungsi sebagai : RST

Pin 9 Merupakan masukan reset bagi *mikrokontroler*. Reset akan aktif dengan memberikan input *high* selama 2 *cycle*.

c. Pin 10 sampai 17

Berfungsi sebagai : P3.0 _ P3.7

Pin 10 sampai 17 merupakan saluran I/O 8 bit dua arah dengan internal pullup. Di samping sebagai saluran I/O, port ini memiliki fungsi pengganti. Bila fungsi pengganti tidak dipakai maka dapat digunakan sebagai port paralel 8 bit serbaguna. Selain itu, sebagian Port 3 dapat berfungsi sebagai sinyal kontrol saat proses pemrograman dan verifikasi.

d. Pin 18

Berfungsi sebagai : XTAL2

Pin 18 merupakan keluaran dari rangkaian osilasi *mikrokontroler*.

e. □ Pin 19

Berfungsi sebagai : XTAL1

Pin 19 merupakan masukan untuk rangkaian osilasi *mikrokontroler*.

f. Pin 20

Berfungsi sebagai : GND

Pin 20 merupakan ground dari sumber tegangan.

g. Pin 21 sampai 28

Berfungsi sebagai : P2.0 _ P2.7

Pin 21 sampai 28 merupakan saluran I/O 8 bit dua arah dengan internal *pull up*. Saat pengambilan data dari program memori *external* atau selama pengaksesan data memori *external* yang menggunakan alamat 16 bit, Port 2 berfungsi sebagai saluran alamat tinggi (A8 – A15). Akan tetapi, saat mengakses data memori *external* yang menggunakan alamat 8 bit, Port 2 mengeluarkan isi P2 pada *Special Function Register*.

h. Pin 29

Berfungsi sebagai : PSEN

Pin ini berfungsi pada saat mengeksekusi program yang terletak pada memori *external*. *Program Strobe Enable* (PSEN) akan aktif dua kali setiap *cycle*.

i. Pin 30

Berfungsi sebagai : ALE/PROG

Pin ini dapat berfungsi sebagai *Address Latch Enable* (ALE) yang menahan *low bytes* address pada saat mengakses memori *external*. Sedangkan pada saat *Flash Programming* (PROG) berfungsi sebagai pulsa input selama proses pemrograman.

j. Pin 31

Berfungsi sebagai : EA/VPP

Pada kondisi *low*, pin ini akan berfungsi sebagai *External Access Enable* (EA) yaitu *mikrokontroler* akan menjalankan program yang ada pada memori *external*. Jika berkondisi *high*, pin ini akan berfungsi untuk menjalankan program yang ada pada memori internal. Pin ini juga berfungsi sebagai masukan tegangan pemrograman selama proses pemrograman.

k. Pin 32 sampai 39

Berfungsi sebagai : D7 _ D0 (A7 _A0)

Pin 32 sampai 39 ialah Port 0 yang merupakan saluran I/O 8 bit *open collector* dan dapat juga digunakan sebagai multipleks bus alamat rendah dan bus data selama adanya akses ke memori program *external*

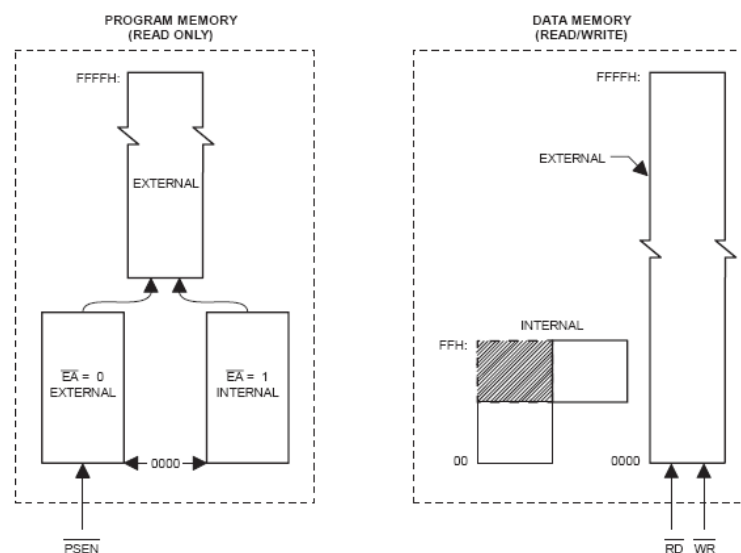
1. Pin 40

Berfungsi sebagai : VCC

Pin 40 merupakan masukan sumber tegangan positif bagi *mikrokontroler* (Putra, 2002)

2.4 Organisasi Memori

Mikrokontroler AT89S51 memiliki ruang alamat memori data dan program yang terpisah. Dengan pemisahan memori program dan data tersebut mengizinkan memori data diakses dengan alamat 8-bit, sehingga dapat dengan cepat dan mudah untuk disimpan serta dimanipulasi oleh CPU 8-bit. Meskipun demikian alamat memori data 16-bit bisa juga dihasilkan melalui *register* DPTR. (Putra, 2002)



Gambar 2.3 Struktur Memory Program Dan Memory Data

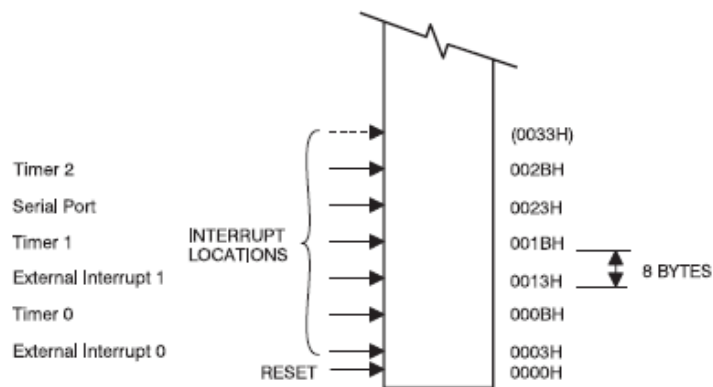
(Sumber : <http://www.atmel.com>, 2009)

Memori program hanya dapat dibaca saja. Ada 64 KByte memori program yang dapat diakses secara langsung. Tanda untuk akses memori program eksternal melalui $\overline{\text{PSEN}}$ (*Program Store Enable*).

Memori data menempati ruang alamat terpisah dengan memori program. Memori eksternal dapat diakses secara langsung hingga 64 K Byte dalam ruang memori data eksternal. CPU akan memberikan sinyal baca dan tulis $\overline{\text{RD}}$ dan $\overline{\text{WR}}$ selama pengaksesan memori data eksternal. (Putra, 2002)

2.5 Memori Program

Memori program dimulai dari lokasi 0000h. setelah direset, CPU memulai program dari alamat tersebut. Alamat-alamat dari memori program paling bawah berada dalam Flash On Chip maupun memori eksternal. Sehingga untuk akses memori internal maupun eksternal memerlukan pengkabelan pada pin \overline{EA} (*External Access*). \overline{EA} dihubungkan ke Vcc untuk akses memori internal, atau dihubung ke Ground jika akan mengakses secara langsung ke memori eksternal sedang pengaksesan ke memori internal tidak dilakukan.



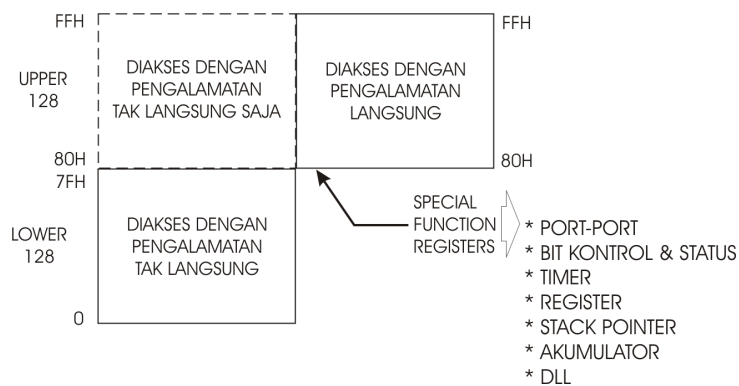
Gambar 2.4 Struktur Memori Program

(Sumber : [http:// www.mytutorialcafe.com](http://www.mytutorialcafe.com), 2009)

Terlihat pada gambar interupsi masing-masing disimpan pada lokasi yang sudah ditentukan yang menempati lokasi dengan jarak 8-byte. Sebuah interupsi menyebabkan CPU akan melompat ke interupsi yang bersangkutan, yaitu letak dari subrutin layanan interupsi tersebut. (Putra, 2002)

2.6 Memori Data

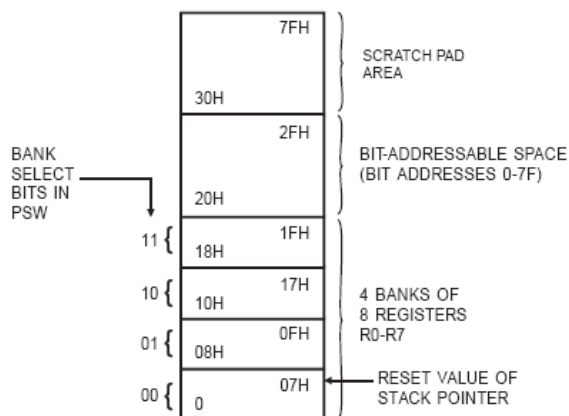
Memori data internal dibagi menjadi tiga blok, 128 bawah, 128 atas, serta SFR (*Special Function Registers*) atau *register* fungsi khusus seperti yang terlihat pada gambar 2.5.



Gambar 2.5 Struktur Memori Data Internal

(Sumber : <http://www.atmel.com>, 2009)

32 *Byte* bawah pada 128 *Byte* bawah RAM internal dikelompokkan menjadi empat bank dan delapan *register* (R0 hingga R7). Dua bit pada PSW (*Program Status Word*). 16 *Byte* bagian atas bank *register* merupakan ruang memori yang bisa teralamatkan per-bit (Bit Addressable). Seluruh *byte* yang berada pada 128 bawah dapat diakses secara langsung maupun tidak langsung.



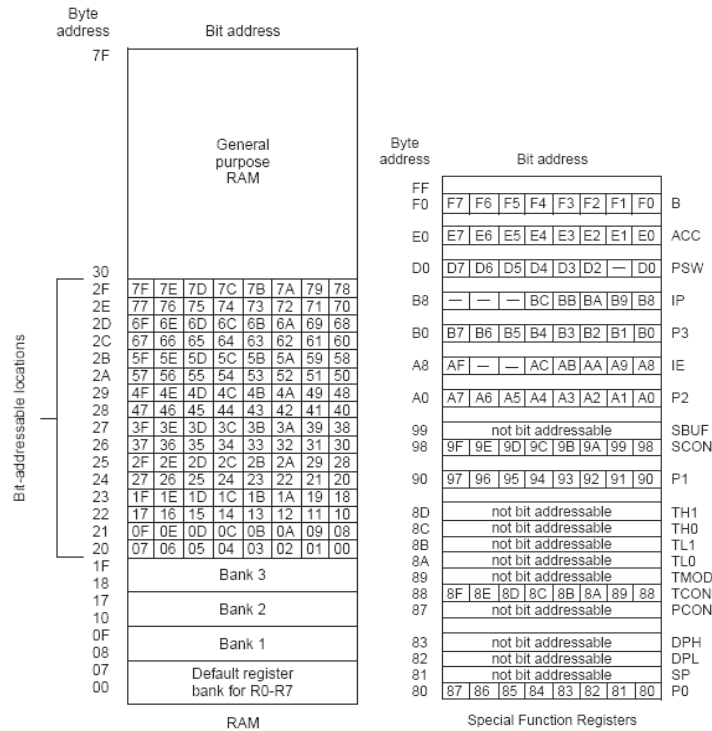
Gambar 2.6 128 *Byte* Rendah RAM Internal

(Sumber : <http://www.atmel.com>, 2009)

Sedangkan pada 128 atas hanya dapat diakses dengan mode pengalamatan tak langsung saja. Bagian ini hanya ada dalam piranti yang memiliki RAM 256 *Byte* saja.

Bagian *Register* fungsi khusus beralamatkan pada lokasi 80h hingga FFh dengan mode pangalamatan langsung. SFR ini mencakup port-port, pewaktu (*Timer*), kontrol *peripheral*, dan lainnya. Secara umum alamat SFR

sama pada semua *mikrokontroler* produksi Atmel. SFR ada yang teralamat per-bit maupun tidak, untuk yang teralamat per-bit memiliki format alamat awal yang khusus yaitu xxxx x000b atau dalam format heksadesimal x0h dan x8h, misalnya 80h (mulai dari 80h hingga 87h) untuk P0, 88h untuk TCON dan seterusnya. (Putra, 2002)



Gambar 2.7 128 Atas RAM Internal Dan Ruang SFR

(Sumber : <http://www.atmel.com>, 2009)

2.7 Instruksi Mikrokontroler

Instruksi *mikrokontroler* sangatlah banyak sekali, dikelompokkan menjadi lima kelompok yaitu : operasi aritmatika, operasi logika, transfer data, instruksi Boolean, serta instruksi pencabangan.

a. Operasi aritmatika

Operasi ini hampir sama dengan kebanyakan *Mikroprosesor* 8-bit. Terdapat penjumlahan (ADD), penjumlahan dengan carry (ADDC), pengurangan dengan borrow (SUBB), increment (INC),

decrement (DEC), serta decimal adjust accumulator (DA). Juga terdapat instruksi perkalian (MUL AB) dan pembagian (DIV AB).

Berikut contoh operasi aritmatika sekaligus penjelasannya :

ADD A, Rn : Jumlahkan A dengan *register* Rn, hasilnya disimpan di A

SUBB A, #n : Kurangkan data n dengan A sekaligus carry, hasil disimpan di A.

MUL AB : Kalikan *byte* dalam A dan B. hasilnya bit tinggi simpan di B, bit rendah di A.

DIV AB : bagi *byte* dalam A dengan B, simpan hasilnya dalam A sisanya dalam B.

b. Operasi logika

Operasi logika yang dimiliki *mikrokontroler* AT89S51 meliputi AND (ANL), OR (ORL), X-OR (XRL), *Clear* (CLR), Complement (CPL), pemutaran (RL, RLC, RR, RRC), serta SWAP A. Instruksi SWAP A berfungsi untuk menukarkan nibble atas dengan bawah pada akumulator.

Berikut contoh operasi logika sekaligus penjelasannya :

ANL A, #n : AND-kan A dengan data n, hasilnya disimpan di A

ORL A, Rn : OR-kan A dengan *register* Rn, hasilnya disimpan di A

CLR A : *Clear*-kan tiap bit di *register* A ke nilai 0

RL A : Putar *register* A satu bit ke posisi kiri

CPL : Komplemenkan tiap bit di A

SWAP A : Tukarkan nibble atas dengan nibble bawah *register* A.

c. Transfer data

Operasi dasar transfer data adalah *move*, yang ada dalam tiga versi yaitu : MOV, MOVC, MOVX. Termasuk juga dalam instruksi ini adalah PUSH, POP, XCH. MOV digunakan untuk merujuk ke RAM internal dan SFR. MOVC digunakan untuk memindah *byte* data dari

memori program ke akumulator dan dari tabel data. MOVX untuk merujuk ke memori eksternal.

Berikut contoh operasi transfer data sekaligus penjelasannya :

MOV A, #FFh : Isi A dengan data FFh

MOV A, Rn : Isi A dengan data di *register* Rn

MOV Rn, #OFh : Isi *register* Rn dengan data OFh

XCH A, Rn : Tukarkan *byte* data antara A dengan *register* Rn.

d. Instruksi Boolean

Yang termasuk dalam instruksi Boolean ini adalah SET, CLR, CPL, ANL, ORL yang diberlakukan dalam level bit. Instruksi transfer data yang bekerja pada level bit dan instruksi jump yang bersyarat nilai bit tertentu juga termasuk dalam kelompok ini.

Berikut beberapa contoh instruksi Boolean yang bekerja pada level bit, sekaligus penjelasannya :

ANL C, b : AND-kan C dengan bit b, simpan hasilnya di C

CPL b : Komplemenkan bit b

CLR b : *Clear* bit b ke nilai 0.

e. Instruksi pencabangan

Instruksi pencabangan meliputi pemanggilan sub rutin (*call* dan *return*). Lompatan juga termasuk ke dalam instruksi ini, baik lompat bersyarat maupun tanpa syarat. Termasuk dalam kelompok ini juga instruksi NOP (*No Operation*) yang menjadikan prosesor tidak mengerjakan apapun dalam satu siklus mesin.

Instruksi lompatan dalam *mikrokontroler* terdapat tiga versi dasar, yakni : SJMP (*Short Jump*) hanya mampu mundur 128 *byte* dan maju 127 *byte*, LJMP (*Long Jump*) mampu ke seluruh memori program, serta AJMP (*Absolute Jump*) yang tujuannya ditentukan dengan alamat. (Putra, 2002)

Berikut contoh instruksi pencabangan sekaligus penjelasannya :

JB P1.0, addr : Lompat ke addr jika P1.0 = 1 ke label

JC, addr : Lompat ke addr jika Carry = 1

- JZ, addr : Lompat ke addr jika akumulator = 0
- CJNE A, #n, addr : Bandingkan A dengan data n, jika tidak sama lompat ke alamat addr.
- DJNZ Rn, addr : Kurang 1 isi *register* Rn, lompat ke alamat addr jika hasilnya belum nol.

2.8 Mode Pengalamatan

Mode pengalamatan merupakan cara untuk mengakses data/ instruksi dalam program yang menunjuk pada alamat atau suatu *register* di dalam *mikrokontroler*. Beberapa macam mode pengalamatan dalam *mikrokontroler* antara lain :

a. Pengalamatan Langsung

Dalam mode pengalamatan ini, operan-operan ditentukan berdasarkan alamat 8-bit (1 *byte*) dalam suatu instruksi. Yang dapat diakses dalam mode pengalamatan ini hanya RAM dan SFR saja.

b. Pengalamatan Tak Langsung

Pengalamatan tak langsung ini, instruksi menentukan suatu *register* yang digunakan untuk menyimpan alamat operan. Baik RAM internal dan eksternal dapat diakses menggunakan pengalamatan tak langsung.

Register alamat untuk alamat-alamat 8-bit bisa menggunakan *Stack Pointer* atau R0 dan R1 dari bank *register* yang dipilih, sedangkan untuk alamat 16-bit hanya bisa menggunakan *register* pointer data 16-bit atau DPTR.

c. Instruksi *Register*

Register-register yang terdapat pada bank-bank *register* mulai dari R0 sampai R7 dapat diakses melalui instruksi yang op-kodenya mengandung 3-bit spesifikasi *register* (000 untuk R0, 001 untuk R1, hingga 111 untuk R7). Pengaksesan *register* dengan cara demikian dapat menghemat penggunaan kode instruksi karena tidak memerlukan sebuah *byte* untuk alamat. Saat instruksi tersebut dikerjakan, satu dari delapan *register* terpilih dikerjakan.

d. Instruksi *Register* Khusus

Beberapa instruksi hanya dikhususkan untuk *register-register* tertentu. Misal suatu instruksi hanya bekerja pada akumulator saja, hingga tidak memerlukan alamat *byte* untuk menunjuk ke akumulator tersebut. Instruksi yang mengacu akumulator sebagai A akan dikodekan dengan op-kode spesifik akumulator.

e. Konstanta Langsung

Nilai suatu konstanta atau data dapat menyatu langsung dengan op-kode pada memori program. Sebagai contoh instruksi MOV A, #100, akan menyimpan data 100 desimal ke akumulator. Hal yang sama juga bisa ditulis dengan MOV A, #64h.

f. Pengalamatan Terindeks

Memori program hanya bisa diakses melalui pengalamatan terindeks, mode pengalamat ini ditujukan untuk membaca tabel tengok (look up tables) atau data base yang tersimpan dalam memori program. Sebuah *register* dasar 16-bit (DPTR atau Program Counter) menunjuk ke awal atau dasar tabel dan akumulator di set dengan angka indeks tabel yang akan diakses. Alamat dari entri tabel dalam memori dibentuk dengan menjumlahkan data akumulator dengan penunjuk awal tabel. (Putra, 2002)

g. *Timer dan Counter*

Secara mendasar *Timer* dan *counter* ini terdiri dari seperangkat pencacah biner (*binary counter*) yang terhubung langsung dengan saluran-saluran data *mikrokontroler*. Pencacah biner ini sangat dipengaruhi oleh sinyal detak (*clock*) yang dihasilkan oleh kristal yang dipasang. *clock* dibedakan menjadi dua macam, yang pertama *clock* dengan frekuensi tetap yang telah diketahui besarnya dan yang kedua ialah *clock* dengan frekuensi yang tak tentu atau bervariasi dan dapat diatur.

Jika *clock* yang dipakai pencacah menggunakan frekuensi yang tetap, maka pencacah tersebut bekerja sebagai *Timer* atau pewaktu, sedangkan jika *clock* yang dipakai menggunakan frekuensi yang bervariasi, maka pencacah tersebut berfungsi sebagai *counter* atau penghitung.

Mikrokontroler AT89S51 memiliki dua *Timer/ Counter* yaitu *Timer 0* dan *Timer 1*. Untuk mengaktifkan *Timer/ Counter* menggunakan *register* khusus dalam SFR (*Special Function Register*). *Timer 0* melalui TL0 (*Timer 0 Low Byte* alamat 6Ah) dan *register* TH0 (*Timer 0 High Byte* alamat 6Ch). sedangkan untuk *Timer 1* melalui TL1 (*Timer 1 Low Byte* alamat 6Bh) dan *register* TH1 (*Timer 1 High Byte* alamat 6Dh). (Putra, 2002)

Register TMOD dibagi menjadi dua kelompok, TMOD.0-TMOD.3 untuk mengatur *Timer 0*, sedangkan TMOD.4-TMOD.7 digunakan untuk mengatur *Timer 1*. Keterangan untuk *register* TMOD sebagai berikut :

GATE : Merupakan pengatur saluran sinyal detak. Bila GATE 0 maka *Timer/ Counter* diatur oleh bit TRx pada *register* TCON (aktif jika TRx bernilai 1). Sebaliknya jika GATE 1 maka *Timer/ Counter* diatur oleh pin INTx (*Timer/ Counter* akan aktif jika pin INTx bernilai 1).

C/ \bar{T} : *Timer/ Counter selector*, jika 0 sebagai *Timer* (*Clock* dari osilator kristal internal). Jika 1 sebagai *Counter* (*Clock* diperoleh dari pin masukan Tx).

M0 : Mode kerja *Timer/ Counter*.

M1 : Mode kerja *Timer/ Counter*.

Tabel 2.2 Mode Kerja *Timer/ Counter*

M0	M1	Mode	Keterangan
0	0	0	Pencacah 13-bit
0	1	1	Pencacah 16-bit
1	0	2	Pencacah 8-bit auto reload
1	1	3	Split <i>Timer</i> mode, pencacah 8-bit pada TL0 dan TH0 yang dikendalikan oleh TRx control bit.

Sedangkan untuk *register* TCON juga dibagi menjadi dua bagian, TCON.0-TCON.3 dipakai untuk mengatur masukan INT0 dan INT1

selanjutnya TCON.4-TCON.7 untuk *Timer*. Penjelasan untuk *Register* TCON adalah sebagai berikut.

TF_x : (TF0 dan TF1) merupakan bit penampung limpahan. TF_x akan 1 jika pencacah biner yang terhubung padanya melimpah (*overflow*). Bit TF_x dapat dinolkan secara manual dengan instruksi CLR TF0 atau CLR TF1. Jika interupsi dari *Timer 0/ Timer1* dipakai, TF_x diberi logika nol saat *mikrokontroler* menjalankan Rutin Layanan Interupsi (ISR – *Interrupt Service Routine*)

TR_x : (TR0 dan TR1) merupakan bit pengatur saluran sinyal detak, bila bit ini = 0 maka sinyal detak tidak disalurkan ke pencacah biner sehingga pencacah berhenti melakukan pencacahan. Bit GATE = 1 pada *register* TMOD, maka saluran sinyal detak ini diatur bersama oleh bit TR_x dan pada sinyal kaki INT0/ INT1

2.9 Sistem Interupsi

Interupsi adalah penghentian sejenak program utama yang sedang dijalankan, kemudian *mikrokontroler* menjalankan interupsi tersebut. AT89S51 menyediakan lima sumber interupsi, yaitu dua interupsi eksternal (*External Interrupt*), dua interupsi pewaktu (*Timer Interrupt*), serta satu interupsi Port serial (*Serial Port Interrupt*). Kesemua interupsi tersebut dapat diaktifkan atau tidak melalui bit-bit yang terdapat dalam *register* IE (*Interrupt Enable*) yang terletak pada alamat A8H. (Putra, 2002)

Register IE dapat mengatur salah satu interupsi yang diaktifkan atau keseluruhan interupsi aktif. Bit yang dapat digunakan untuk mengaktifkan atau tidak interupsi secara keseluruhan yaitu IE.7 (EA). Jika bit ini di set maka seluruh interupsi akan diaktifkan, sebaliknya jika di *clear* akan mematikan interupsi secara keseluruhan.

Keterangan untuk bit-bit pada *Register* IE sebagai berikut.

- EA : Untuk mengaktifkan atau mematikan seluruh interupsi secara serentak atau keseluruhan.
- ES : Untuk mengaktifkan atau mematikan Interupsi Port Serial.
- ET1 : Untuk mengaktifkan atau mematikan Interupsi *Timer 1 overflow*.
- EX1 : Untuk mengaktifkan atau mematikan Interupsi Eksternal 0.
- ET0 : Untuk mengaktifkan atau mematikan Interupsi *Timer 0 overflow*.
- EX0 : Untuk mengaktifkan atau mematikan Interupsi Eksternal 0.

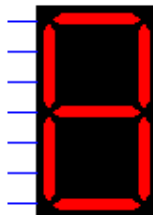
2.10 Seven Segmen

Dikenal sebagai suatu *seven-segment* indikator, adalah suatu format dari alat tampilan yang suatu alternatif ke *dot-matrix* tampilan yang semakin kompleks. *Seven-segment* biasanya digunakan di dalam elektronika sebagai metoda dari mempertunjukkan umpan balik klasifikasi sistim desimal dengan operasi internal tentang alat.

Seven segment diatur sebagai segiempat panjang dari dua segmen yang vertikal pada masing-masing sisi dengan satu segmen yang horizontal di bagian atas dan alast. Segmen yang ketujuh membagi dua bagian segiempat panjang secara horizontal.

Kebanyakan 7-segment menggunakan tampilan suatu array dari diode pemancar cahaya (LEDS), meskipun demikian ada lain jenis teknologi alternatif penggunaan seperti lucutan gas katode dingin, ruang hampa kawat pijar yang bercahaya berpijar, tampilan hablur cair (LCD), dan lain lain. Karena gambar lambang suku harga gas dan lain tanda yang besar, *electromagnetically* dilemparkan *light-reflecting* segmen masih biasanya digunakan. Suatu alternatif bagi 7-segment di 1950 sampai 1970 adalah nixie tabung ruang hampa yang seperti tabung. Mulai 1970, RCA yang dijual suatu alat tampilan yang dikenal sebagai numitron yang menggunakan kawat pijar bercahaya mengatur ke dalam suatu *seven-segment*.

Sebagian dari tampilan yang terintegrasi ini menyertakan decoder internal mereka sendiri, meskipun demikian paling tidak masing-masing LED individu diterbitkan persis sama benar menghubungkan pin seperti diuraikan. *multiple-digit* led seperti yang digunakan dalam kalkulator saku dan alat yang serupa digunakan tampilan *multiplexed* untuk mengurangi banyaknya IC pin diperlukan untuk mengendalikan tampilan. Sebagai contoh, semua kutub positif dari suatu segmen dari tiap posisi digit akan dihubungkan bersama-sama dan persis sama benar pengarah meletakkan/menjejit, selagi katode dari semua segmen untuk masing-masing digit akan dihubungkan. Untuk beroperasi manapun segmen yang tertentu tentang segala digit, *integrated sirkuit* pengendalian akan memutar dengan diam-diam pengarah katode untuk digit yang terpilih, dan pengarah kutub positif untuk segmen yang diinginkan hingga batas tertentu yang pendek/singkat hingga kosong interval digit yang berikutnya akan terpilih dan pengarah kutub positif untuk segmen yang diinginkan, kemudian hingga batas tertentu yang pendek/singkat kosong interval digit yang berikutnya akan terpilih dan segmen yang baru dinyalakan. (<http://one.indoskripsi.com/node/667>)



Gambar 2.8 *Seven Segment*

2.11 Sensor Ultrasonik

Sensor ultrasonik adalah sensor yang bekerja berdasarkan prinsip pantulan gelombang suara dan digunakan untuk mendeteksi keberadaan suatu objek tertentu di depannya, frekuensi kerjanya pada daerah diatas gelombang suara dari 40 KHz hingga 400 KHz. Gelombang ini dapat merambat dalam medium padat, cair dan gas, hal disebabkan karena gelombang ultrasonik merupakan rambatan energi dan momentum mekanik sehingga merambat sebagai interaksi dengan molekul dan sifat enersia medium yang dilaluinya. Karakteristik gelombang ultrasonik yang melalui medium mengakibatkan getaran partikel dengan medium amplitudo sejajar dengan arah rambat secara longitudinal sehingga menyebabkan partikel medium membentuk rapatan (*Strain*) dan tegangan (*Stress*). (<http://tiyoavianto.com/sensor-ultrasonic.html>, 2010)



Gambar 2.9 Sensor Ultrasonik

(<http://atmelmikrokontroler.wordpress.com/2009/06/12/sensor-ultrasonik>, 2010)

BAB III

DESAIN DAN PERANCANGAN

3.1. Identifikasi Kebutuhan

Untuk membantu dalam pembuatan *prototype* penghitung bis masuk, maka diperlukan identifikasi kebutuhan terhadap alat yang akan dibuat antara lain :

1. Adanya alat yang mengirimkan suatu data digital.
2. Adanya alat yang menerima suatu data digital.
3. Adanya alat yang mengolah data digital untuk mendapatkan hasil
4. Adanya alat yang menunjukkan hasil dari pencacah.
5. Adanya alat yang memberikan daya.

Adapun beberapa identifikasi kebutuhan dalam perancangan alat yang akan dibuat, antara lain:

1. Sistem minimum AT89S51 (sebagai pengolah data digital)
2. Rangkaian Catu daya (sebagai pemberi daya)
3. Rangkaian display 7-Segment (*display* hasil)
4. Sensor Ultrasonik TX dan RX (sebagai penerima dan pengirim sinyal digital)

3.2. Analisis Kebutuhan

Berdasarkan identifikasi kebutuhan tersebut diatas, maka diperoleh beberapa analisis kebutuhan alat yang dibuat dengan spesifikasi sebagai berikut :

1. Sensor Ultrasonik Transmitter sebagai pengirim data digital.
2. Sensor Ultrasonik Receiver sebagai penerima data digital.
3. *Mikrokontroller* AT89S51 sebagai komponen yang mengolah data dari *receiver*.
4. *Seven Segmen* 3 digit sebagai *display* dari komponen pengolah data.

5. Catu daya digunakan untuk memberikan daya pada alat agar seluruh komponen yang terpasang pada alat tersebut dapat bekerja dengan baik.

3.3. Pembuatan Alat

25

bahan yang diperlukan.

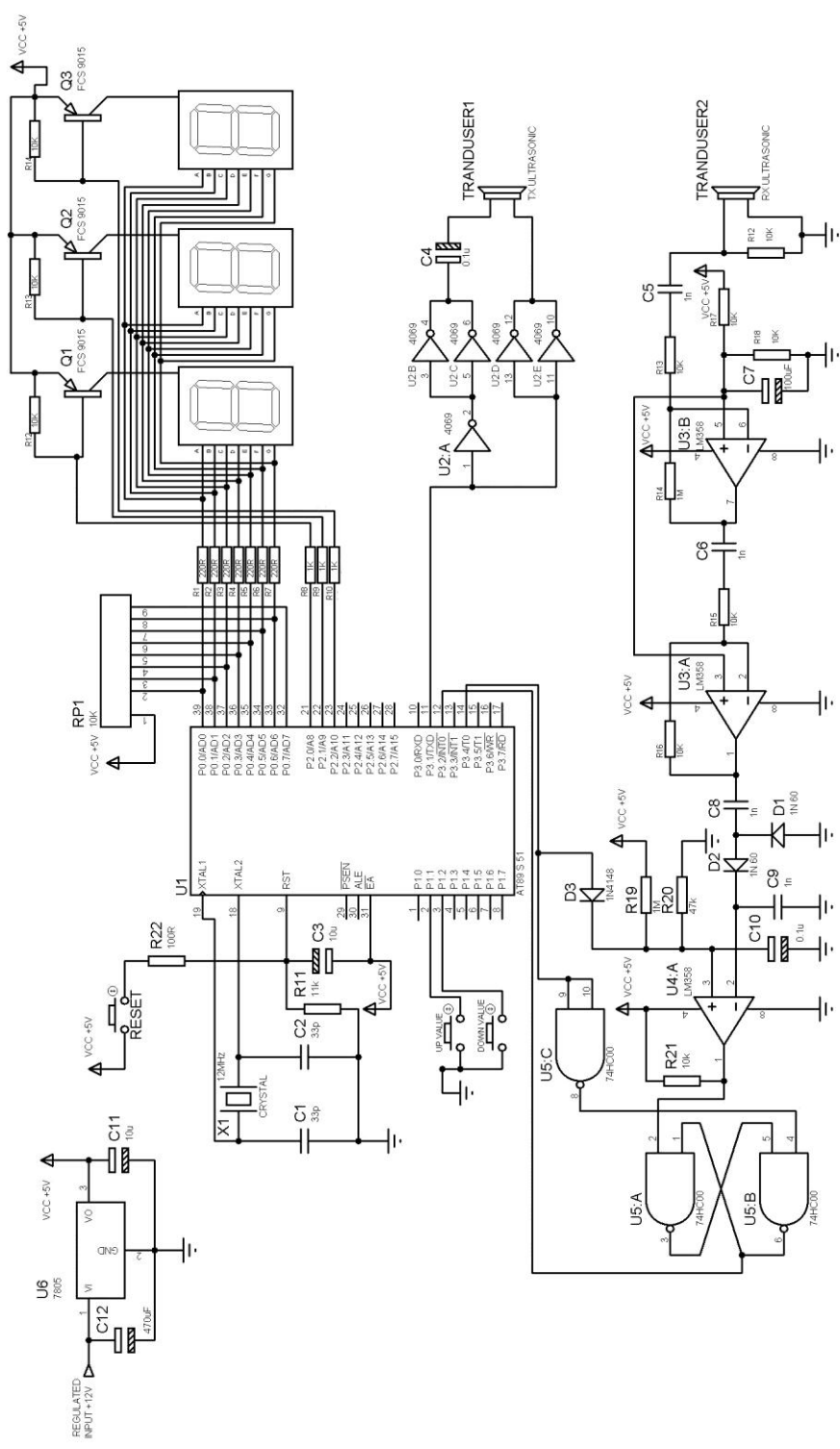
1. Alat dan bahan yang diperlukan antara lain:
 - a. Komputer untuk menggambar *layout* PCB, memprogram untuk *mikrokontroler* AT89S51.
 - b. Downloader untuk *mikrokontroler* AT89S51
 - c. Multimeter
 - d. Solder dan timah solder
 - e. Papan PCB
 - f. Bor
 - g. Komponen-komponen yang diperlukan
2. Proses pembuatan alat adalah sebagai berikut:
 - a. Membuat program aplikasi buat *mikrokontroler* dan PLC
 - b. Menggambar *layout* PCB dengan bantuan computer
 - c. Mencetak *layout* pada papan PCB
 - d. Menguji kondisi komponen
 - e. Merangkai komponen
 - f. Menguji rangkaian

3.4. Perancangan alat keras

Dalam perancangan alat ini diperlukan 3 bagian komponen alat yaitu sensor ultrasonik (*Receiver* dan *Transmitter*), *mikrokontroller* (berisi program yang menjalankan alat penghitung bis masuk) dan display yang menggunakan 7 Segmen 3 digit.

TX US (*Transmitter Ultrasonic*) mengirimkan sebuah sinyal jika mengenai objek (yang dimaksudkan disini adalah sebuah bis) akan diterima RX US (*Receiver Ultrasonic*) yang kemudian dikirimkan ke *mikrokontroller* yang didalamnya sudah terdapat program pencacah dan diperlihatkan hasilnya melalui *seven segment*.

Dan didalam alat tersebut selain melibatkan 3 komponen penting diatas terdapat pula *power supply* dan tombol reset (untuk mereset/mengulang penghitungan dari nol), tombol *up value* dan *down value* (tombol ini dapat berfungsi jika menggunakan penghitungan secara manual).



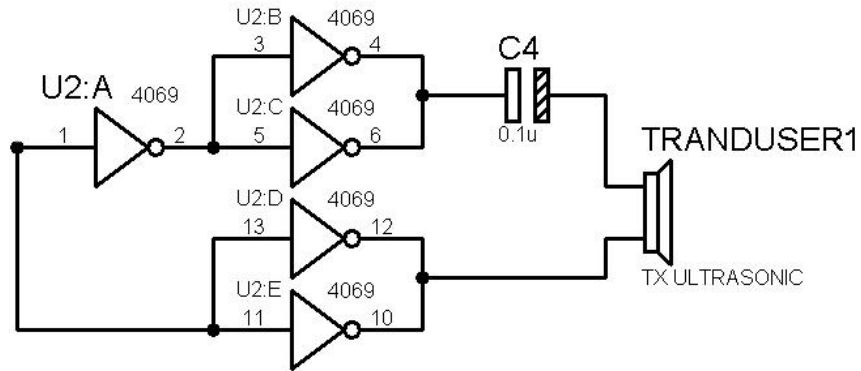
Gambar 3.1 Desain *Prototype* Penghitung Jumlah Bis

3.4.1 Bagian Pemancar Sensor Ultrasonic

Pada bagian ini memuat komponen pengirim dan penerima data digital.

1. Bagian Pemancar

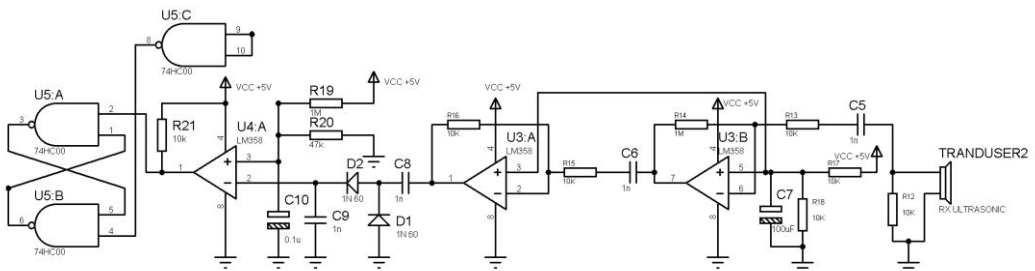
Pada komponen ini mengirim sebuah sinyal terus menerus, jika mengenai sebuah objek maka sinyal tersebut akan dipantulkan ke bagian penerima.



Gambar 3.2 Bagian Pemancar Sensor ultrasonik

2. Bagian Penerima

Setelah menerima sebuah data dari pemancar maka akan dikirimkan ke dalam *Mikrokontroller* yang kemudian diproses untuk memperlihatkan hasil.

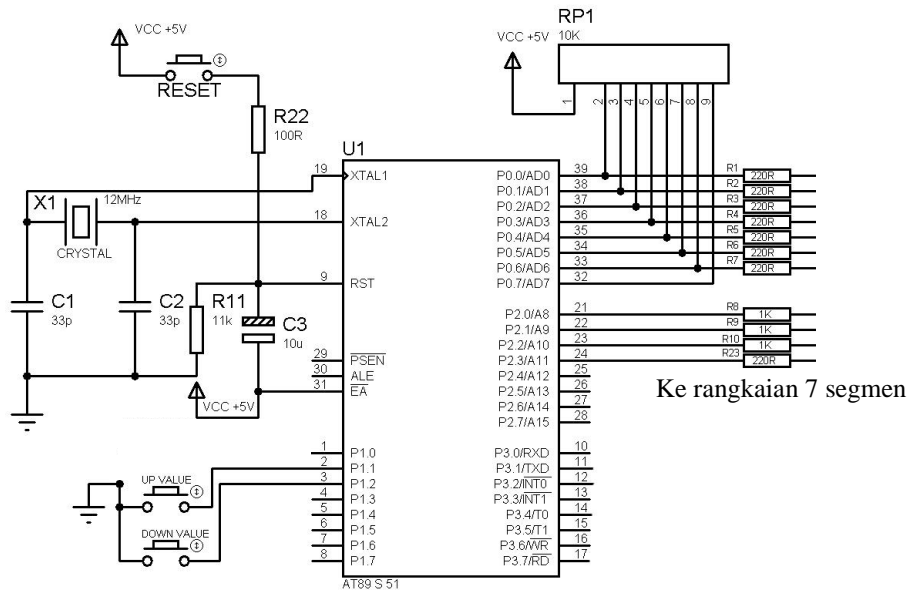


Gambar 3.3 Bagian Penerima Sensor Ultrasonik

3.4.2 Mikrokontroler AT89S51

Didalam mikro ini terdapat program-program yang mengacu terhadap kinerja alat penghitung bis. Didalam mikro ini terdapat program-program yang dieksekusi atau dijalankan, antara lain :

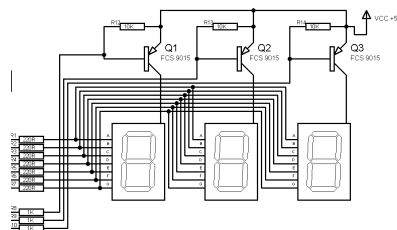
1. Program Pencacah
2. Program Reset
3. Program UP Value dan DOWN Value



Gambar 3.4 Bagian Mikrokontroler

3.4.3 Display

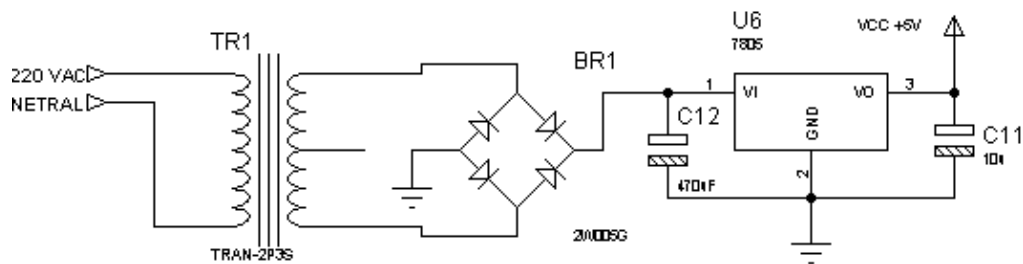
Untuk menampilkan hasil pengitungan dari Mikro tersebut disini menggunakan *seven segmen* 3 digit.



Gambar 3.5 *Display Seven Segmen 3 Digit*

3.4.4 Rangkaian Catu Daya

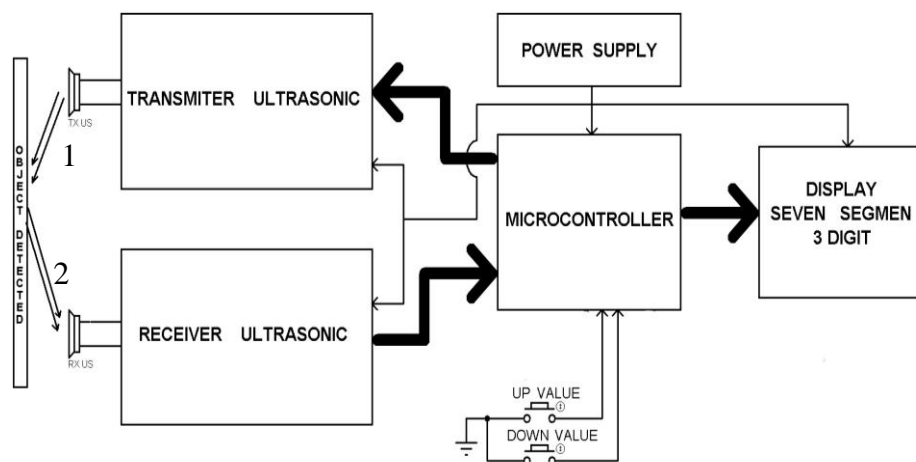
Rangkaian catu daya berfungsi untuk memberikan daya pada alat. Rangkaian ini mengubah tegangan bolak – balik (AC) dari listrik PLN menjadi tegangan searah (DC) yang berguna sebagai sumber tenaga alat dan membuat alat tersebut hidup atau menyala. Rangkaian ini menggunakan sebuah transformator atau trafo sebagai pengubah besar tegangan. Pada rangkaian ini menggunakan trafo step down. Keluaran dari trafo diambil tegangan AC yang diambil 12 Volt disearah dioda menjadi 12 Volt DC kemudian diturunkan menjadi 5 Volt menggunakan IC Regulator.



Gambar 3.6 Bagian Catu Daya

3.5. Diagram Blok *Prototype* Penghitung Bis Masuk

Untuk mempermudah pengerjaan alat penghitung bis maka sebelumnya dibuat sebuah diagram blok yang secara sistematis menerangkan bagaimana alat ini bekerja.



Keterangan : yang digaris tebal merupakan input dan output dari mikro.

nomor 2 : sinyal keluaran dari sensor

nomor 3 : sinyal masukan ke sensor

Gambar 3.7 Diagram Blok

Seperti yang digambarkan diatas dimana alat tersebut bekerja dimulai dari *transmitter ultrasonic* mengirimkan data berupa sinyal dan mengenai objek (bis) kemudian diterima oleh *receiver ultrasonic* dan diperkuat sinyal yang masuk tersebut dan diolah oleh program mikro dan hasilnya ditampilkan melalui *seven segmen 3 digit*.

Untuk membuat alat ini lebih optimal maka dibuat 2 versi penghitungan dalam alat ini. Versi ini dibuat untuk menghindari jika sensor ultrasonik tidak berfungsi atau rusak, alat ini masih dapat bekerja. Untuk

tombol pilihan jika ditekan untuk penghitungan manual maka lampu led akan menyala dan jika untuk penghitungan otomatis maka lampu led tidak menyala.

Ada 3 proses yang terjadi dalam penghitungan jumlah bis masuk, adalah :

1. Pengambilan sinyal digital oleh Sensor Ultrasonik

Alat dinyalakan, maka sensor TX (*Transmitter*) akan memancarkan sinyal ultrasonik ke objek (yang dimaksudkan disini adalah sebuah bis), kemudian setelah mengenai objek sinyal akan dipantulkan oleh objek dan diterima oleh sensor RX (*Receiver*) yang kemudian akan meneruskan ke bagian *mikrokontroler* untuk selanjutnya data diproses untuk memperoleh hasil.

2. Pengolahan data

Data yang telah diperoleh dari sensor RX (*Receiver*) maka akan menjadi input di dalam *mikrokontroler*. Data ini akan diolah untuk proses penghitungan bis masuk yang menjadi dasar dari fungsi kerja alat ini.

3. Display 7 Segment

Data-data yang masuk ke *mikrokontroler* akan diolah dengan menggunakan program-program assembler untuk kemudian ditampilkan pada display 7 Segment melalui port 0 pada AT89S51.

Proses penampilan hasil pengukuran tinggi benda yang dikeluarkan melalui 7 segment

3.6. Perancangan Software

3.6.1 Software Pendukung

Software yang dipakai pada proses pembuatan alat, antara lain:

- a. Notepad

Pemrograman pada *mikrokontroler* AT89S51 ditulis dengan menggunakan notepad dan kemudian disimpan dengan memberikan ekstensi *.asm

b. ASM51

Program ini digunakan untuk mengeksekusi program yang telah ditulis pada notepad yang berekstensi *.ASM agar menghasilkan file listing (*.LST) dan file hexa (*.HEX).

c. AEC_ISP

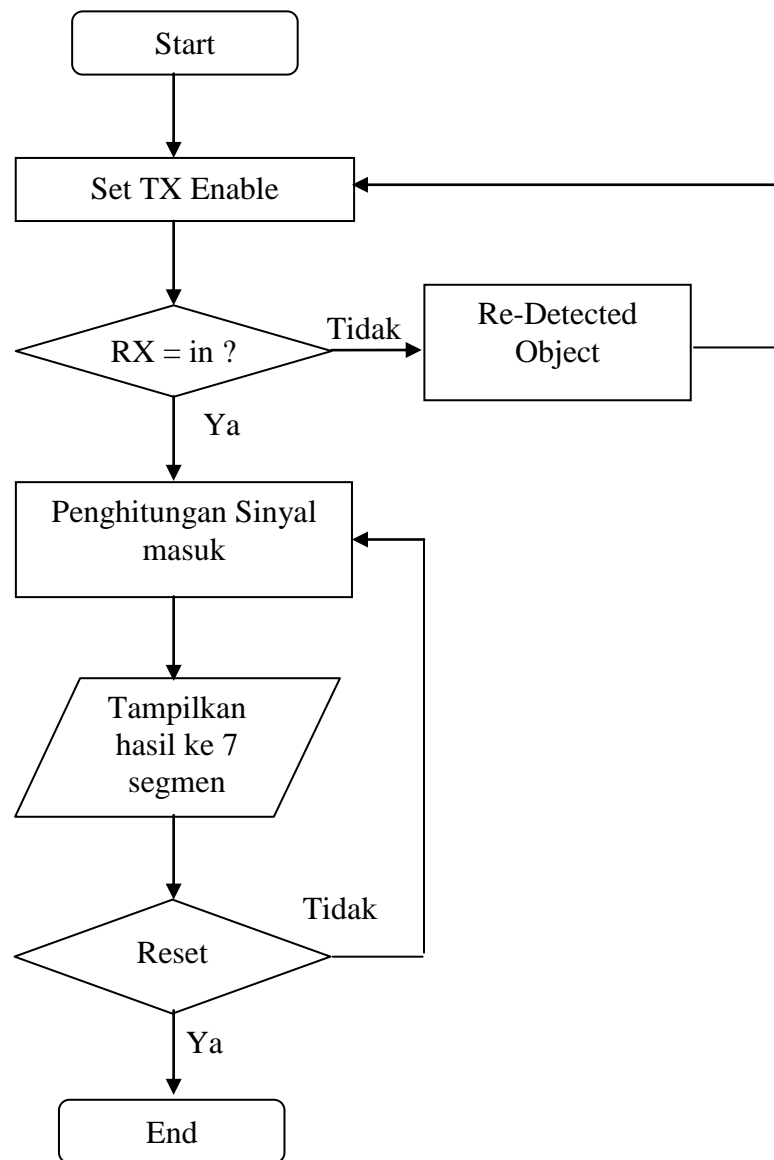
Program AEC_ISP sebagai program untuk memanggil file hexa (*.HEX) yang kemudian didownload ke dalam *mikrokontroller* AT89S51 untuk menjalankan instruksi – instruksi yang telah dibuat.

d. PROTEUS 7.4

Merupakan program simulasi elektronik yang digunakan sebagai media pembelajaran dalam pembuatan rangkaian elektronika dan menggambar rangkaian tersebut.

3.6.2 Flowchart

Untuk memudahkan perancangan *software* maka diawali dengan membuat *flowchart*.



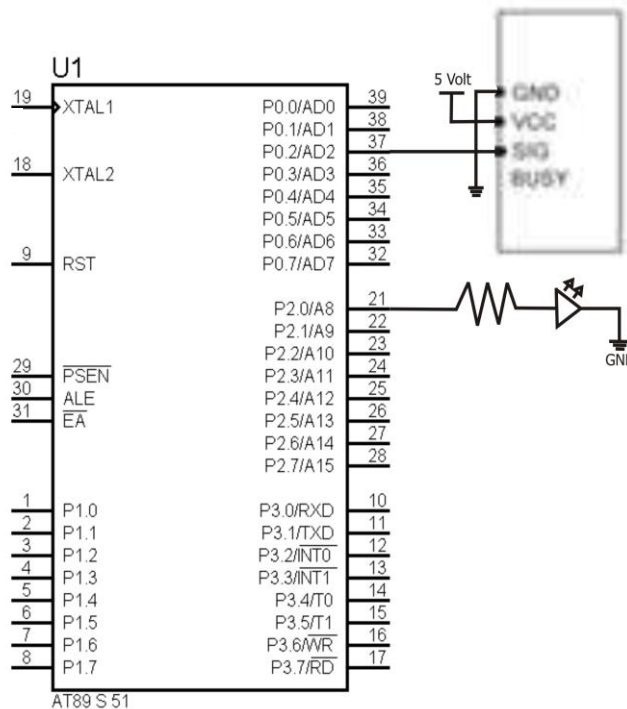
Gambar 3.8 *Flow Chart*

BAB IV PENGUJIAN DAN ANALISIS

4.1 Pengujian Alat

4.1.1 Pengujian Sensor Ultrasonik

Tujuan pengujian sensor ultrasonik adalah untuk mengecek apakah sensor bekerja dengan baik. Untuk menguji sensor ultrasonik ini menggunakan rangkaian sebagai berikut.

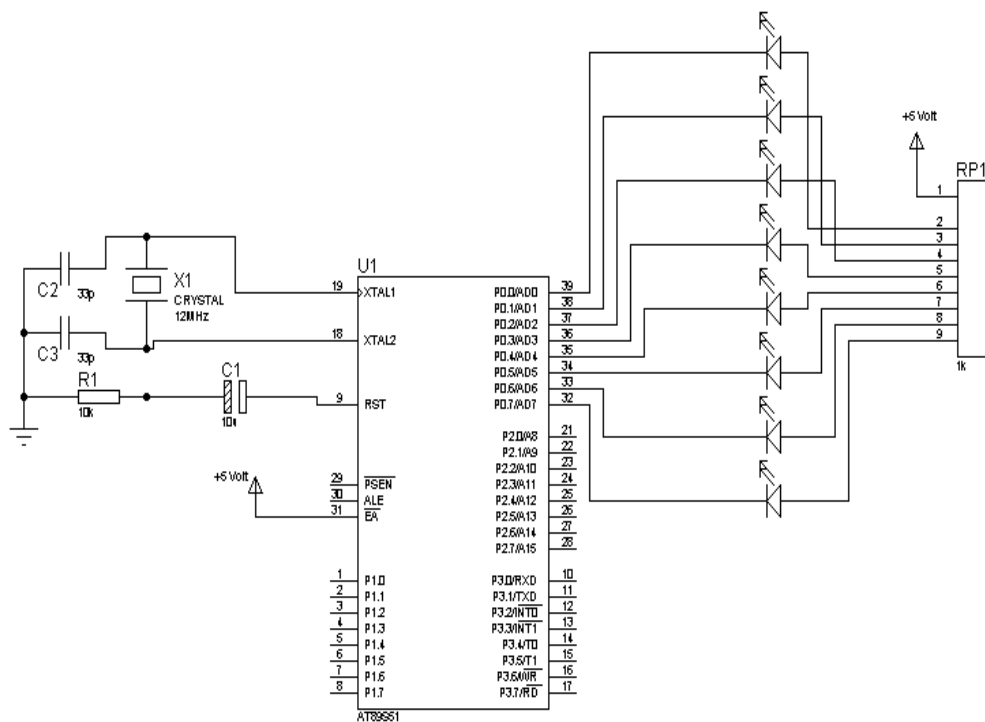


Gambar 4.1 Rangkaian Uji Sensor Ultrasonik

Pada rangkaian di atas, sensor ultrasonik dihubungkan ke mikrokontroler AT89S51 dan juga sebuah led untuk indikator yang menunjukkan sensor bekerja dengan baik atau tidak. Ketika suatu objek terdeteksi oleh sensor maka lampu led akan menyala, dan jika tidak ada objek yang terdeteksi oleh sensor, maka led akan mati dikarenakan tidak ada suatu input yang berasal dari sensor ultrasonik, menandakan sensor berfungsi dengan baik.

4.1.2 Pengujian Mikrokontroller

Untuk mengecek apakah mikrokontroller sudah berfungsi secara benar maka dilakukan pengujian atau pengecekan, dan pengecekan dilakukan pada kaki-kaki mikrokontroller. Alat yang dibutuhkan untuk menguji mikrokontroller adalah Led, Resistor, dan Minimum Sistem Mikrokontroller AT89S51. Rangkaian elektroniknya adalah sebagai berikut:



Gambar 4.2 Rangkaian Uji Mikrokontroller

Adapun urutan langkah – langkah pengujiannya adalah sebagai berikut:

1. Untuk memastikan bahwa mikrokontroller bekerja dengan baik maka hal yang pertama dilakukan adalah membuat suatu program untuk menyalakan LED yang akan diproses dalam mikrokontroller AT89S51. Berikut listing program yang dimasukkan adalah sebagai berikut:

```

$mod51
ORG    00H
LJMP   START
ORG    4200H
MOV    SP,#20H
SETB   P0.0
SETB   P0.1
SETB   P0.2
SETB   P0.3
SETB   P0.4
SETB   P0.5
SETB   P0.6
SETB   P0.7
SJMP   $
END

```

2. Hasil setelah program dimasukkan ke dalam *mikrokontroller* adalah:

Tabel 4.1 Hasil Uji *Mikrokontroller*

LED	Hasil
0	Menyala
1	Menyala
2	Menyala
3	Menyala
4	Menyala
5	Menyala
6	Menyala
7	Menyala

Pengujian sistem minimum dilakukan untuk mendapatkan kebenaran kerja dari rangkaian sistem minimum, adapun pengujian secara sederhana dilakukan dengan memasukkan program pendek pada

mikrokontroler semisal menghidupkan Led pada rangkaian, apabila nyala led telah sesuai dengan program, maka rangkaian sistem minimum telah berfungsi dengan baik.

3. Analisis

Lampu led pada rangkaian akan menyala diberi logika 1.

4.1.3 Pengujian Secara Keseluruhan

Pengujian ini bertujuan untuk mengetahui dan memastikan unjuk kerja alat secara keseluruhan mulai dari *transmitter*, *receiver*, sistem minimum, rangkaian alat. Dalam hal ini unjuk kerja alat telah sesuai dengan yang diharapkan, tombol-tombol manual yang digunakan telah berfungsi sesuai dengan yang diharapkan.

Adapun prosedur pengujiannya adalah:

1. Menghubungkan alat dengan catu daya
2. Melewatkan objek diatas sensor.
3. Mencatat hasil penghitungan.

Setelah melakukan pengujian alat, didapat hasil dari pengujian alat sebagai berikut:

Tabel 4.2 Hasil Uji Sistem Keseluruhan

Pengujian	Jarak	Otomatis
Pengujian 1	1 cm	Sensor memberikan nilai
Pengujian 2	2 cm	Sensor memberikan nilai
Pengujian 3	3 cm	Sensor memberikan nilai
Pengujian 4	4 cm	Sensor memberikan nilai
Pengujian 5	5 cm	Sensor memberikan nilai
Pengujian 6	6 cm	Sensor memberikan nilai
Pengujian 7	7 cm	Sensor memberikan nilai
Pengujian 8	8 cm	Sensor memberikan nilai
Pengujian 9	9 cm	Sensor memberikan nilai
Pengujian 10	10 cm	Sensor memberikan nilai

Pengujian 11	15 cm	Sensor memberikan nilai
Pengujian 12	20 cm	Sensor memberikan nilai
Pengujian 13	25 cm	Sensor memberikan nilai
Pengujian 14	30 cm	Sensor memberikan nilai
Pengujian 15	35 cm	Sensor memberikan nilai
Pengujian 16	40 cm	Sensor memberikan nilai
Pengujian 17	45 cm	Sensor memberikan nilai
Pengujian 18	46 cm	Sensor terkadang memberikan nilai
Pengujian 19	47 cm	Sensor terkadang memberikan nilai
Pengujian 20	48 cm	Sensor tidak memberikan nilai
Pengujian 21	50 cm	Sensor tidak memberikan nilai

Dari hasil pengujian, jarak terjauh untuk sensor dapat memberikan hasil adalah 45 cm, sedangkan untuk 46 cm dan 47 cm sensor terkadang memberikan nilai dan tidak, sedangkan untuk jarak 48 ke atas sensor sama sekali tidak memberikan nilai. Tombol manual pada alat untuk memberikan input jika sensor *ultrasonik* tidak berfungsi.



Gambar 4.3 Alat Sesudah dijalankan



Gambar 4.4 *Prototype* Penghitung Bis

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pengamatan dan pembahasan pada pembuatan tugas akhir ini, maka penulis dapat memberikan kesimpulan sebagai berikut:

1. Pembuatan *prototype* penghitung bis masuk menggunakan AT89S51 terdiri dari beberapa bagian, yaitu rangkaian sensor *ultrasonic*, rangkaian *mikrocontroller* dan rangkaian *display* menggunakan *seven segment*.
2. Kinerja alat penghitung bis sebagai berikut :
 - a. Secara otomatis, yaitu sensor mendeteksi objek dan mengirim data untuk diproses dan dihitung kemudian ditampilkan hasilnya.
 - b. Secara manual, yaitu jika sensor *ultrasonic* rusak atau tidak berfungsi dengan baik maka dapat menggunakan tombol *up value* dan *down value* yang sudah tersedia dalam alat tersebut.
3. Hasil uji coba alat secara keseluruhan jarak yang terdeteksi dari 1 cm sampai 45 cm.

5.2 Saran

Berdasarkan perancangan dan analisa data pengamatan, untuk mendapatkan hasil yang maksimal atau peralatan ini dapat dikembangkan lebih lanjut, maka penulis menyarankan untuk:

1. *Prototype* penghitung bis ini dapat dikembangkan lagi dengan penambahan fitur – fitur lain seperti kamera terintegrasi didalam alat ini sehingga petugas bisa langsung membatalkan data yang masuk jika objek yang terdeteksi bukan bis.
2. Penggunaan alat ini diletakkan pada tempat yang khusus, sehingga keakuratan dari hasil pengukuran akan lebih optimal dan untuk menghindari kesalahan pendeteksian objek.
3. *Prototype* penghitung bis dapat dikembangkan lebih lanjut untuk memenuhi kebutuhan lain secara global.

4. DAFTAR PUSTAKA

5.

6. Putra, A.E, 2002, *Belajar Mikrokontroller AT89C51/52/55 (Teori dan Aplikasi)*, Gava Media, Yogyakarta.
- 7.
8. <http://www.atmel.com> (diakses tanggal 4 Mei 2009)
- 9.
10. <http://www.atmelmikrokontroler.wordpress.com/2009/06/12/sensor-ultrasonik>
11. (diakses tanggal 7 Februari 2010)
- 12.
13. <http://www.electroniclab.com> (diakses tanggal 4 Mei 2009)
- 14.
15. <http://www.ilmukomputer.com> (diakses tanggal 4 Mei 2009)
- 16.
17. <http://www.mytutorialcafe.com> (diakses tanggal 4 Mei 2009)
- 18.
19. <http://www.one.indoskripsi.com/node/667> (diakses tanggal 5 Mei 2009)
- 20.
21. <http://www.toko-elektronika.com/tutorial/uc1.html> (diakses tanggal 4 Mei 2009)
- 22.
23. <http://www.tiyoavianto.com/sensor-ultrasonic.html> (diakses tanggal 7 Februari 2010)
- 24.