

**Perancangan Sistem Deteksi Sinyal Suara (*Audio*) pada
Modulasi FM (*Frekuensi Modulasi*) dengan Menggunakan
Mikrokontroler AT89C51**

SKRIPSI

*Diajukan Untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik*



Disusun Oleh :

Frederik B. P. Situmorang

0310633044-63

**DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2009**

**Perancangan Sistem Deteksi Sinyal Suara (Audio) pada
Modulasi FM (Frekuensi Modulasi) dengan Menggunakan
Mikrokontroler AT89C51**

SKRIPSI

*Diajukan Untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik*



Disusun Oleh :

Frederik B. P. Situmorang

0310633044-63

Dosen Pembimbing :

Ir. Endah Budi P., MT
NIP 131 839 362

Ali Mustofa, ST., MT.
NIP. 132 258 187

**Perancangan Sistem Deteksi Sinyal Suara (*Audio*) pada
Modulasi FM (*Frekuensi Modulasi*) dengan Menggunakan
Mikrokontroler AT89C51**

Disusun Oleh :

Frederik B. P. Situmorang

0310633044-63

Skripsi ini telah diuji dan dinyatakan lulus
pada 28 Januari 2009

Dosen Penguji :

Ir. Erfan Ahmad Dahlan, MT
NIP. 131 124 663

Rusmi Ambarwati, ST., MT.
NIP. 132 258 188

Ir. Wahyu Adi P., MT.
NIP. 131 759 602

Mengetahui
Ketua Jurusan Teknik Elektro

Ir. Heru Nurwarsito, M. Kom
NIP. 131 879 033

KATA PENGANTAR

Puji syukur penulis panjatkan atas kehadiran Tuhan Yang Maha Ku asa karena berkat kasih dan karunia-Nya penulis dapat menyelesaikan skripsi dengan judul "Perancangan Sistem Deteksi Kualitas Sinyal Informasi pada Modulasi FM (*Frekuensi Modulasi*) dengan Menggunakan Mikrokontroler AT89C51".

Pada kesempatan ini, penulis mengucapkan terima kasih kepada:

1. Yesus Kristus, yang senantiasa menyertai dan memberikan kasihnya dalam menyelesaikan skripsi ini.
2. Bapak dan Mamak tercinta yang selalu memberikan doa dan dukungannya, terima kasih atas semua yang telah diberikan sehingga tercapai semua cita-cita ini.
3. Bapak Ir. Heru Nurwarsito, M.Kom. selaku Ketua Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
4. Bapak Rudy Yuwono, ST, MSc. selaku Sekretaris Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
5. Ibu Ir. Endah Budi P., MT. selaku Ketua Kelompok Dosen Keahlian Bidang Telekomunikasi dan sekaligus pembimbing I dalam penyusunan skripsi ini.
6. Bapak Ali Mustofa, ST., MT. selaku pembimbing II dalam penyusunan skripsi ini.
7. Bapak dan Ibu dosen serta karyawan Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
8. Seseorang yang sangat berarti dan selalu menginspirasi, Corry Atur Rosida terima kasih atas cinta dan semua dukungannya.
9. Untuk keluarga besar penulis, Mbak Yun dan Mas made beserta keluarga, Mas Taufik dan keluarga, Mas wawan dan keluarga, Mas didik dan keluarga, dan semua pihak keluarga yang penulis tidak bisa sebut satu-persatu, terima kasih atas doa dan dukungannya.
10. Keluarga besar di Semboro Jember, terima kasih atas doa dan dukungannya.
11. Hendra dan Tharom terima kasih atas semuanya.
12. Semua teman-teman kontrakan Bratan IV, Joko, Dani, terima kasih atas doa dan dukungannya.
13. Serta semua pihak yang tak dapat disebutkan satu persatu yang telah turut membantu baik secara langsung maupun tidak langsung dalam penyelesaian Skripsi ini.

Dalam pembuatan Skripsi ini penulis menyadari terdapat kekurangan dan keterbatasan di dalamnya. Oleh karena itu, dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang bersifat membangun dari semua pihak demi penyempurnaan Skripsi ini. Semoga, Skripsi ini dapat bermanfaat bagi kita semua. Amin.

Malang, 15 Januari 2009

Penulis



DAFTAR ISI

KATA PENGANTAR i

DAFTAR ISI iii

DAFTAR TABEL vii

DAFTAR GAMBAR viii

ABSTRAKSI x

BAB I PENDAHULUAN

1.1 Latar Belakang 1

1.2 Rumusan Masalah 3

1.3 Ruang Lingkup 3

1.4 Tujuan 3

1.5 Manfaat 4

1.6 Sistematika Penulisan 4

BAB II TINJAUAN PUSTAKA

2.1 FM (Frekuensi Modulasi) 6

 2.1.1 Siaran FM 6

 2.1.2 Pemancar FM 7

 2.1.3 Modulasi 9

 2.1.4 Modulasi Frekuensi 10

2.2 *Signal to Noise Ratio* 12

 2.2.1 *Noise Figure* 13

2.3 *High Pass Filter (HPF)* 13

2.4 ADC tipe SAR 15

2.5 Rangkaian *Integrator* 16

2.6 Penguat Tak Membalik (*Non Inverting Amplifier*) 17

2.7 Pembanding Tak Membalik (*Non Inverting Comparator*) 19

2.8 Mikrokontroler AT89C51 20

2.9 Bahasa *Assembly* 23

 2.9.1 Konstruksi Bahasa *Assembly* 24

2.10 Multiplexer CD 4051 26



2.11	Gerbang EXOR.....	27
2.12	<i>Borland Delphi 7</i>	28
2.12.1	<i>Serial Port</i> pada <i>Borland Delphi 7</i>	29
2.13	RS-232.....	31
2.14	Pengertian <i>7 Segment</i> dan <i>Serial Port DB 9</i>	32
2.14.1	Pengertian <i>7 Segment</i>	32
2.14.2	<i>Serial Port DB 9</i>	33

BAB III METODOLOGI

3.1	Studi Literatur.....	35
3.2	Analisis Kebutuhan.....	35
3.3	Perancangan Sistem.....	36
3.3.1	Perancangan Perangkat Keras.....	36
3.3.2	Pembuatan Perangkat Keras.....	36
3.3.3	Perancangan dan Pembuatan Perangkat Lunak.....	37
3.4	Pengujian dan Perhitungan pada Perangkat Keras.....	38
3.5	Pengujian dan Perhitungan pada Mikrokontroler.....	38
3.6	Pengujian <i>Validitas</i> Sinyal Informasi.....	38
3.7	Analisis Data dan Pengambilan kesimpulan.....	38

BAB IV PERANCANGAN DAN PEMBUATAN ALAT

4.1	Spesifikasi Umum Alat.....	42
4.2	Penentuan Spesifikasi Umum Alat.....	42
4.3	Perancangan Blok Diagram Alat.....	43
4.4	Prinsip Kerja Alat.....	43
4.5	Perancangan dan Pembuatan Perangkat Keras.....	44
4.5.1	Perancangan Perangkat Keras.....	44
4.5.2	Pembuatan Perangkat Keras.....	45
4.5.2.a	Pembuatan Perangkat Keras.....	45
4.5.2.b	Reset Mikrokontroler.....	46
4.5.3	Perancangan <i>Generator Uji</i>	48
4.5.4	Perancangan Pendeteksi Beda Fasa.....	50
4.5.4.a	Rangkaian Proteksi Komparator.....	51
4.5.4.b	Rangkaian Komparator Beda Fasa.....	52

4.5.4.c	Perancangan Rangkaian <i>Integrator</i>	54
4.5.5	Perancangan Pendeteksi Sinyal / <i>Strength Meter</i>	57
4.5.6	Perancangan Analog Multiplekser	58
4.5.7	Perancangan Tampilan <i>7 Segment</i>	60
4.5.8	<i>Interface</i> Mikrokontroler dengan Komputer	61
4.6	Perancangan Perangkat Lunak	63
4.6.1	Perancangan Program <i>Assembly</i> AT89C51.....	63
4.6.2	Perancangan Program <i>Borland Delphi 7</i>	65

BAB V PENGUJIAN DAN ANALISA DATA

5.1	Pengujian Komunikasi Alat dengan Komputer	67
5.1.1	Tujuan	67
5.1.2	Peralatan yang Digunakan	67
5.1.3	Prosedur Pengujian	67
5.1.4	Hasil Pengujian	68
5.1.5	Analisis Hasil Pengujian	68
5.2	Pengujian Pengukuran Beda Fasa pada Alat	69
5.2.1	Tujuan	69
5.2.2	Peralatan yang Digunakan	69
5.2.3	Prosedur Pengujian	69
5.2.4	Hasil Pengujian	71
5.2.5	Analisis Hasil Pengujian	72
5.3	Pengujian Berdasarkan Jarak Peman car dan Penerima	72
5.3.1	Tujuan	72
5.3.2	Peralatan yang digunakan	72
5.3.3	Prosedur Pengujian	73
5.3.4	Hasil Pengujian	74
5.3.5	Analisis Hasil Pengujian	75
5.4	Pengujian Berdasarkan Tinggi Antena Pemancar	75
5.4.1	Tujuan	75
5.4.2	Peralatan yang Digunakan	75
5.4.3	Prosedur Pengujian	76
5.4.4	Hasil Pengujian	76
5.4.5	Analisis Hasil Pengujian	77

5.5	Pengujian Pengukuran Pemancar Radio FM di Malang	78
5.5.1	Tujuan	78
5.5.2	Peralatan yang Digunakan	78
5.5.3	Prosedur Pengujian	78
5.5.4	Hasil Pengujian	79
5.5.5	Analisis Hasil Pengujian	80

BAB VI PENUTUP

6.1	Kesimpulan	81
6.2	Saran	81

DAFTAR PUSTAKA

LAMPIRAN



DAFTAR TABEL

Tabel 2.1 Tabel Bentuk Program <i>Assembly</i>	25
Tabel 2.2 Tabel Kebenaran Gerbang EXOR	28
Tabel 2.3 Tabel Perbedaan <i>Serial Port</i> dan <i>Paralel Port</i>	29
Tabel 4.1 Tabel Kebenaran Gerbang EXOR	52
Tabel 4.2 Tabel Kebenaran Gerbang EXOR Perancangan	53
Tabel 4.3 Tabel Kebenaran AMUX 4051	59
Tabel 4.4 Tabel Alokasi Bit SCON	62
Tabel 4.5 Tabel Nilai dan Konfigurasi <i>Baud Rate</i>	62
Tabel 5.1 Tabel Perbandingan Posisi Kabel	72
Tabel 5.2 Tabel Perbandingan Berdasarkan Jarak Pancar Antena	75
Tabel 5.3 Tabel Perbandingan Berdasarkan Tinggi Antena Pemancar	77
Tabel 5.4 Tabel Perbandingan Kuat Sinyal Pemancar Radio di malang	80



DAFTAR GAMBAR

Gambar 2.1 Blok Diagram Sistem Pemancar FM	7
Gambar 2.2 Blok Diagram Pemancar FM	8
Gambar 2.3 Proses Modulasi Sinyal pada FM	11
Gambar 2.4 Filter <i>High Pass</i> Untuk Landaian Sebesar -20 dB/dekade	14
Gambar 2.5 Tanggapan Frekuensi Gambar 2.4	14
Gambar 2.6 ADC dengan Metode SAR A/D <i>Converter</i>	15
Gambar 2.7 Rangkaian Op Amp <i>Integrator</i> Sederhana	17
Gambar 2.8 Penguat Tak Membalik	18
Gambar 2.9 Rangkaian Komparator Pembanding dengan Referensi 0 volt	19
Gambar 2.10 Blok Diagram Mikrokontroler	20
Gambar 2.11 Mikrokontroler AT89C51	21
Gambar 2.12 Diagram Blok AT89C51	22
Gambar 2.13 Rangkaian AT89C51 Untuk <i>Running LED</i> dan <i>7 Segment</i>	23
Gambar 2.14 Bagan Kerja Proses <i>Assembly</i>	24
Gambar 2.15 Skema Multiplexer 2 <i>Input</i> dan pin out CD 4051	26
Gambar 2.16 Ilustrasi Multiplexer	27
Gambar 2.17 Simbol Gerbang EXOR	27
Gambar 2.18 Rangkaian Logika Gerbang EXOR	28
Gambar 2.19 IDE (<i>Integrated Development Environment</i>)	29
Gambar 2.20 Contoh <i>Desain Tampilan</i> pada <i>Delphi</i> dan <i>Form Tampilan</i>	31
Gambar 2.21 <i>7 Segment Indikator DT-I/O Cost MMI</i>	32
Gambar 2.22 <i>Port DB jantan</i> dan <i>Port DB Betina</i>	33
Gambar 4.1 Desain Perancangan Alat Pendeteksi Kualitas Sinyal Informasi	40
Gambar 4.2 Blok Diagram Sistem Deteksi Sinyal	43
Gambar 4.3 Rangkaian <i>Reset</i> Mikrokontroler	46
Gambar 4.4 Skema Rangkaian <i>Generator Uji</i>	48
Gambar 4.5 Blok Periode pada <i>Generator Uji</i>	49
Gambar 4.6 Rangkaian Komparator dalam Perancangan	51
Gambar 4.7 <i>Output HPF</i> pada Komparator	51
Gambar 4.8 Susunan Gerbang EXOR pada Rangkaian Komparator Beda Fasa	53
Gambar 4.9 Rangkaian <i>Integrator</i> yang digunakan	54

Gambar 4.10 <i>Estimasi</i> tegangan <i>Output Integrator</i>	56
Gambar 4.11 Blok Perancangan Rangkaian <i>Buffer</i>	57
Gambar 4.12 Skema Rangkaian <i>Buffer</i>	58
Gambar 4.13 Rangkaian AMUX 4051 yang Dirancang	59
Gambar 4.14 Rangkaian <i>Interface</i> Mikrokontroler dengan 7 <i>Segment</i>	60
Gambar 4.15 Rangkaian pada 7 <i>Segment</i>	61
Gambar 4.16 Diagram Alir Proses Perhitungan di Mikrokontroler	64
Gambar 4.17 Diagram Alir Proses Menampilkan Data di <i>Borland Delphi 7</i>	65
Gambar 5.1 Hasil Pengujian Komunikasi Alat dengan komputer	68
Gambar 5.2 Blok Diagram Pengujian Pengukuran Beda Fasa	69
Gambar 5.3 Posisi Kabel Sejajar ke Atas	70
Gambar 5.4 Posisi Kabel Menyilang	70
Gambar 5.5 Hasil Pengukuran pada Alat dan Gambar Sinyal pada <i>Osiloskop</i>	71
Gambar 5.6 Hasil Pengukuran pada Alat dan Gambar Sinyal pada <i>Osiloskop</i>	71
Gambar 5.7 Blok Pengujian Pengukuran Kuat Sinyal dan Beda Fasa	73
Gambar 5.8 Tampilan pada <i>Borland Delphi 7</i>	74
Gambar 5.9 Pengujian pada Alat dengan jarak pancar 300 cm	74
Gambar 5.10 Tampilan pada <i>Borland Delphi 7</i>	76
Gambar 5.11 Pengujian pada Alat dengan Tinggi Antena Pemancar 65 cm	77
Gambar 5.12 Blok Pengujian Pengukuran Kuat Sinyal dan Beda Fasa	78
Gambar 5.13 Besar Kuat Sinyal Andalus FM pada Tampilan <i>Borland Delphi 7</i> ...	79
Gambar 5.14 Besar Kuat Sinyal Andalus FM pada Alat	79



Abstraksi

Alat ini adalah sebuah sistem deteksi kualitas sinyal informasi pada Frekuensi Modulasi (FM). Alat ini mampu menampilkan besarnya kuat sinyal yang dipancarkan / *strength meter* dan beda fasa.

Alat ini berbasis pada mikrokontroler sebagai pengolah data, dan terbentuk dari beberapa sub sistem, antara lain pemancar mini 10 mW sebagai pemancar yang digunakan, rangkaian *generator* uji sebagai pembangkit sinyal untuk pengujian, rangkaian proteksi dan komparator sebagai rangkaian proteksi pengemudian lebih (*over drive*), dan FM *tuner* sebagai penerima.

Dalam pengujian alat ini menampilkan data di 7 *segment* dan juga di komputer sistem dalam bentuk tampilan *database Borland Delphi 7*. Pengujian yang dilakukan meliputi pengujian berdasarkan jarak pancar antara pemancar dan penerima, pengujian berdasarkan tinggi antenna pemancar, dan pengujian pada pemancar radio FM yang berada di Malang dan sekitarnya.

Hasil dalam pengujian berdasarkan jarak pancar antara pemancar dan penerima diperoleh level tegangan antara 78-226 V dengan beda fasa antara 1-79⁰, sedangkan pada pengujian berdasarkan tinggi antenna pemancar diperoleh level tegangan antara 127-227 V dengan beda fasa antara 0-3⁰, dan pada pengujian pengukuran pada pemancar radio FM di Malang dan sekitarnya diperoleh level tegangan 95-229 V dengan tidak dilakukan pengukuran beda fasa karena tidak ada *sinkronisasi* antara sinyal yang dipancarkan dengan sinyal yang diterima. Dari pengujian didapatkan bahwa semakin jauh jarak antara pemancar dan penerima, maka semakin kecil level tegangan kuat pancar sinyal / *strength meter* dan semakin besar beda fasanya.

Dan hasilnya alat ini dapat bekerja dengan mendeteksi besarnya kuat sinyal yang dipancarkan / *strength meter* serta beda fasa dari suatu sistem transmisi, berdasarkan jarak dan tinggi antenna pemancar.

Kata kunci : kuat sinyal / *strength meter*, beda fasa, FM (Frekuensi Modulasi).

BAB I PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan teknologi maka piranti elektronika yang berperan paling penting dalam kehidupan juga akan ikut semakin berkembang, salah satu piranti tersebut adalah pemancar radio. Pemancar radio yang paling populer hingga saat ini adalah pemancar radio FM. Karena pemancar radio FM mempunyai banyak kelebihan dibanding pemancar radio yang lain. Di antara keuntungan FM adalah bebas dari pengaruh gangguan udara, *bandwidth* (lebar pita) yang lebih besar, dan *fidelitas* yang tinggi. Jika dibandingkan dengan sistem Pemancar Radio AM.

Selain kebutuhan sarana dan prasarana komunikasi, kualitas dari komunikasi sendiri sudah menjadi kebutuhan pokok bagi pengguna. Salah satu contohnya adalah kebutuhan akan kualitas penerimaan sinyal informasi yang diterima. Semakin tingginya persaingan di bidang telekomunikasi, maka sudah seharusnya diberikan pelayanan yang terbaik bagi pengguna. Disamping itu perkembangan industri yang semakin meningkat sangat berhubungan dengan dunia telekomunikasi, hal ini dibuktikan dengan semakin banyaknya iklan dari suatu industri yang menjadi suatu bagian dari telekomunikasi, dimana dalam penyampaiannya memang harus baik, sehingga dapat diterima dengan jelas oleh pendengar.

Perancangan sistem deteksi sinyal suara (*audio*) pada modulasi FM (Frekuensi Modulasi) dengan menggunakan mikrokontroler AT89C51 dapat memberikan solusi yang tepat terhadap masalah kualitas sinyal informasi. Ditambah dengan dukungan teknologi mikrokontroler yang memungkinkan dibentuknya sebuah sistem yang memiliki efisiensi biaya, dan tingkat ketelitian yang baik untuk menentukan kualitas sinyal informasi yang diterima.

Alat yang sudah ada dan digunakan saat ini di beberapa perusahaan radio yang menggunakan modulasi FM, adalah spektrum *analyzer*. Spektrum *analyzer* merupakan alat yang digunakan untuk mendeteksi *bandwidth* frekuensi. Alat ini hanya digunakan untuk mengetahui berapa *range* frekuensi yang ada di suatu daerah tertentu. Alat ini juga tidak mendeteksi bagaimana kualitas suara (*audio*) yang diterima, dan hasil yang muncul pada alat ini berbentuk gambar (gelombang *sinusioda*).

Pada skripsi ini akan dirancang sistem deteksi sinyal suara (*audio*) pada modulasi FM (Frekuensi Modulasi) dengan menggunakan mikrokontroler AT89C51. Sistem yang dirancang disini yaitu sistem deteksi sinyal suara (*audio*), dimana dalam sistem akan menghubungkan antara penerima dengan sistem mikrokontroler sebagai tempat sistem, dan komputer sebagai media pembaca kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa. Sinyal suara (*audio*) yang diterima pada sistem mikroprosesor akan diteruskan ke komputer untuk kemudian akan ditampilkan besar kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa dari sinyal suara (*audio*) tersebut. Kelebihan dari alat ini adalah besaran nilai yang ditampilkan dalam bentuk angka, sehingga lebih mudah terbaca dan lebih akurat dalam perhitungannya di media komputer.

Sebelum menentukan pilihan judul ini, ada beberapa aspek yang menentukan. Salah satunya adalah dengan melakukan survei kepada salah satu perusahaan penyiaran radio di kota Malang (Radio Republik Indonesia). Alasan dari survei ini adalah untuk melakukan perbandingan antara alat pendeteksi yang sudah ada dengan perancangan alat yang akan dilakukan. Dari survei tersebut didapatkan beberapa perbedaan karakteristik dari alat yang sudah ada dengan alat yang akan dirancang, dengan rincian sebagai berikut :

- **Alat di Radio Republik Indonesia (RRI) Stasiun Malang**

- a. Alat pendeteksi yang digunakan adalah Spektrum Analyzer 2650 BK-PRECISION by micronix. 3,3 GHz.
- b. Merupakan alat pendeteksi yang digunakan untuk mengukur *bandwidth* frekuensi (dB), bukan untuk mendeteksi kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa.
- c. Tampilan yang muncul pada alat ini berbentuk gambar (gelombang *sinusioda*).

- **Alat Yang Akan Dirancang**

- a. Alat pendeteksi yang akan dirancang merupakan alat yang berfungsi membaca langsung sinyal suara (*audio*) yang diterima dan prosesnya berbasis pada mikrokontroler AT89C51.
- b. Merupakan alat pendeteksi yang digunakan untuk mengukur sinyal suara (*audio*) yang diterima pada penerima dan kemudian digunakan untuk mendeteksi kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa.

- c. Tampilan yang muncul pada alat ini berbentuk *digit* (angka).

1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang, maka permasalahan yang akan dibahas adalah:

- a. Bagaimana merancang dan membuat alat pengukur sinyal suara (*audio*) pada modulasi FM (Frekuensi Modulasi) dengan menggunakan mikrokontroler AT89C51.
- b. Mengetahui kualitas sinyal suara (*audio*) yang diterima oleh penerima, dengan cara mendengarkan kualitas suara (*audio*) yang diterima.
- c. Mengetahui layak atau tidaknya informasi yang diterima oleh penerima dengan parameter perhitungan dari kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa.

1.3 Ruang Lingkup

Agar pembahasan lebih terarah, maka pembahasan dibatasi dalam beberapa hal berikut:

- a. Pada proses pengiriman sinyal menggunakan sistem pengiriman sinyal pada FM (Frekuensi Modulasi).
- b. Mikrokontroler yang digunakan adalah AT89C51.
- c. *Range* frekuensi sinyal informasi yang dapat diproses oleh alat antara 88 -108 MHz.
- d. Sinyal informasi yang dikirim harus *stabil* (tidak berubah – ubah). Pada kasus ini akan dipakai frekuensi yang sesuai dengan kebutuhan pada saat pengujian dan analisa data.
- e. Sinyal informasi dimodulasi menggunakan metode FM.
- f. Pemancar FM yang digunakan adalah pemancar FM mini 10 mili watt.
- g. Catu daya yang digunakan pada mikrokontroler sistem adalah sebesar 5 volt.

1.4 Tujuan

Tujuan penyusunan skripsi ini adalah melakukan pengukuran kualitas sinyal suara (*audio*) yang diterima oleh penerima, kemudian untuk mengetahui kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa. yang dihasilkan, untuk selanjutnya menentukan apakah informasi layak diterima atau tidak.

1.5 Manfaat

Manfaat yang diharapkan dapat diperoleh melalui pengerjaan skripsi ini adalah :

1.5.1 Bagi Penyusun

- Penyusun dapat merancang alat untuk mengukur alat pendeteksi sinyal suara (*audio*) pada modulasi FM.
- Memperoleh pemahaman mengenai perancangan yang dilakukan dan pengembangan menggunakan Mikrokontroler AT89C51.
- Menambah bekal pengetahuan dan wawasan baru serta dapat belajar memecahkan masalah dengan efisien dan berfikir secara lebih kritis dan tepat dalam membuat sebuah keputusan.
- Mengimplementasikan ilmu yang telah didapat di masa mendatang.

1.5.2 Bagi Pengguna

- Memberikan kemudahan bagi orang lain untuk mendeteksi sinyal suara (*audio*) pada modulasi FM.

1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini adalah sebagai berikut:

BAB I : PENDAHULUAN

Berisi tentang : latar belakang, rumusan masalah, ruang lingkup, tujuan penulisan, dan manfaat penulisan skripsi.

BAB II : TINJAUAN PUSTAKA

Membahas mengenai dasar teori yang digunakan dalam perancangan sistem pendeteksi sinyal suara (*audio*) pada modulasi FM (*Frekuensi Modulasi*) dengan menggunakan Mikrokontroler AT89C51.

BAB III : METODOLOGI PENELITIAN

Memuat tentang metode yang digunakan dalam menyelesaikan permasalahan yang dibahas dalam skripsi ini.

BAB IV : PERANCANGAN DAN PEMBUATAN SISTEM

Berisikan pembahasan mengenai perancangan sistem pendeteksi sinyal suara (*audio*) pada modulasi FM (*Frekuensi Modulasi*) dengan menggunakan Mikrokontroler AT89C51, bentuk antarmuka rangkaian

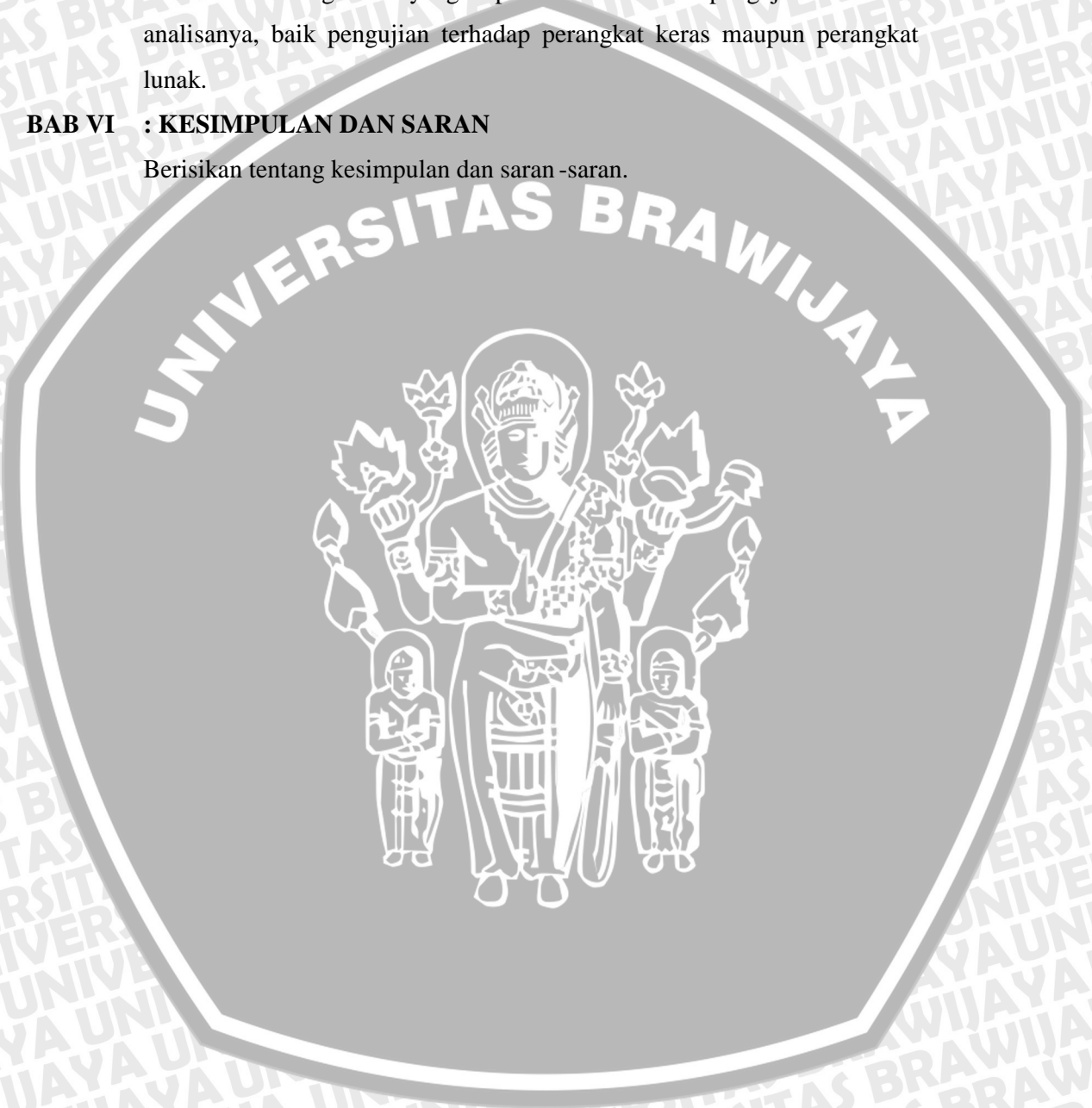
yang digunakan, perbandingan dan perhitungan rangkaian yang terdapat dalam sistem, dan membuat perancangan sistem sehingga diperoleh efisiensi (miniaturisasi alat dan penyederhanaan sistem).

BAB V : PENGUJIAN DAN ANALISA DATA

Berisikan tentang data yang diperoleh dari hasil pengujian alat dan analisisnya, baik pengujian terhadap perangkat keras maupun perangkat lunak.

BAB VI : KESIMPULAN DAN SARAN

Berisikan tentang kesimpulan dan saran-saran.



BAB II

TINJAUAN PUSTAKA

Dalam merencanakan dan merealisasikan perancangan alat ini dibutuhkan pemahaman tentang beberapa dasar teori yang digunakan. Pemahaman ini akan bermanfaat untuk merancang perangkat keras dan perangkat lunak sistem. Adapun beberapa dasar teori yang mendukung perencanaan dan realisasi alat meliputi:

1. Teori Mengenai FM (*Frekuensi Modulasi*).
2. Pengertian *Signal to Noise Ratio* (SNR).
3. Teori *High Pass Filter* (HPF).
4. Teori Tentang *Analog Digital Converter* (ADC).
5. Rangkaian *Integrator*.
6. Teori Penguat Tak Membalik (*Non Inverting Amplifier*).
7. Teori Pembanding Tak Membalik (*Non Inverting Comparator*).
8. Mikroprosesor AT89C51.
9. Pengertian Bahasa *Assembly*.
10. Teori Tentang Multiplexer CD4051.
11. Pengertian Gerbang EXOR.
12. *Borland Delphi 7*.
13. Teori tentang RS 232.
14. Pengertian Tentang *7 Segment* dan *Serial Port DB 9*.

2.1 FM (Frekuensi Modulasi)

2.1.1 Siaran FM

Siaran FM terjadi pada jalur VHF dari lebar dasar utama yang sekarang banyak digunakan adalah 88 Mhz sampai dengan 108 Mhz dengan deviasi maksimum yang diinginkan sebesar kurang lebih 75 kHz. Jarak saluran adalah 200 kHz dan keluaran-keluaran daya yang dipakai dapat mencapai 200 KW.

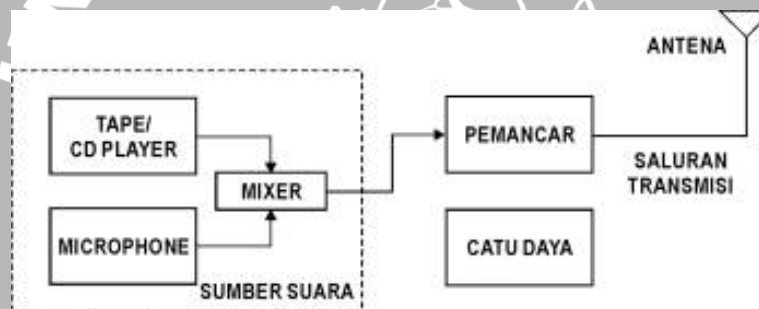
Hampir semua sistem FM menggunakan program *stereo* dua saluran, selanjutnya beberapa stasiun FM masih memultiplek pembagian frekuensi suatu saluran tambahan pada gedung-gedung yang sifatnya umum. Sistem ini diberi lisensi sebagai suatu otoritas komunikasi bantuan SCA. *Subscidary Communication Authorization* (SCA) adalah pita atau *band* sisa yang terdapat pada lebar pita sinyal

yang dimodulasi frekuensi (FM). Pita SCA biasanya digunakan untuk keperluan - keperluan seperti pemberian musik latar belakang pada gedung atau rumah.

Siaran SCA adalah siaran atau program tambahan yang dipancarkan bersama dengan siaran utama stasiun pemancar FM komersial, pesawat penerima yang dilengkapi dengan piranti untuk menerima transmisi siaran SCA akan meloloskan sinyal *detektor* FM menuju bagian dekoder khusus SCA.

2.1.2 Pemancar FM

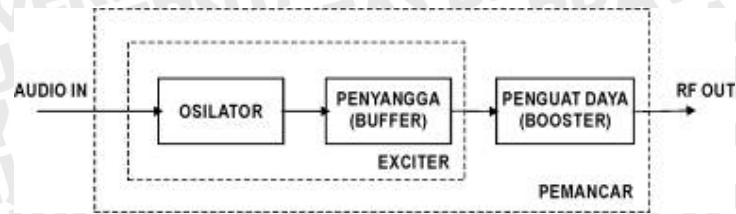
Pemancar adalah alat yang berfungsi untuk memproses dan membawa sinyal informasi untuk ditransmisikan, proses ini dilakukan dengan menumpangkan siny al pembawa yang memiliki frekuensi lebih tinggi dari pada sinyal informasi. Pemancar FM adalah pemancar yang mentransmisikan sinyal yang dimodulasi frekuensinya.



Gambar 2.1 Blok diagram sistem pemancar FM.

Sumber: Anonymous, Owner Manual, 1997.

Sumber suara yang dapat digunakan bermacam-macam. *Tape*, *CD-player*, *mp3-player*, *microphone* bahkan radio juga dapat dipakai. Segala jenis catu daya juga dapat dipakai pada sistem pemancar FM asalkan catu daya tersebut bisa menghasilkan tegangan yang sesuai dan arus yang cukup. Bagian yang penting dari sistem pemancar FM adalah antena, saluran transmisi dan pemancar itu sendiri. Pemancar FM secara umum terdiri dari blok dalam Gambar 2.2.



Gambar 2.2 Blok diagram pemancar FM.

Sumber: Anonymous, Owner Manual, 1997.

- Osilator : Inti dari sebuah pemancar adalah osilator. Untuk dapat membangun sistem komunikasi yang baik harus dimulai dengan osilator yang dapat bekerja dengan sempurna. Pada sistem komunikasi, osilator menghasilkan gelombang sinus yang dipakai sebagai sinyal pembawa. Sinyal informasi kemudian ditumpangkan pada sinyal pembawa dengan proses modulasi. Osilator dengan frekuensi yang bisa dirubah disebut VFO (*Variable Frequency Oscillator*). VFO memiliki kelebihan pada deviasi frekuensinya yang lebar. Untuk menghasilkan frekuensi 88MHz – 108MHz dapat dipakai VFO. Karena pada VFO dipakai induktor dan kapasitor sebagai penentu frekuensinya maka kestabilan VFO sangat tergantung dari kestabilan nilai induktor dan kapasitor. Komponen-komponen pada VFO yang mudah terpengaruh oleh suhu menyebabkan VFO mempunyai kestabilan yang rendah.
- Penyangga : Semua jenis osilator membutuhkan penyangga. Penyangga berfungsi untuk menstabilkan frekuensi atau amplitudo osilator akibat dari pembebanan tingkat selanjutnya. Biasanya penyangga terdiri dari 1 atau 2 tingkat penguat transistor yang dibias sebagai kelas A. Dengan penguat kelas A akan didapatkan penguatan dan linearitas yang tinggi meskipun demikian penguat kelas A memiliki efisiensi yang paling rendah dibandingkan kelas yang lain. Osilator yang dilengkapi dengan penyangga biasanya disebut sebagai *exciter*. Dan *exciter* sebenarnya sudah bisa dipakai sebagai pemancar FM dengan daya yang relatif kecil.
- Penguat daya : Sinyal yang didapat dari *exciter* masih relatif lemah. Untuk mendapatkan daya yang lebih besar dibutuhkan penguat daya

frekuensi radio. Parameter-parameter yang perlu diperhatikan pada penguat daya frekuensi radio adalah : *bandwidth* dan faktor kualitas, penguatan tiap tingkat dan daya *input output* tiap tingkat, impedansi *input* dan *output* tiap tingkat, linearitas dan efisiensi.

- Saluran : Daya yang dihasilkan oleh pemancar akan diradiasikan oleh antenna.

Transmisi Saluran transmisi adalah bagian yang menghantarkan daya yang dihasilkan pemancar ke antenna. Sebagai bagian yang menghantarkan daya, saluran transmisi yang ideal tidak akan mengurangi daya yang dihantarkannya dan juga tidak meradiasikan daya yang menjadi tugas antenna.

- Antena : Antena adalah bagian yang paling penting dari sistem pemancar. Antena berfungsi sebagai alat yang dapat meradiasikan gelombang radio. Sebagai bagian dari sistem penerima, antena berfungsi sebagai bagian yang dapat menangkap radiasi gelombang radio. Antena yang ideal akan meradiasikan gelombang radio kesegala arah. Antena yang ideal disebut sebagai antena isotropis. Sebagai gambaran, jika antena isotropis diletakkan pada titik pusat dari bola maka antena isotropis akan mengisi semua ruang yang ada pada bola tersebut dengan radiasi gelombang radio.

2.1.3 Modulasi

Modulasi dapat diartikan sebagai proses dimana beberapa karakteristik dari gelombang dengan frekuensi relatif tinggi, yang dinamakan pembawa (*carrier*), berubah sehubungan dengan harga sesaat dari gelombang frekuensi rendah, yang dinamakan gelombang pemodulasi. Modulasi sangat penting dalam komunikasi radio. Disini yang dimaksud informasi adalah pembicaraan atau musik yang didengar merupakan sinyal pemodulasi.

Biasanya pembawa yang dimodulasi berbentuk *sinusoidal*, dan secara umum digambarkan oleh :

$$e_c = E_c \cos \phi(t) \quad (2.1)$$

dimana e_c merupakan harga sesaat, E_c adalah amplitudo, dan $\phi(t)$ adalah perpindahan sudut total selama t .

Kecepatan sudut sesaat $\omega(t)$ didefinisikan sebagai laju kenaikan sesaat perpindahan sudut $\phi(t)$, yaitu :

$$\omega(t) = \frac{d\phi(t)}{dt} \quad (2.2)$$

Dari persamaan di atas didapatkan :

$$\phi(t) = \int \omega(t) dt \quad (2.3)$$

dimana θ adalah konstanta integrasi. Dalam suatu pembawa tidak termodulasi, $\omega(t)$ biasanya tetap, dan dinyatakan oleh ω_c . Sehingga, untuk pembawa tidak termodulasi dari persamaan tersebut dengan mengganti $\omega(t)$ oleh ω_c , didapatkan :

$$\phi(t) = \omega_c t + \theta \quad (2.4)$$

Jika $t = 0, \theta = \phi(0)$, yaitu θ menggambarkan sudut fase awal, didapatkan persamaan sebagai berikut :

$$e_c = E_c \cos(\omega_c t + \theta) \quad (2.5)$$

Jika waktu t diukur dari titik acuan di mana $\theta = 0$, didapatkan :

$$e_c = E_c \cos \omega_c t \quad (2.6)$$

Pada dasarnya modulasi dapat dihasilkan dari dua cara, yaitu dengan mengubah amplitudo E_c dari pembawa, dan dengan mengubah sudut fasa total $\theta(t)$. Dua proses ini dinamakan modulasi amplitudo dan modulasi sudut. Modulasi sudut sendiri dibagi menjadi dua, yaitu modulasi frekuensi dan modulasi fasa.

2.1.4 Modulasi Frekuensi (FM)

Sesuai dengan namanya yaitu Frekuensi Modulasi berarti pada proses modulasi ini hanya frekuensi dari sinyal pembawa (*carrier*) saja yang berubah-ubah dan bukan pada amplitudo maupun fasanya. Banyaknya perubahan dari frekuensi tersebut akan ditentukan oleh amplitudo dari sinyal yang termodulasi mengingat bahwa banyaknya perubahan yang terjadi akan ditentukan oleh frekuensi dari sinyal termodulasi itu.

Dalam *carrier* FM, sinyal-sinyal informasi yang dibawa sangat bervariasi yang tergantung dari pada frekuensinya. Jika amplitudonya berubah maka frekuensi dari *carrier* yang termodulasi itu akan berubah juga, begitu pula sebaliknya. Frekuensi itu akan berada pada frekuensi tinggi pada saat amplitudo sinyalnya paling rendah jika sinyal amplitudonya mencapai nilai negatif maksimum. Pada saat nilai

amplitudo sinyalnya sama dengan nol, maka frekuensi *carrier*-nya berada pada posisi normal. Sehingga frekuensi sudut sesaat dari pembawa termodulasi frekuensi (FM) dapat dinyatakan dalam persamaan 2.7 berikut :

$$\omega(t) = \omega_c + K_f E_m \cos \omega_m t \quad (2.7)$$

di mana K_f adalah konstanta perbandingan dan menyatakan faktor perubahan frekuensi, dan besarnya tergantung dari sistem pemodulasi.

Persamaan untuk gelombang FM adalah :

$$(e_c)_{FM} = E_c \cos \left(\omega_c t + \frac{K_f E_m}{\omega_m} \sin \omega_m t \right) \quad (2.8)$$

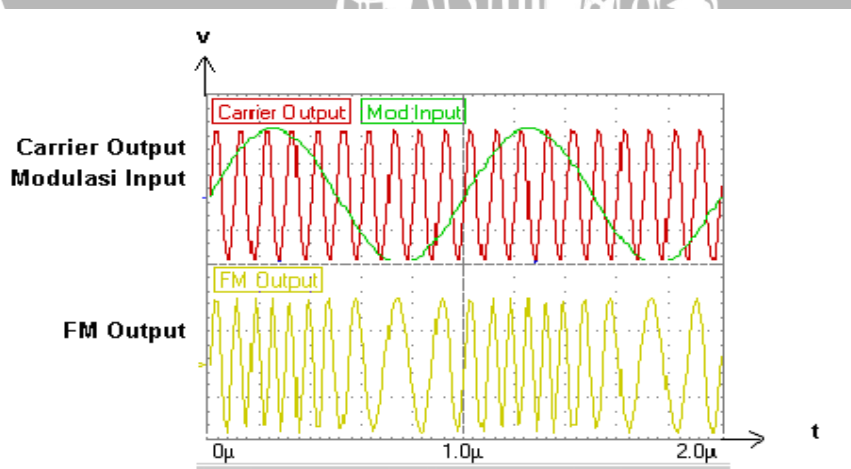
di mana frekuensi sesaat dari FM didapatkan dari :

$$f(t) = \frac{\omega(t)}{2\pi} = f_c + \frac{K_f E_m}{2\pi} \cos \omega_m t \quad (2.9)$$

karena itu, simpangan frekuensi maksimum dari harga tanpa modulasi diberikan oleh :

$$(\Delta f)_{\max} = [f(t) - f(c)]_{\max} = \frac{K_f E_m}{2\pi} \quad (2.10)$$

Dari persamaan di atas diperoleh bahwa simpangan maksimum $(\Delta f)_{\max}$ berbanding lurus dengan amplitudo sinyal pemodulasi, tetapi tidak tergantung pada frekuensi. Perlu diingat, bahwa walaupun frekuensi FM berubah menurut waktu, besar amplitudonya tetap. Pada gambar di bawah ini ditunjukkan pembawa termodulasi frekuensi bersama dengan sinyal pemodulasinya, sedangkan frekuensi pemodulasi menentukan laju perubahan frekuensi.



Gambar 2.3 Proses modulasi sinyal pada FM.

Sumber: Dasar Elektronika, D.Chattopadhyay, Calcutta University, 1989.

Dalam modulasi frekuensi, sudut fase sesaat $\phi(t)$ mengalami simpangan menurut waktu. Simpangan maksimum sudut fase dalam gelombang FM adalah :

$$[(\Delta\phi_{maks})]_{FM} = \frac{K_f E_m}{\omega_m} = \frac{K_f E_m}{2\pi f_m} \quad (2.11)$$

dapat ditunjukkan bahwa gelombang termodulasi frekuensi, terdiri dari sejumlah besar (tidak terhingga) komponen frekuensi di tiap sisi pembawa. Frekuensi sisi tersebut adalah $(f_c \pm f_m)$, $(f_c \pm 2f_m)$, $(f_c \pm 3f_m)$, dan seterusnya. Amplitudo komponen frekuensi sisi tergantung pada *indeks* modulasi dan menjadi terabaikan untuk frekuensi-frekuensi sisi cukup jauh dari frekuensi pembawa. Karena itu dalam praktek, jumlah pasangan frekuensi sisi terbatas, seperti dalam gelombang AM, informasi berisi dalam komponen frekuensi sisi. Sehingga lebar pita yang diperlukan untuk penyaluran atau penerimaan lewat modulasi didapatkan dari :

$$(BW)_{FM} = 2nf_m \quad (2.12)$$

di mana n adalah jumlah pasangan frekuensi yang terlihat dan f_m menggambarkan pemodulasi tertinggi.

Dari penjelasan sebelumnya didapatkan, bahwa daya dalam gelombang FM tidak berubah karena modulasi. Daya dalam komponen frekuensi sisi diperoleh dengan berkurangnya daya dari komponen pembawa. Jadi, daya total yang terdiri dari daya pembawa dan daya dalam komponen frekuensi tidak berubah. Hal ini berbeda dari peristiwa modulasi amplitudo di mana daya total membesar dengan kedalaman modulasi.

2.2 Signal to Noise Ratio (SNR)

Noise merupakan fluktuasi *random* dari perulangan pengukuran yang terjadi pada suatu basis kontinyu atau merupakan standar deviasi dari suatu sinyal instrumen. Sedangkan *signal to noise ratio* adalah perhitungan dari besarnya nilai *noise* yang terdapat diantara pemancar FM dan penerimanya.

$$\frac{S}{N} = \frac{P_s}{P_n} = \frac{E_s^2 / R}{E_n^2 / R} = \left(\frac{E_s}{E_n} \right)^2 \quad (2.13)$$

Persamaan 2.13 merupakan perbandingan dari sinyal dan *noise* yang berada dalam suatu proses pengiriman sinyal informasi.

2.2.1 Noise Figure

Noise figure adalah perbandingan pada penerima atau pemancar yang bekerja pada tingkat impedansi yang berbeda.

$$F = \frac{S_i / N_i}{S_o / N_o} \quad (2.14)$$

S_i = Besar sinyal pada *input* sistem.

S_o = Besar sinyal pada *output* sistem.

N_i = Noise pada *input* sistem.

N_o = Noise pada *output* sistem.

F = Besar *noise figure*.

2.3 High Pass Filter (HPF)

Gambar 2.4 adalah filter *high pass* yang umum digunakan. Tujuan dari digunakannya rangkaian *high pass filter* adalah untuk mencegah tegangan DC masuk (DC blocking). Untuk mencari nilai V_o digunakan persamaan sebagai berikut :

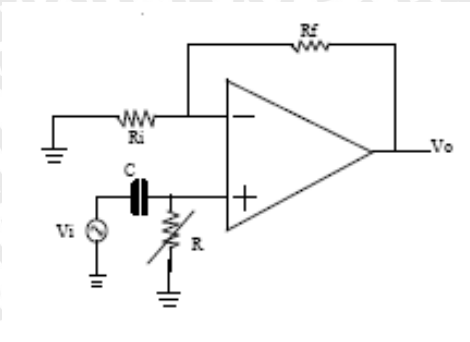
$$V_o = \left(1 + \frac{R_f}{R_1}\right) \cdot \frac{j2\pi fRC}{1 + j2\pi fRC} \cdot V_{in} \quad (2.15)$$

frekuensi sendiri ditentukan dalam satuan radian per detik ($2\pi f = \omega$) dan j sama dengan $\sqrt{-1}$, dengan demikian akan didapatkan :

$$A_f = 1 + \frac{R_f}{R_1} \quad (2.16)$$

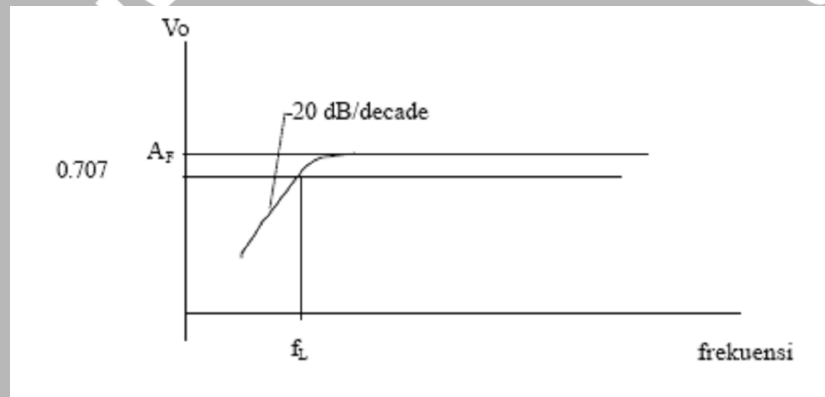
$$f_L = \frac{1}{2\pi RC} \quad (2.17)$$

Sesuai dengan persamaan (2.17), bahwa pada *high pass filter* nilai dari A_f selalu berubah-ubah sesuai dengan frekuensi. Pada frekuensi-frekuensi yang sangat rendah, yaitu ketika ω mendekati 0 maka $A_{CL} = 0$, dan pada frekuensi yang sangat tinggi, yaitu ketika ω mendekati tak berhingga maka $A_{CL} = 1$.



Gambar 2.4 Filter *high pass* untuk landaian sebesar -20 db/dekade.

Sumber: Penguat Operasional dan Rangkaian Terpadu Linear, Robert F. Coughlin, Frederick F. Driscoll, 1992.



Gambar 2.5 Tanggapan frekuensi untuk Gambar 2.4.

Sumber: Penguat Operasional dan Rangkaian Terpadu Linear, Robert F. Coughlin, Frederick F. Driscoll, 1992.

Gambar 2.5 merupakan perbandingan A_F dengan f dan memperlihatkan bahwa untuk frekuensi-frekuensi yang lebih besar dari frekuensi *cutoff* f_L , A_F naik sebesar 20 dB/dekade.

Dengan melihat Gambar 2.5 didapatkan :

$$\left| \frac{V_o}{V_i} \right| = \frac{A_F \left(\frac{f}{f_L} \right)}{\sqrt{1 + \left(\frac{f}{f_L} \right)^2}} \quad (2.18)$$

Keterangan persamaan 2.18 :

V_o = Tegangan *output* (volt)

V_i = Tegangan *input* (volt)

A_F = Penguatan filter

f = Frekuensi yang digunakan (Hz)

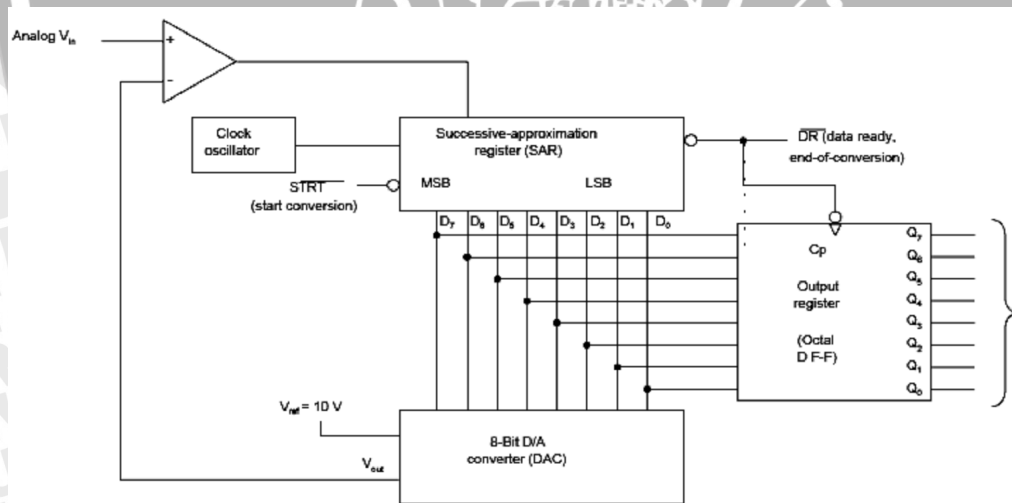
f_L = Frekuensi *cut off* (Hz)

2.4 ADC tipe SAR (*Successive Approximation Register*)

Analog to Digital Converter (ADC) dibutuhkan apabila akan diproses sinyal-sinyal analog. Misal dari fenomena alam (cahaya, suhu, suara, dan lain-lain), dengan bantuan sensor/*transducer* supaya dihasilkan tegangan. Terdapat 2 rangkaian dasar analog to digital (ADC), yaitu:

- ADC jenis *counter ramp*
- ADC jenis SAR (*Successive Approximation Register*)

Karena yang akan digunakan dalam pembahasan perancangan ini adalah ADC jenis SAR (*Successive Approximation Register*), maka pembahasan akan lebih ditekankan pada ADC jenis ini.



Gambar 2.6 ADC dengan metode SAR A/D converter.

Sumber: Soclof, Sidney, *Design and Applications of Analog Integrated Circuits*, Prentice Hall, 1971.

Metode SAR ini lebih cepat yaitu dengan memakai konfigurasi, mengeluarkan kombinasi bit MSB = 1 \geq 10000000, apabila belum sama (kurang dari tegangan analog

input) maka bit MSB berikutnya $= 1 \geq 11000000$, apabila tegangan analog *input* ternyata lebih kecil dari tegangan yang dihasilkan DAC maka langkah berikutnya menurunkan kombinasi bit ≥ 10100000 dan seterusnya.

Misalkan diberikan tegangan analog *input* sebesar 6.84 volt dan tegangan referensi ADC sebesar 10 volt, sehingga didapatkan:

D7=1, Vout =5 volt

D6=1, Vout =2,5 volt

D5=1, Vout =1,25 volt

D4=1, Vout =0,625 volt

dst...

D0=1, Vout =0,0390625 volt

Analog to digital converter dalam bentuk IC *chip A/D converter* yang banyak digunakan serta tersedia di pasaran adalah jenis:

ADC 0808 dibuat dengan teknologi CMOS.

2.5 Rangkaian *Integrator*

Rangkaian *integrator* op-amp berasal dari rangkaian *inverting* dengan tahanan umpan baliknya diganti dengan kapasitor. Proses perhitungannya sebagai berikut:

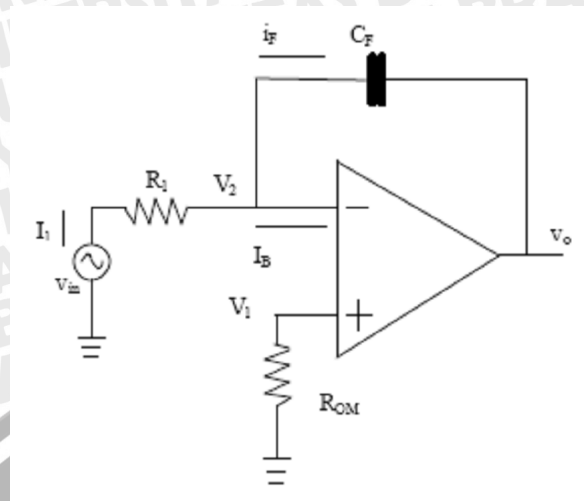
$$i_1 = I_B + i_F \quad (2.19)$$

nilai I_B diabaikan karena sangat kecil nilainya sehingga didapatkan :

$$i_1 = i_F \quad (2.20)$$

Pada Gambar 2.7 ditunjukkan rangkaian *integrator* yang belum di tambah tahanan yang

diparalel dengan kapasitor. Biasanya untuk aplikasi pada rangkaian ada penambahan tahanan yang diparalel dengan kapasitor.



Gambar 2.7 Rangkaian op amp *integrator* sederhana.

Sumber: Penguat Operasional dan Rangkaian Terpadu Linear, Robert F. Coughlin, Frederick F. Driscoll, 1992.

Batas frekuensi yang dilewatkan pada kapasitor dalam rangkaian *integrator* adalah sebagai berikut :

$$f_c = \frac{1}{2\pi R_1 C_F} \quad (2.21)$$

sedangkan untuk mencari tegangan keluaran dari rangkaian *integrator* adalah :

$$V_o = -\frac{1}{R_1 C_F} \int_0^t V_{in} dt + C \quad (2.22)$$

Keterangan persamaan 2.21 dan 2.22 :

f_c = Frekuensi yang melalui kapasitor (Hz)

V_o = Tegangan *output* (volt)

V_{in} = Tegangan *input* (volt)

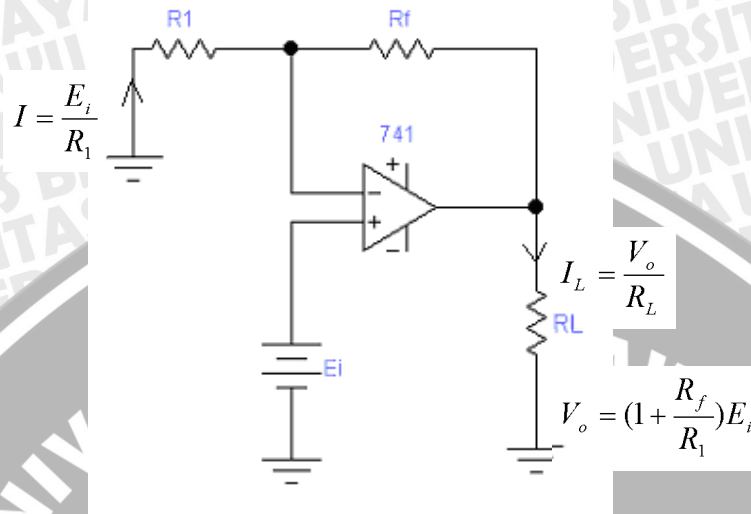
R_1 = Resistansi (Ω)

C = Kapasitansi (F)

2.6 Penguat Tak Membalik (*Non Inverting Amplifier*)

Penguat tak membalik merupakan suatu penguat dimana tegangan keluarannya atau V_o mempunyai polaritas yang sama dengan tegangan masukan atau E_i . Tahanan masukan dari penguat pembalik adalah R_1 , tetapi tahanan masukan dari penguat tak

membalik sangat besar, biasanya melebihi $100 \text{ M}\Omega$. Rangkaian penguat tak membalik ditunjukkan pada Gambar 2.8



Gambar 2.8 Penguat tak membalik.

Sumber: Penguat Operasional dan Rangkaian Terpadu Linear, Robert F. Coughlin, Frederick F. Driscoll, 1992.

Pada Gambar 2.8 arus i mengalir ke R_i karena impedansi masukan op amp sangat besar sehingga tidak ada arus yang mengalir pada kedua terminal masukannya. Tegangan pada R_i sama dengan E_i karena perbedaan tegangan pada kedua terminal masukannya mendekati 0 V.

$$i = \frac{E_i}{R_i}$$

(2.23)

Tegangan yang melintasi R_f dinyatakan oleh V_{Rf} dan dinyatakan sebagai :

$$V_{Rf} = I.R_f = \frac{R_f}{R_i} \times E_i \quad (2.24)$$

Tegangan keluaran V_o didapat dengan menambahkan tegangan pada R_i yaitu E_i dengan tegangan pada R_f yaitu V_{Rf} , dan didapatkan :

$$V_o = E_i + \frac{R_f}{R_i} \cdot E_i \quad (2.25)$$

atau

$$V_o = \left(1 + \frac{R_f}{R_i} \right) \cdot E_i \quad (2.26)$$

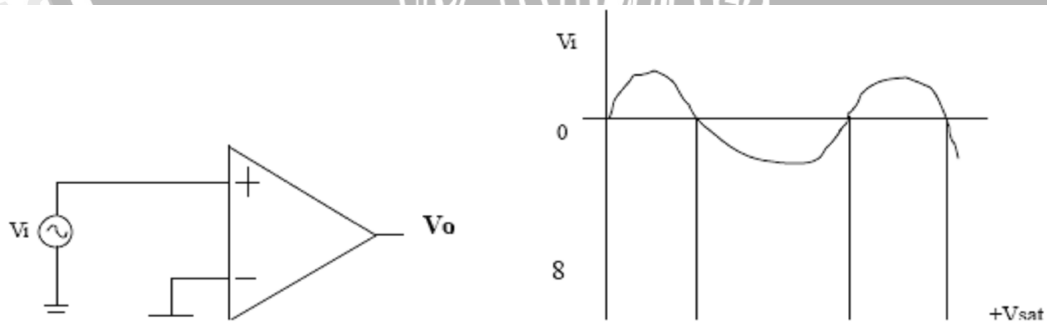
dan untuk menyusun *gain* tegangannya, didapatkan :

$$A_{CL} = \frac{V_o}{E_i} = 1 + \frac{R_f}{R_i} \quad (2.27)$$

dari persamaan 2.27 memperlihatkan bahwa gain tegangan dari sebuah penguat tak membalik menyamai besarnya gain sebuah penguat pembalik $\left(\frac{R_f}{R_i} \right)$ ditambah 1.

2.7 Pembanding Tak membalik (*Non Inverting Comparator*)

Rangkaian komparator bekerja sebagai sebuah pembanding. Rangkaian pembanding dengan 1 op amp tanpa jendela *input*, artinya rangkaian komparator/pembanding yang langsung dibandingkan. Seperti pada Gambar 2.9 yang merupakan komparator biasa dan hasilnya langsung dibandingkan dengan referensinya. Rangkaian komparator dengan jendela input, rangkaiannya hampir sama dengan rangkaian *non inverting* hanya saja parameternya terbalik.



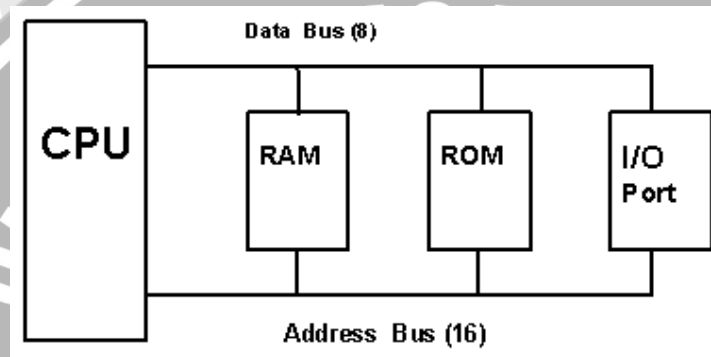
Gambar 2.9 Rangkaian komparator pembanding dengan referensi 0 volt.

Sumber: Penguat Operasional dan Rangkaian Terpadu Linear, Robert F. Coughlin, Frederick F. Driscoll, 1992.

2.8 Mikrokontroler AT89C51

Mikrokontroler adalah sebuah *single chip* yang di dalamnya sudah berisi :

- CPU (*Central Processing Unit*)
- RAM (*Random Acces Memory*)
- ROM (*Read Only Memory*)
- I/O (Unit Input / Output)
- *Timer/Counter , Serial COM Port.*



Gambar 2.10 Blok diagram mikrokontroler.

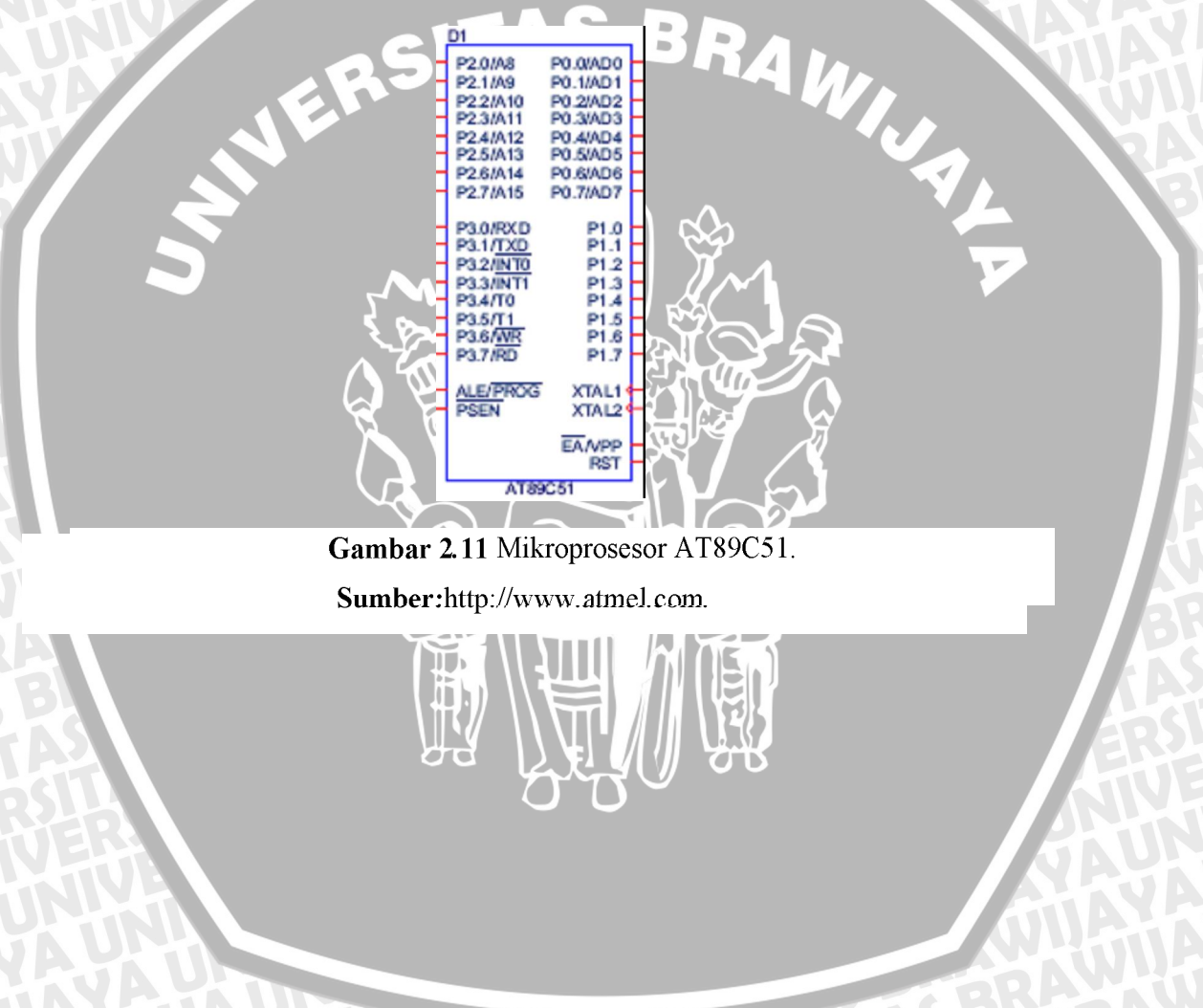
Sumber: <http://www.atmel.com>.

Mikrokontroler memiliki fungsi seperti mikroprosesor. Perbedaan yang mencolok antara mikrokontroler dengan mikroprosesor adalah kesederhanaan mikrokontroler yang dapat berdiri sendiri tanpa kehadiran komponen lainnya (seperti ADC, dan Memori program). Mikrokontroler merupakan rangkaian terintegrasi yang dikemas seperti komponen IC pada umumnya. Perbedaan utama dengan IC adalah pada kemampuan untuk melaksanakan perintah yang tersimpan dalam memori programnya. Di samping itu, mikrokontroler juga dilengkapi dengan blok ALU yang dapat melakukan operasi aritmetika. Tanpa adanya program yang disimpan dalam memori program, maka mikrokontroler tidak dapat melakukan apapun.

Mikrokontroler AT89C51 merupakan produk dari ATMEL yang memiliki ruang alamat memori data dan program yang terpisah. Mikrokontroler jenis ini adalah komplemen dengan mikrokontroler produk INTEL yang dikenal dengan keluarga

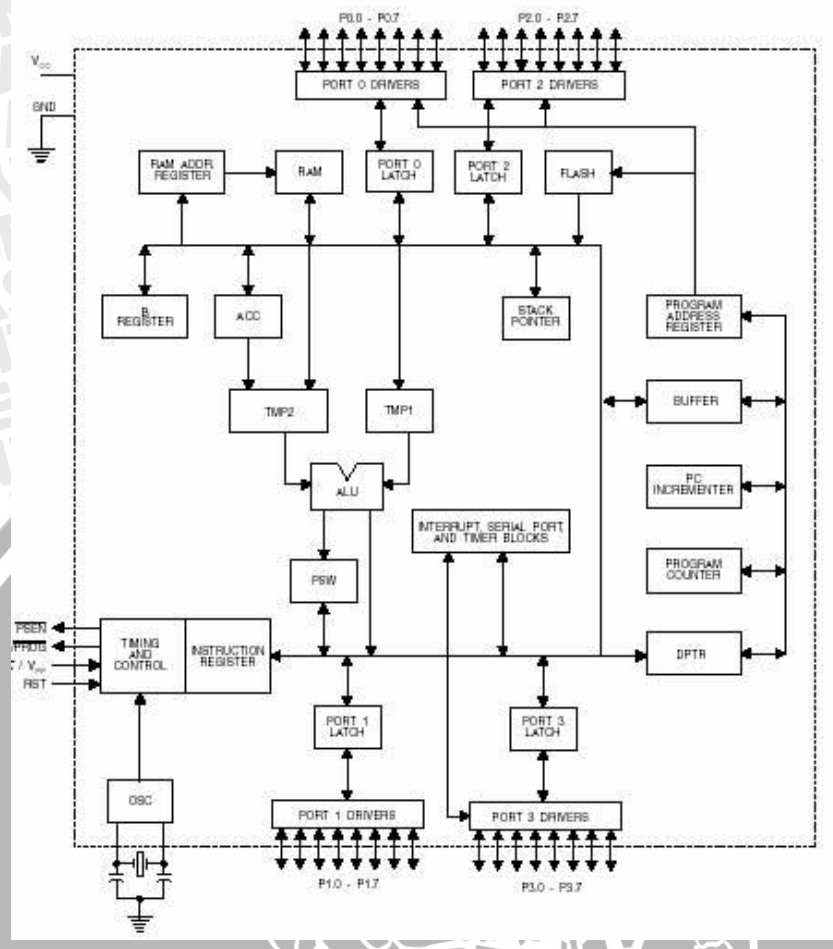
MCS 51. Berikut ini adalah perbandingan mikrokontroler ATMEL keluarga 51 yang mencakup tipe mikrokontroler, kapasitas memori program, memori data, jumlah pewaktu 16 bit dan teknologi pembuatannya.

Mikrokontroler AT89C51 memiliki prosesor 8 bit dan lebar bus data juga 8 bit sehingga memerlukan 8 pena (D0....D7). Akan tetapi karena jumlah pena mikrokontroler terbatas, pena data ini tidak dikeluarkan, hanya didalam *chip*. Pena untuk *bus* data dimultipleks dengan alamat A0....A7 pada *port* 0. Oleh karena itu, *port* 0 sering dituliskan sebagai AD0....AD7.



Gambar 2.11 Mikroprosesor AT89C51.

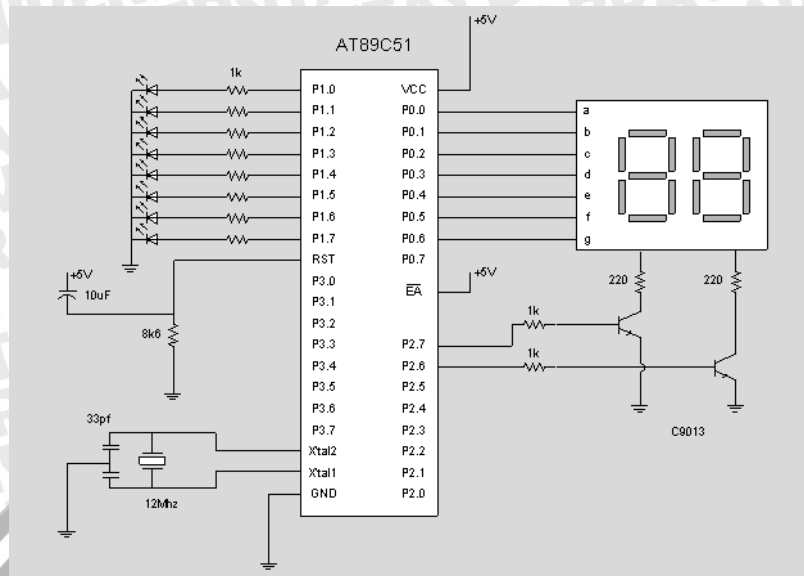
Sumber:<http://www.atmel.com>.



Gambar 2.12 Diagram blok AT89C51.

Sumber: <http://www.atmel.com>.

Seperti ditunjukkan Gambar 2.13 yang merupakan contoh aplikasi dari mikrokontroler AT89C51 untuk *running* LED dan 7 *segment*.



Gambar 2.13 Rangkaian at89c51 untuk *running* LED dan 7 *segment*.

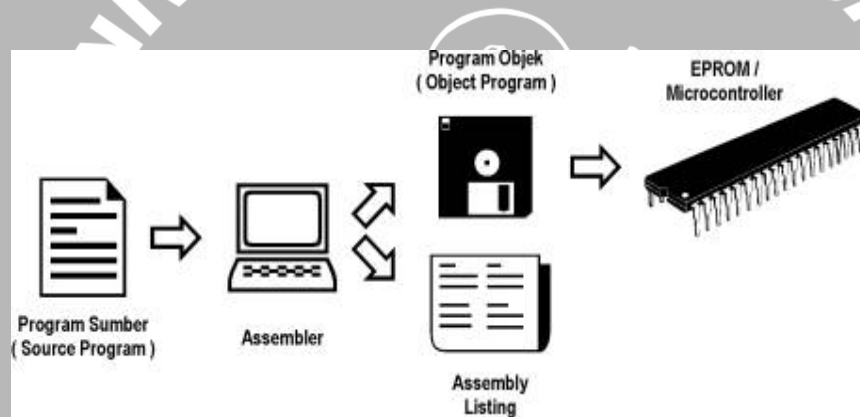
Sumber:<http://www.atmel.com>.

2.9 Bahasa Assembly

Bahasa *assembly* adalah bahasa komputer yang kedudukannya di antara bahasa mesin dan bahasa level tinggi misalnya bahasa C atau Pascal. Bahasa C atau Pascal dikatakan sebagai bahasa level tinggi karena memakai kata-kata dan pernyataan yang mudah dimengerti manusia, meskipun masih jauh berbeda dengan bahasa manusia sesungguhnya. Bahasa mesin adalah kumpulan kode biner yang merupakan instruksi yang bisa dijalankan oleh komputer. Sedangkan bahasa *assembly* memakai kode *mnemonic* untuk menggantikan kode biner, agar lebih mudah diingat sehingga lebih memudahkan penulisan program.

Program yang ditulis dengan bahasa *assembly* terdiri dari label; kode *mnemonic* dan lain sebagainya, pada umumnya dinamakan sebagai program sumber (*source code*) yang belum bisa diterima oleh prosesor untuk dijalankan sebagai program, tapi harus diterjemahkan dulu menjadi bahasa mesin dalam bentuk kode biner. Program sumber dibuat dengan program editor biasa, misalnya *note pad* pada windows atau *sidekick* pada DOS, selanjutnya program sumber diterjemahkan ke bahasa mesin dengan menggunakan program *assembler*. Hasil kerja program *assembler* adalah program objek dan juga *assembly listing*. Program objek berisikan kode-kode bahasa mesin, kode-kode bahasa mesin inilah yang diumpankan ke memori program prosesor.

Dalam dunia mikrokontroler biasanya program objek ini diisikan ke UV EPROM, dan khusus untuk mikrokontroler buatan Atmel, program ini diisikan ke dalam Flash PEROM yang ada di dalam *chip* AT89C51 atau AT89C2051. *Assembly listing* merupakan naskah yang berasal dari program sumber, dalam naskah tersebut pada bagian sebelah setiap baris dari program sumber diberi tambahan hasil terjemahan program *assembler*. Tambahan tersebut berupa nomor memori program berikut dengan kode yang akan diisikan pada memori program bersangkutan. Naskah ini sangat berguna untuk dokumentasi dan sarana untuk menelusuri program yang ditulis. Yang perlu diperhatikan adalah setiap prosesor mempunyai konstruksi yang berlainan, instruksi untuk mengendalikan masing-masing prosesor juga berlainan, dengan demikian bahasa *assembly* untuk masing-masing prosesor juga berlainan, yang sama hanyalah pola dasar cara penulisan program *assembly* saja.



Gambar 2.14 Bagan kerja proses *assembly*.

Sumber: <http://www.atmel.com>.

2.9.1 Konstruksi Program *Assembly*

Program sumber dalam bahasa *assembly* menganut prinsip 1 baris untuk satu perintah, setiap baris perintah tersebut bisa terdiri atas beberapa bagian (*field*), yakni bagian *label*, bagian *mnemonic*, bagian *operand* yang bisa lebih dari satu dan terakhir bagian komentar.

AT89C51 memiliki sekumpulan instruksi yang sangat lengkap. Jika telah mempelajari bahasa *assembly* mikroprosesor keluarga intel (misal 8086), ada sedikit perbedaan dengan bahasa *assembly* di mikrokontroler. Instruksi MOV untuk *byte* dan *bit* dikelompokkan sesuai dengan mode pengalamatan (*addressing modes*). Mode pengalamatan menjelaskan bagaimana *operand* dioperasikan. Berikut penjelasan dari

berbagai mode pengalamatan. Bentuk program *assembly* yang umum ialah sebagai berikut :

Tabel 2.1. Tabel Bentuk Program *Assembly*.

Label	Mnemonic	Operand 1	Operand 2	Komentar
Isi memori : 4000 7430	Opcode : MOV	A	#30 H	Kirim 30 H ke akumulator A

Sumber:<http://www.atmel.com>.

Untuk membedakan masing-masing bagian tersebut dibuat ketentuan sebagian berikut:

- Masing-masing bagian dipisahkan dengan spasi atau TAB, khusus untuk operand yang lebih dari satu masing-masing *operand* dipisahkan dengan koma.
- Bagian-bagian tersebut tidak harus semuanya ada dalam sebuah baris, jika ada satu bagian yang tidak ada maka spasi atau TAB sebagai pemisah bagian tetap harus ditulis.
- Bagian *label* ditulis mulai huruf pertama dari baris, jika baris bersangkutan tidak mengandung *label* maka *label* tersebut digantikan dengan spasi atau TAB, yakni sebagai tanda pemisah antara bagian *label* dan bagian *mnemonic*.

Berikut akan dijelaskan secara sekilas mengenai istilah-istilah yang terdapat di dalam bahasa *assembly*. Dan istilah-istilah tersebut adalah sebagai berikut :

- *Label*

Mewakili nomor memori-program dari instruksi pada baris bersangkutan, pada saat menulis instruksi *JUMP*, *Label* ini ditulis dalam bagian *operand* untuk menyatakan nomor memori program yang dituju. Dengan demikian *label* selalu mewakili nomor memori program dan harus ditulis dibagian awal baris instruksi .

- *Symbol*

Yaitu satu nama untuk mewakili satu nilai tertentu dan nilai yang diwakili bisa apa saja tidak harus nomor memori program. Cara penulisan *symbol* sama dengan cara penulisan *label*, harus dimulai di huruf pertama dari baris instruksi.

- *Mnemonic*

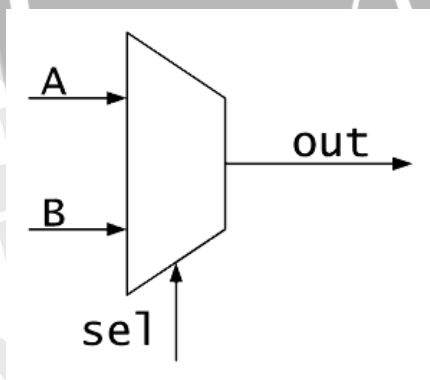
Artinya sesuatu yang memudahkan diingat, merupakan singkatan perintah, dikenal dua macam *mnemonic*, yakni *mnemonic* yang dipakai sebagai instruksi mengendalikan prosesor, misalnya ADD, MOV, DJNZ dan lain sebagainya. Ada pula *mnemonic* yang dipakai untuk mengatur kerja dari program *assembler* misalnya ORG, EQU atau DB, *mnemonic* untuk mengatur kerja dari program *assembler* ini dinamakan sebagai *assembler directive*.

- *Operand*

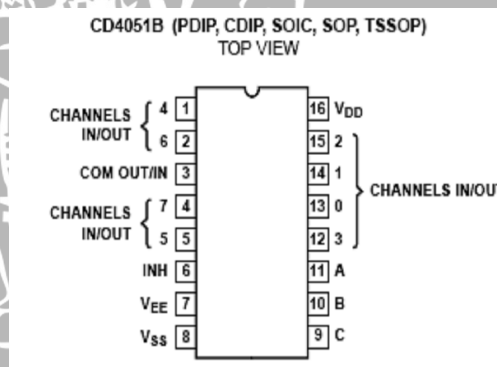
Adalah bagian yang letaknya di belakang bagian *mnemonic*, merupakan pelengkap bagi *mnemonic*. Kalau sebuah instruksi diibaratkan sebagai kalimat perintah, maka *mnemonic* merupakan subjek (kata kerja) dan *operand* merupakan objek (kata benda) dari kalimat perintah tersebut.

2.10 Multiplexer CD 4051

Multiplexer atau disingkat MUX adalah alat atau komponen elektronika yang bisa memilih *input* (masukan) yang akan diteruskan ke bagian *output* (keluaran). Pemilihan *input* mana yang dipilih akan ditentukan oleh sinyal yang ada di bagian kontrol (kendali) *select*.



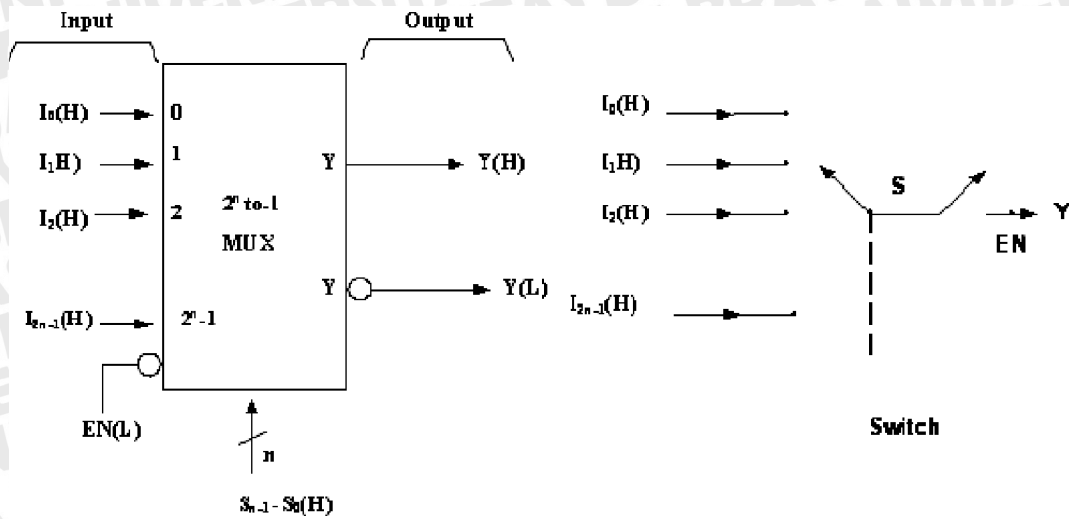
Gambar 2.15(a)



Gambar 2.15(b)

Gambar 2.15 (a) Skema multiplexer 2 input, (b) Pinout CD 4051.

Sumber: M. Morris Mano and Charles R. Kime, Logic and Computer Design Fundamentals, Prentice Hall, Inc., 2008.



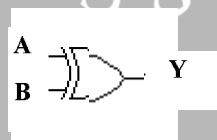
Gambar 2.16 Ilustrasi Multiplekser.

Sumber: M. Morris Mano and Charles R. Kime, Logic and Computer Design Fundamentals, Prentice Hall, Inc., 2008.

Komponen yang berfungsi kebalikan dari MUX ini disebut demultiplekser (DEMUX). Pada DEMUX, jumlah masukannya hanya satu, tetapi bagian keluarannya banyak. Signal pada bagian *input* ini akan disalurkan ke bagian *output* yang mana tergantung dari kendali pada bagian *select* nya.

2.11 Gerbang EXOR

Rangkaian ini merupakan pengembangan dari rangkaian OR dan disebut exclusive OR atau disingkat EXOR. Pada rangkaian EXOR nilai keluaran akan tinggi jika nilai kedua masukan berbeda. Sebaliknya nilai keluaran akan rendah jika kedua masukan bernilai sama. Simbol gerbang EXOR ini diperlihatkan pada Gambar 2.17



Gambar 2.17 Simbol gerbang EXOR.

Sumber: Mikroelektronika, Sistem Digital dan Rangkaian Analog Jilid I, Jacob Millman, 1986.

Tabel 2.2 Tabel Kebenaran Gerbang EXOR

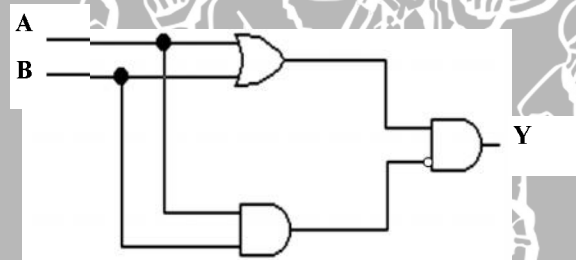
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Sumber: Mikroelektronika, Sistem Digital dan Rangkaian Analog
Jilid I, Jacob Millman, 1986.

Keterangan di atas sama dengan pernyataan berikut, jika $A=1$ atau $B=1$ tetapi tidak bersamaan maka $Y=1$, dimana dalam persamaan *boole* ditulis :

$$Y = (A + B)\overline{AB} \quad (2.28)$$

Fungsi dari persamaan tersebut terdapat pada Gambar 2.18 berikut:

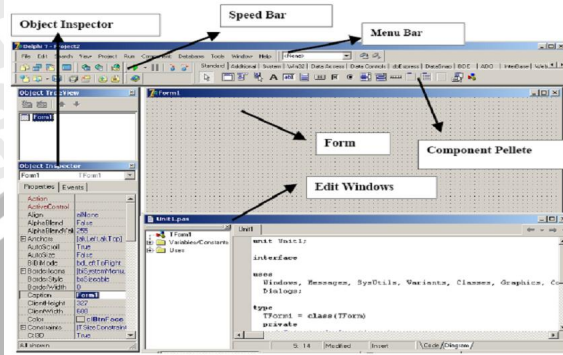
**Gambar 2.18** Rangkaian logika gerbang EXOR.

Sumber: Mikroelektronika, Sistem Digital dan Rangkaian Analog
Jilid I, Jacob Millman, 1986.

2.12 Borland Delphi 7

Delphi merupakan bahasa pemrograman berbasis *windows* yang menyediakan fasilitas pembuatan aplikasi *visual* seperti *visual basic*. *Delphi* memberikan kemudahan dalam menggunakan kode program, kompilasi yang cepat, penggunaan *file* unit ganda untuk pemrograman *modular*, pengembangan perangkat lunak, pola desain yang menarik serta diperkuat dengan bahasa pemrograman yang terstruktur dalam bahasa pemrograman *object pascal*. *Delphi 7* memiliki tampilan khusus yang

didukung suatu lingkup kerja komponen *delphi 7* untuk membangun suatu aplikasi dengan menggunakan *Visual Component Library (VCL)*. Sebagian besar pengembang *delphi* menuliskan dan mengkompilasi kode program dalam *Integrated Development Environment (IDE)*.



Gambar 2.19 IDE (*Integrated Development Environment*).

Sumber: Database Menggunakan Delphi, Inge Martina, 2002

2.12.1 Serial Port pada Borland Delphi 7

Serial port merupakan standar *interface* untuk komputer, selain *serial port* ada juga *paralel port* yang dapat digunakan untuk mengakses *hardware*. Perbedaan antara *serial port* dan *paralel port* tersebut yaitu cara *port* tersebut untuk mengirim dan menerima data. *Serial port* pada komputer sering digunakan untuk *mouse* (model lama), modem dan lain-lain sedangkan *paralel port* sering digunakan untuk *printer*, *scanner* dan lain sebagainya.

Perbedaan *serial port* dan *paralel port* :

Tabel 2.3 Tabel Perbedaan *Serial Port* dan *Paralel Port*

	Serial	Paralel
Metode pengiriman dan penerimaan data	Serial	Paralel
Kemampuan pengiriman data	± 20 M	± 2 M
Fleksibel	Membutuhkan kabel lebih sedikit	Membutuhkan kabel lebih banyak
Kecepatan data	Cepat	Lebih cepat

Sumber: Database Menggunakan Delphi, Inge Martina, 2002.

Metode Pengaksesan Perangkat Luar

Metode disini akan menggolongkan metode akses perangkat luar dengan 2 (dua) golongan yaitu aktif dan pasif, dimana penjelasannya adalah sebagai berikut :

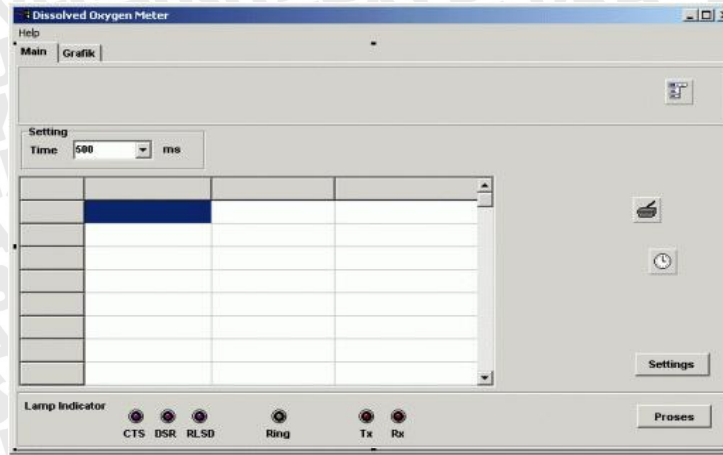
- Aktif, metode ini mengharuskan komputer untuk mengirim data ke perangkat luar terlebih dahulu sebelum perangkat luar mengirimkan datanya ke komputer. Data yang dikirimkan komputer dapat berupa kata atau huruf. Kata atau huruf yang akan dikirim ke perangkat luar ini tidak memiliki standar baku karena tiap perangkat luar memiliki susunan masing-masing, untuk dapat mengetahui kata atau huruf yang akan dikirimkan dapat dilihat dalam buku manual yang telah disertakan oleh tiap perangkat. Contohnya PH meter *digital*, beberapa model *barcode*, instrumen pengendalian dan lain-lain.
- Pasif, metode ini perangkat luarlah yang aktif mengirimkan data sedangkan komputer hanya menerima data. Contohnya *mouse*.

Konfigurasi

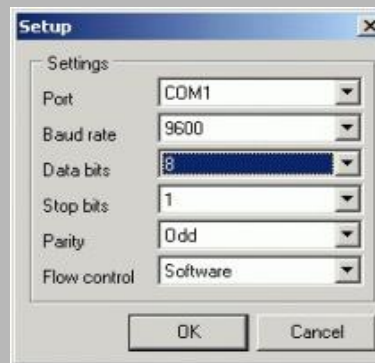
Untuk dapat mengakses perangkat lunak kita harus mengetahui terlebih dahulu konfigurasi dari perangkat luar tersebut antara lain :

- *Port* yang akan digunakan, komputer yang sekarang terdapat 2 *port serial*
- *Baud rate*, kecepatan data yang akan digunakan
- Data bit, bit data
- Stop bit, bit untuk memisahkan data
- *Parity*, bit untuk validasi data yang telah dikirim
- *Flow control*

Konfigurasi tiap perangkat berbeda-beda sehingga kita harus memiliki buku manual untuk dapat mengaksesnya.



Gambar 2.20 (a)



Gambar 2.20 (b)

Gambar 2.20 (a) Contoh desain tampilan pada delphi 7, **(b)** Form tampilan.

Sumber: Database Menggunakan Delphi, Inge Martina, 2002.

2.13 RS-232

Standar sinyal komunikasi serial yang banyak digunakan ialah standar RS-232. Standar ini hanya menyangkut komunikasi data antara komputer (*Data Terminal Equipment* – DTE) dengan alat – alat pelengkap komputer (*Data Circuit-Terminating Equipment* – DCE). Standar RS-232 inilah yang biasa digunakan pada serial port.

Standar sinyal serial RS-232 memiliki ketentuan level tegangan sebagai berikut:

- Logika '1' disebut '*mark*' terletak antara -3 volt hingga -25 volt.
- Logika '0' disebut '*space*' terletak antara +3 volt hingga +25 volt.
- *Invalid level*, yaitu daerah tegangan yang tidak memiliki level logika pasti.

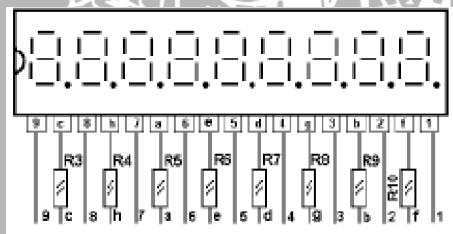
Daerah tegangan antara -3 volt hingga +3 volt adalah *invalid level*, yaitu daerah tegangan yang tidak memiliki level logika pasti sehingga harus dihindari. Demikian juga level tegangan lebih negatif dari -25 volt atau lebih positif dari +25 volt juga harus dihindari karena dapat merusak *line driver* pada saluran RS232.

2.14 Pengertian 7 Segment dan Serial Port DB 9

2.14.1 Pengertian 7 Segment

7 *segment* diatur sebagai segi empat panjang dari dua *segment* yang *vertikal* pada masing-masing sisi, dengan satu *segment* yang *horizontal* di bagian atas dan bawah. Dimana, *segment* yang ketujuh membagi dua bagian segi empat panjang secara *horizontal*. Kebanyakan 7 *segment* menggunakan tampilan suatu *array* dari diode pemancar cahaya (LED), meskipun demikian pada jenis yang lain ada teknologi alternatif yang menggunakan seperti lucutan gas katode dingin, ruang hampa kaw at pijar yang bercahaya berpijar, tampilan hablur cair (LCD), dan lain lain.

Suatu alternatif bagi 7 segment pada tahun 1950 sampai 1970 an adalah *nixie* tabung ruang hampa yang seperti tabung. Mulai 1970, RCA yang dijual berupa suatu alat tampilan yang dikenal sebagai Numitron yang menggunakan kawat pijar bercahaya.



Gambar 2.21 7 segment indikator DT-I/O Low Cost MMI.

Sumber:<http://www.mikron123.com>.

DT-I/O Low Cost MMI merupakan sebuah modul *Man Machine Interface* yang digunakan sebagai sarana antarmuka antara manusia dan sistem elektronik. Modul ini tersusun atas 8 digit 7 *segment* dan 4 buah tombol. Modul ini dapat difungsikan sebagai *input* dan *display output* dalam aplikasi pengendali kecepatan motor, robotik, dan aplikasi lainnya, serta *kompatibel* penuh dengan DT-51™ *Low Cost Series* dan DT AVR *Low Cost Series*.

Dimensi:

10,15 cm (P) x 4,45 cm (L) x 2,1 cm (T)

Fitur:

- 8 digit 7 *Segment* dengan tinggi 0,56 inchi jenis *super bright red*,
- 4 buah tombol sebagai *input* dengan fungsi tergantung aplikasi,
- Membutuhkan tegangan + 5V DC sebagai *catu daya*,
- Desain PCB standar industri dengan bahan 2 *layer* FR4 dan *Plated Through Hole* (PTH). Serta menggunakan komponen SMD (*Surface Mount Device*) membuat PCB DT-I/O *Low Cost* MMI ini lebih baik.

2.14.2 Serial Port DB 9

Pada PC / laptop standar, biasanya terdapat sebuah *port* untuk komunikasi *serial*. Pada prinsipnya, komunikasi *serial* ialah komunikasi dimana pengiriman data dilakukan per bit, sehingga lebih lambat dibandingkan komunikasi *pararel* seperti pada *port* printer yang mampu mengirim 8 bit sekaligus dalam sekali detak. Beberapa contoh penerapan komunikasi *serial* ialah *mouse*, *scanner* dan sistem akuisisi data yang terhubung ke *port serial* COM1/COM2.

Komunikasi serial membutuhkan *port* sebagai saluran data. Berikut tampilan *port serial* DB9 yang umum digunakan sebagai *port serial* :



Gambar 2.22 (a)



Gambar 2.22 (b)

Gambar 2.22 (a) Port DB 9 jantan, **(b)** Port DB 9 Betina.

Sumber:<http://www.elexmedia.co.id>.

Untuk menghubungkan antara 2 buah PC, biasanya digunakan format *null mode*, dimana pin TxD dihubungkan dengan RxD pasangan, pin sinyal *ground* (5) dihubungkan dengan SG di pasangan, dan masing masing pin DTR, DSR dan CD dihubungkan singkat, dan pin RTS dan CTS dihubungkan singkat di setiap *devais*.



BAB III METODOLOGI

Penyusunan skripsi ini didasarkan pada masalah yang aplikatif, yaitu perancangan alat agar dapat menampilkan unjuk kerja sesuai dengan yang direncanakan dengan mengacu pada rumusan masalah. Data dan spesifikasi komponen yang digunakan dalam perancangan diambil dari buku data komponen elektronika dan beberapa situs internet. Pemilihan komponen berdasarkan perancangan disesuaikan dengan komponen yang ada di pasaran. Langkah-langkah yang perlu dilakukan untuk merealisasikan alat adalah sebagai:

3.1 Studi Literatur

Melakukan kajian pustaka untuk memahami tentang teori FM (*Frekuensi Modulasi*), mikrokontroler AT89C51, *borland delphi7*, dan juga perangkat pendukung lainnya yang dibutuhkan dalam proses perancangan alat.

3.2 Analisis Kebutuhan

Menentukan kebutuhan dari alat yang direncanakan dan memiliki spesifikasi dasar sebagai berikut:

- a. Alat dapat dijalankan dengan menggunakan bahasa pemrograman *assembly*.
- b. Untuk menjalankan sistem dibutuhkan suatu pemancar FM dengan *range* frekuensi antara 88 MHz-108 Mhz, yang berfungsi mengirimkan sinyal informasi kepada penerima yang kemudian akan diukur kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa.
- c. Alat yang dirancang memiliki kemampuan untuk menampilkan sinyal suara (*audio*) yang diterima oleh penerima, dimana sinyal suara (*audio*) yang diterima akan ditampilkan dalam bentuk angka oleh 7 *segment*.
- d. Kualitas sinyal suara (*audio*) merupakan hasil perhitungan dari kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa.

3.3 Perancangan Sistem

3.3.1 Perancangan Perangkat Keras

Dalam perancangan hardware, yang harus dilakukan adalah :

1. Melihat konfigurasi *interface* antara pemancar FM mini 10 mW yang digunakan dengan penerima yang selanjutnya akan terhubung dengan mikrokontroler sistem, sebab dari konfigurasi itulah yang akan menjadi acuan bentuk rangkaian mikrokontroler untuk mengirim perintah kepada sistem. Dimana pada *hardware* dibuat sebuah perancangan sistem deteksi sinyal informasi yang dilengkapi dengan pembacaan secara langsung oleh komputer sistem.
2. Konfigurasi antara mikrokontroler sistem dan komputer sistem, dimana data yang ada pada mikrokontroler sistem akan dikirim ke komputer melalui *serial port* (COM Port 1) menggunakan media RS-232 untuk perhitungan kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa. Misalnya data pada mikrokontroler sistem mendeteksi kuat sinyal sebesar 100 mV, maka data yang diterima oleh komputer sistem harus sama yaitu 100 mV, jika tidak sama maka perlu dilakukan koreksi terhadap perancangan yang sudah dilakukan.

3.3.2 Pembuatan Perangkat Keras

Berdasarkan hasil perancangan, maka selanjutnya adalah proses pembuatannya. Dalam pembuatan *hardware* ini ada beberapa alat yang dibuat, yaitu antara lain, rangkaian mikrokontroler beserta rangkaian *interface* pemancar FM dan penerimanya, dan sebuah *hardware* yang merupakan bagian dari komputer sistem. Adapun keterangannya adalah sebagai berikut:

3.3.2.a Pembuatan Rangkaian Mikrokontroler

Untuk membuat rangkaian mikrokontroler mengacu pada konsep perancangan, langkah pertama adalah menggambar rangkaian tipikal mikrokontroler sesuai dengan *datasheet* AT89C51 untuk komunikasi serial dan kemudian dihubungkan dengan semua gambar rancangan dari *hardware* pendukung lainnya, selanjutnya pemilihan komponen, diantaranya digunakan IC mikrokontroler AT89C51. Langkah terakhir adalah membentuk dan merangkai komponen yang ada dan menghubungkan rangkaian mikrokontroler ini dengan rangkaian pendukung lainnya.

3.3.2.b Pembuatan Rangkaian Interface Pemancar FM Dan Mikrokontroler

Pembuatan rangkaian interface pemancar FM dan mikrokontroler mengacu gambar rangkaian pada *datasheet* mikrokontroler AT89C51, dan dilengkapi dengan 7 *segment* untuk menampilkan hasil yang didapat, untuk kemudian dilanjutkan dengan merangkai dan membentuknya.

3.3.2.c Perancangan Hubungan Mikrokontroler Sistem Dengan Komputer Sistem

Digunakan suatu media penghubung berupa *serial port* antara mikrokontroler sistem dengan komputer sistem, agar data yang terbaca pada mikrokontroler sistem dapat terbaca pada komputer sistem .

3.3 Perancangan dan Pembuatan Perangkat Lunak

3.3.1 Perancangan Perangkat lunak

Dalam perancangan *software* ini, yang perlu diperhatikan adalah *software* pada bagian mikrokontroler sistem dan komputer sistemnya, dimana sinyal informasi yang sudah melalui proses demodulasi dan diterima oleh penerima akan dikirim langsung ke sistem mikrokontroler sistem. Pada mikrokontroler akan digunakan *software* dalam bentuk bahasa pemrograman *assembly*, Sedangkan pada komputer sistem akan dirancang sebuah sistem data base dengan menggunakan *software borland delphi 7*.

3.3.2 Pembuatan Perangkat Lunak

Berdasarkan hasil perancangan *software*, maka program bahasa *assembly* pada mikrokontroler sistem diatur untuk memproses data program yang diterima, dalam kasus ini yang diterima berupa sinyal suara (*audio*) yang berasal dari penerima, untuk kemudian ditampilkan hasilnya, sebagai contoh dengan kondisi jarak pancar ± 5 meter dari pemancar dan dengan cuaca yang baik maka proses pengiriman dilakukan, setelah melalui berbagai proses maka sinyal informasi akan sampai kepada penerima, misalnya besarnya kuat sinyal yang diterima sebesar 75 mV. Dari data yang diperoleh tersebut komputer sistem yang menggunakan *software borland delphi 7*, akan menampilkan besarnya kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa secara otomatis dan sesuai dengan besarnya nilai yang terbaca pada mikrokontroler sistem. Semakin besar kuat sinyal yang diterima dan semakin kecil beda fasa -nya,

maka informasi yang diterima juga semakin baik. Begitu juga sebaliknya, jika kuat sinyal yang diterima semakin kecil dan beda fasanya semakin besar, maka informasi yang diterima tidak baik, dan proses selanjutnya adalah mengatur ulang pada proses awalnya, misalnya dengan memperpendek jarak pancar dengan demikian diharapkan sinyal informasi yang diterima oleh penerima akan lebih baik.

3.4 Pengujian dan Perhitungan pada Perangkat Keras

Proses pengujian dan perhitungan ini dilakukan pada rangkaian beda fasa, hal ini bertujuan untuk mengetahui apakah rangkaian yang akan digunakan sudah *valid* atau tidak, karena pengukuran beda fasa merupakan hal yang penting dalam pengukuran yang akan dilakukan .

3.5 Pengujian dan perhitungan pada Mikrokontroler

Untuk pengujian pada mikrokontroler dilakukan dengan cara menghubungkan mikrokontroler dengan komputer, hal ini bertujuan untuk mengetahui bahwa ada *koneksi* antara sistem mikrokontroler dengan sistem komputer, *software* yang digunakan dalam pengujian ini adalah *hyper terminal*. Apabila sudah terjadi koneksi, maka sistem secara keseluruhan dapat dijalankan.

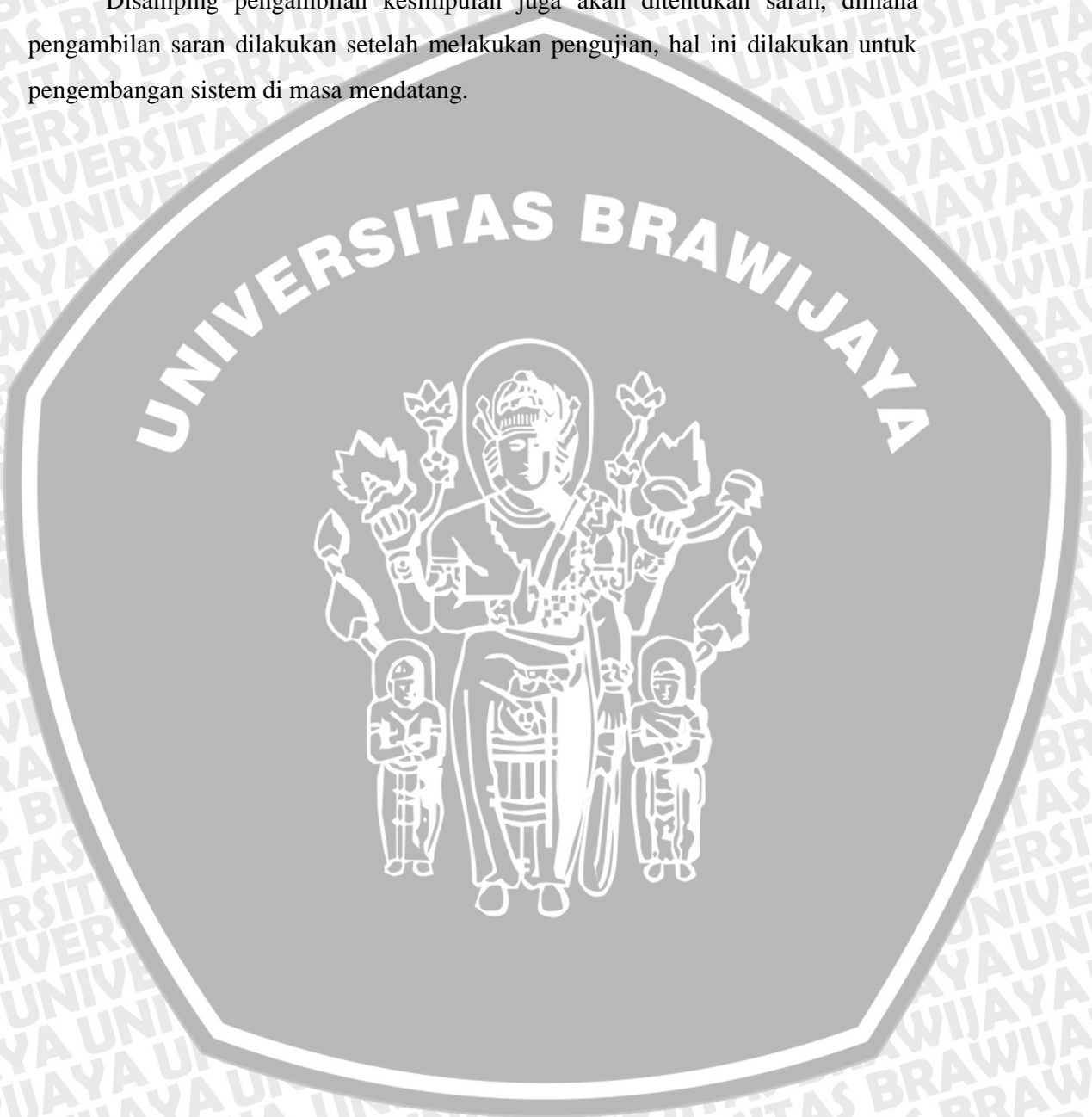
3.6 Pengujian Validitas Sinyal Informasi

Pengujian *validitas* sinyal informasi ini dilakukan untuk mengetahui apakah sinyal suara (*audio*) yang dipancarkan sudah layak atau tidak untuk diterima oleh penerima, pengujian ini berdasarkan besarnya kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa yang dihasilkan, *parameter* yang digunakan dalam pengujian adalah dengan melakukan pengukuran kuat sinyal yang dipancarkan / *strength meter*, dan beda fasa berdasarkan jarak pemancar dan penerima, serta pengukuran kuat sinyal yang dipancarkan / *strength meter*, dan beda fasa berdasarkan tinggi antena pemancar. Disamping itu pengukuran juga dilakukan terhadap beberapa pemancar radio FM yang berada di Malang dan sekitarnya, dalam hal ini diambil data dari 12 stas iun radio. Dalam pengukuran pada pemancar FM lain di Malang dan sekitarnya yang diukur hanya sebatas kuat sinyal yang dipancarkan / *strength mete*.

3.7 Analisis Data dan Pengambilan Kesimpulan

Pengambilan kesimpulan akan didasarkan pada kesesuaian antara teori dan praktek, dimana kesimpulan secara teori akan disesuaikan dengan literatur yang telah ditulis.

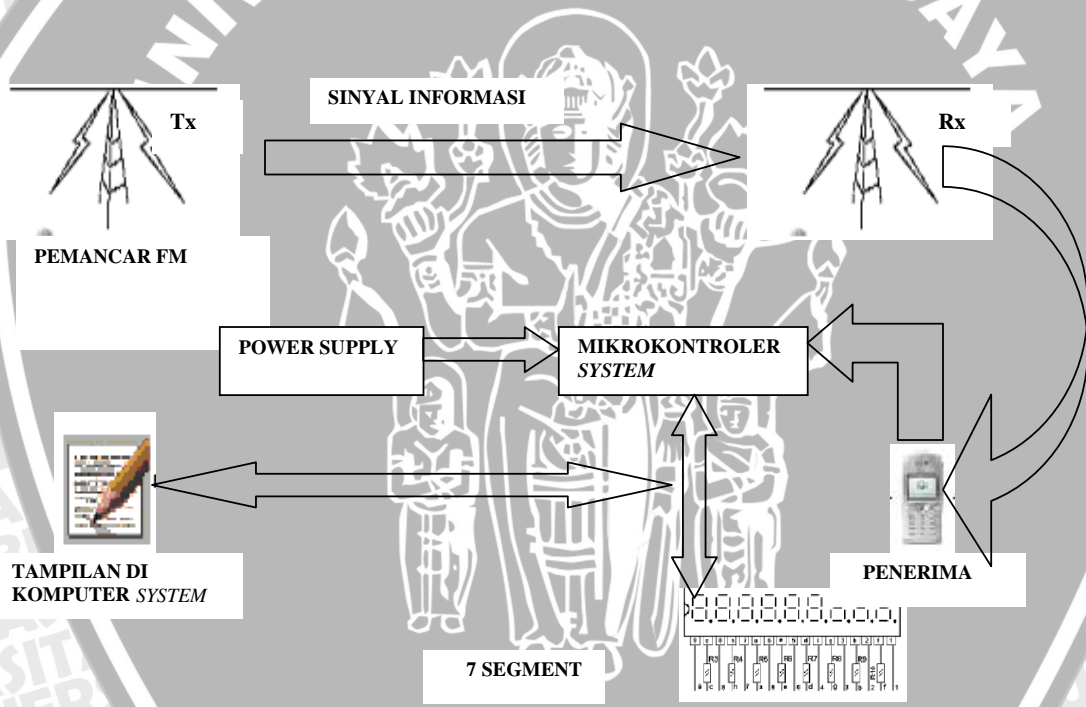
Disamping pengambilan kesimpulan juga akan ditentukan saran, dimana pengambilan saran dilakukan setelah melakukan pengujian, hal ini dilakukan untuk pengembangan sistem di masa mendatang.



BAB IV
PERANCANGAN DAN PEMBUATAN ALAT

Dalam perancangan sistem deteksi sinyal suara (*audio*) pada FM (*Frekuensi Modulasi*) dengan menggunakan mikrokontroler AT89C51 ini, terdapat beberapa komponen yang memiliki fungsi penting. Gambar 4.1 merupakan gambaran bagian-bagian dan proses kerja dari sistem deteksi ini.

Dibawah ini merupakan desain perancangan alat pendeteksi kualitas sinyal suara (*audio*) pada modulasi FM (*Frekuensi Modulasi*) dengan menggunakan Mikrokontroler AT89C51. Pada desain ini digambarkan blok mengenai sistem pengiriman sinyal informasi sampai penerimaan dan pembacaan yang dilakukan oleh sistem mikrokontroler.



Gambar 4.1 Desain perancangan alat pendeteksi sinyal suara (*audio*) dengan menggunakan mikrokontroler AT89C51.

Sumber: Perancangan.

Gambar diatas menjelaskan tentang alur kerja sistem. Dan keterangan dari masing-masing bagian, adalah sebagai berikut :

1. Pemancar FM.
Berfungsi sebagai pemancar untuk memancarkan sinyal informasi dengan menggunakan modulasi frekuensi yang akan dikirimkan kepada penerima, frekuensi yang digunakan adalah sesuai dengan keadaan.
2. Penerima.
Merupakan penerima sinyal FM dari pemancar, yang kemudian sinyal tersebut di demodulasikan sehingga menghasilkan bentuk sinyal informasi yang diterima.
3. *Power supply*.
Merupakan perangkat yang memberikan sumber tegangan catu daya terhadap mikrokontroler dan beberapa komponen yang ada, dimana besar tegangan yang diberikan sebesar 5 volt.
4. Mikrokontroler sistem.
Merupakan elemen penting dalam sistem dimana terdapat Mikrokontroler AT89C51 di dalamnya yang berfungsi sebagai pusat dari seluruh proses pada alat, yang bertujuan untuk mendeteksi kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa, yang merupakan hasil perhitungan pada mikrokontroler sistem.
5. Tampilan di komputer sistem.
Menampilkan hasil perhitungan secara lebih rinci dari perhitungan sebelumnya yang sudah dilakukan oleh mikrokontroler sistem.

Sistem deteksi ini dirancang untuk memiliki proses kerja sebagai berikut :

1. Menentukan besarnya frekuensi yang akan dipancarkan kepada penerima.
2. Setelah frekuensi ditentukan, dan melalui proses modulasi maka proses selanjutnya adalah mengirimkan sinyal informasi dengan menggunakan antena pemancar.
3. Dengan menggunakan antena penerima sistem penerima menerima sinyal informasi yang dikirimkan, setelah melalui proses demodulasi maka sinyal informasi masuk ke pesawat penerima.
4. Dari pesawat penerima, sinyal informasi akan dilanjutkan ke mikrokontroler sistem. Pada mikrokontroler sistem akan dilakukan proses perhitungan kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa.
5. Selanjutnya komputer sistem akan menampilkan proses perhitungan kuat sinyal

yang dipancarkan / *strength meter*, serta beda fasa dengan lebih rinci dan jelas.

4.1 Spesifikasi Umum Alat

Dalam perancangan ini terdapat spesifikasi umum alat yang terdiri dari:

1. Sumber tegangan yang digunakan adalah sumber tegangan AC 220 V.
2. Besarnya frekuensi yang akan dikirimkan ditentukan pada saat proses pengiriman sinyal informasi.
3. Jarak pengukuran antara pemancar dan penerima pada saat proses perhitungan ditentukan berdasarkan kebutuhan pengujian sistem, dimana jarak maksimum adalah 5 M.
4. Dimensi alat berukuran :

Panjang : ± 30 cm

Lebar : ± 15 cm

Tinggi : ± 20 cm

Berat : ± 750 gram

4.2 Penentuan Spesifikasi Khusus Alat

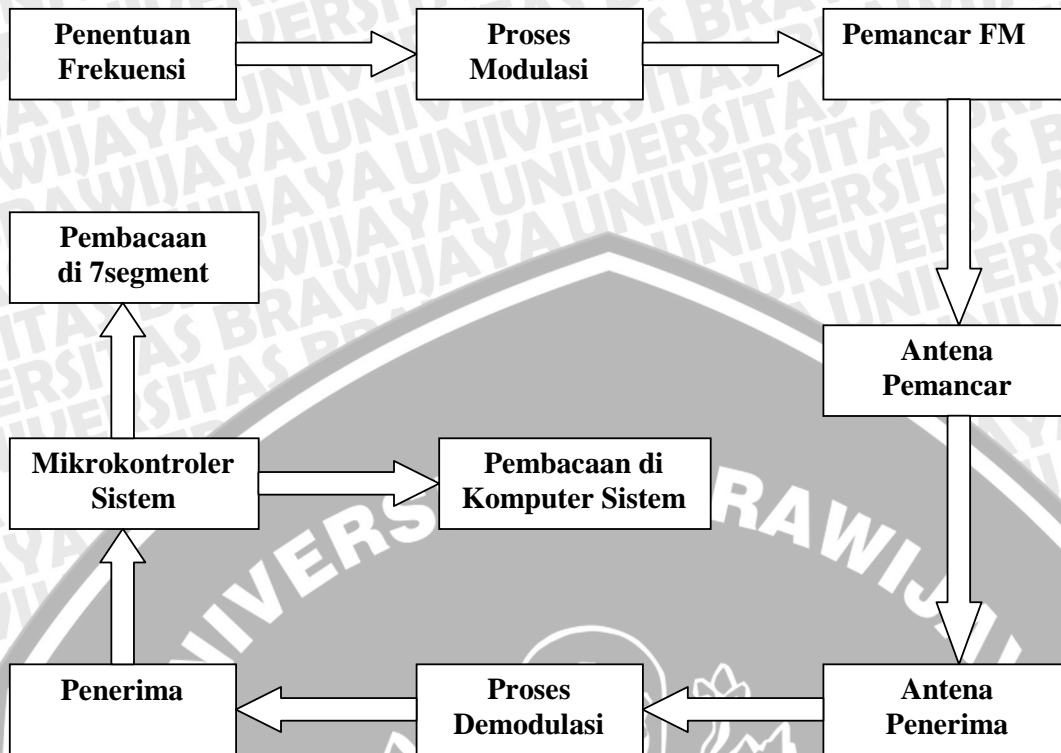
Dalam perancangan alat sistem deteksi ini, spesifikasi khusus yang akan direalisasikan adalah sebagai berikut:

1. Sistem mikrokontroler AT89C51 berfungsi sebagai kontrol sistem.
2. Menggunakan FM *tuner* sebagai penerima.
3. Untuk menghubungkan mikrokontroler sistem dengan komputer sistem dihubungkan melalui *com port* I komputer, dengan media kabel RS 232.

4.3 Perancangan Blok Diagram Alat

Agar perancangan dan pembuatan alat dapat dilakukan dengan sistematis dan terstruktur, maka perlu dibuat blok diagram yang menjelaskan sistem yang dirancang.

Blok diagram sistem ditunjukkan dalam Gambar 4.2 berikut:



Gambar 4.2 Blok Diagram Sistem Deteksi Sinyal.
Sumber: Perancangan.

4.4 Prinsip Kerja Alat

Dibawah ini akan dibahas beberapa prinsip kerja dari alat yang dirancang, adapun prinsip kerjanya adalah sebagai berikut:

1. *Analog multiplekser* (AMUX 4051) akan menerima 2 jenis *input* (masukan), yaitu:
 - A1 : kuat sinyal yang dipancarkan / *strength meter*,
 - A2 : besar beda fasa.
2. Ketika 2 *input* sudah masuk ke dalam analog multiplekser, maka saklar *selector* akan mengirimkan secara bergantian 2 *input* tersebut ke mikrokontroler.
3. Sebelum masuk ke mikrokontroler 2 *input* tersebut akan dikonversi terlebih dahulu oleh ADC (*Analog Digital Converter*), untuk merubah dari sinyal analog menjadi sinyal digital.

4. Setelah melalui proses di ADC maka akan terjadi perubahan, dimana akan didapatkan keluaran baru berupa 2 sinyal keluaran digital, dan perinciannya adalah sebagai berikut:
A1 : kuat sinyal yang dipancarkan / *strength meter*, menjadi digital A1.
A2 : besar beda fasa, menjadi digital A2.
5. Setelah berbentuk sinyal digital maka 2 *input* akan masuk ke mikrokontroler, dimana di dalam mikrokontroler masing-masing dari *input* (masukan) akan berada pada *memory* yang terpisah antara satu dan yang lainnya.
6. Kemudian setelah itu, data digital yang telah diperoleh akan di konversi ke EX2BCD.
7. Data dari 2 *input* (masukan) akan ditampilkan dalam bentuk angka oleh 7 *segment*, adapun yang muncul pada 7 *segment* berupa tampilan dari kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa.
8. Selain itu data yang masih tersimpan di *memory* mikrokontroler akan dikirim ke komputer dengan dihubungkan melalui *com port* 1 komputer, dengan media kabel RS 232 , yang sebelumnya diproses dengan *serial buffer*.
9. Data keluaran akan ditampilkan di komputer dengan menggunakan program *borland delphi 7*.

4.5 Perancangan dan Pembuatan Perangkat Keras

4.5.1. Perancangan Perangkat Keras

Dalam perancangan perangkat keras, yang harus dilakukan adalah sebagai berikut:

1. Melihat konfigurasi *interface* antara pemancar FM mini 10 mW yang digunakan dengan penerima yang selanjutnya akan terhubung dengan mikrokontroler sistem, sebab dari konfigurasi itulah yang akan menjadi acuan bentuk rangkaian mikrokontroler untuk mengirim perintah kepada sistem. Dimana pada *hardware* dibuat sebuah perancangan sistem deteksi sinyal suara (*audio*) yang dilengkapi dengan pembacaan sinyal yang dipancarkan / *strength meter*, serta beda fasa secara langsung oleh komputer sistem.
2. Konfigurasi antara mikrokontroler sistem dan komputer sistem, dimana data yang ada pada mikrokontroler sistem akan dikirim ke komputer melalui *serial port* (COM Port 1) menggunakan media RS-232 untuk perhitungan sinyal yang dipancarkan / *strength meter*, serta beda fasa. Misalnya pada mikrokontroler

sistem mendeteksi kuat sinyal sebesar 2,5 mV, maka data yang diterima oleh komputer sistem harus sama yaitu 2,5 mV, jika tidak sama maka perlu dilakukan koreksi terhadap perancangan yang sudah dilakukan.

4.5.2 Pembuatan Perangkat Keras

Berdasarkan hasil perancangan, maka selanjutnya adalah proses pembuatannya. Dalam pembuatan *hardware* ini ada beberapa alat yang dibuat, yaitu antara lain, rangkaian mikrokontroler beserta rangkaian *interface* pemancar FM dan penerimanya, dan sebuah *hardware* yang merupakan bagian dari komputer sistem. Adapun keterangannya adalah sebagai berikut:

4.5.2.a Pembuatan Perangkat Keras

Untuk membuat rangkaian mikrokontroler mengacu pada konsep perancangan, langkah pertama adalah menggambar rangkaian tipikal mikrokontroler sesuai dengan *datasheet* AT89C51 untuk komunikasi *serial* dan kemudian dihubungkan dengan semua gambar rancangan dari *hardware* pendukung lainnya, selanjutnya pemilihan komponen, diantaranya digunakan IC mikrokontroler AT89C51. Langkah terakhir adalah membentuk dan merangkai komponen yang ada dan menghubungkan rangkaian mikrokontroler ini dengan rangkaian pendukung lainnya.

4.5.2.b Reset Mikrokontroler

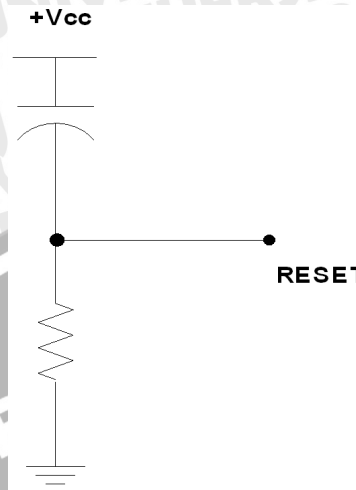
Rangkaian *reset* diperlukan untuk me-*reset* mikrokontroler secara otomatis setiap kali catu daya dinyalakan, hal ini akan me-*reset* program *counter* sehingga perintah program yang akan dieksekusi dimulai pada alamat 0. Ketika catu daya diaktifkan, rangkaian *reset* menahan logika tinggi pin RST dengan jangka waktu yang ditentukan oleh lamanya pengisian muatan C. Jika pin RST diberi logika *high* selama dua siklus mesin, maka mikrokontroler akan di-*reset*.

$$1 \text{ siklus mesin} = \left(\frac{12}{f_{osc}} \right) \quad (4.1)$$

maka untuk kristal dengan frekuensi 11,0592 MHz membutuhkan waktu 1,085 μs , sehingga untuk me-*reset* mikrokontroler dibutuhkan waktu minimal 2,17 μs .

Tegangan RST yang diijinkan sebesar $0,7 V_{CC} - V_{CC} + 0,5$. Jika tegangan RST minimal

3,5 V maka tegangan pada kapasitor maksimal 1,5 V untuk dapat me -reset mikrokontroler. Rangkaian *reset* dibentuk dengan rangkaian RC sebagai berikut:



Gambar 4.3 Rangkaian *reset* mikrokontroler.

Sumber: Perancangan.

Dari rangkaian setara dengan Gambar 4.3 diperoleh:

$$V_o = \frac{R}{R + \frac{1}{sC}} \cdot V_i \quad (4.2)$$

$$V_o = \frac{sCR}{sCR + 1} \cdot V_i \quad (4.3)$$

dengan tegangan V_i adalah V_{cc} yaitu 5 volt, dalam fungsi *laplace* adalah $\frac{5}{s}$ sehingga:

$$V_o = \frac{5}{s} \cdot \frac{sCR}{sCR + 1}$$

$$V_o = 5 \cdot \frac{CR}{sCR + 1}$$

$$V_o = 5 \left[\frac{1}{s + 1/RC} \right]$$

$$V_o = 5 \cdot e^{-\frac{t}{RC}}$$

$$\frac{5}{V_o} = e^{\frac{t}{RC}}$$

$$\ln \frac{5}{V_o} = \frac{t}{RC}$$

$$t = RC \left[\ln \frac{5}{V_o} \right]$$

dengan V_o adalah tegangan logika nominal yang diijinkan pin RST dimana:

$$V_o = 0,7 \times V_{cc} = 0,7 \times 5 \text{ volt} = 3,5 \text{ volt}$$

Sehingga:

$$t = RC \left[\ln \frac{5}{3,5} \right] \tag{4.4}$$

$$t = 0,357 \cdot R \cdot C$$

dengan menggunakan $R=10 \text{ k}\Omega$ dan t minimumnya adalah $2,17 \mu\text{s}$, maka:

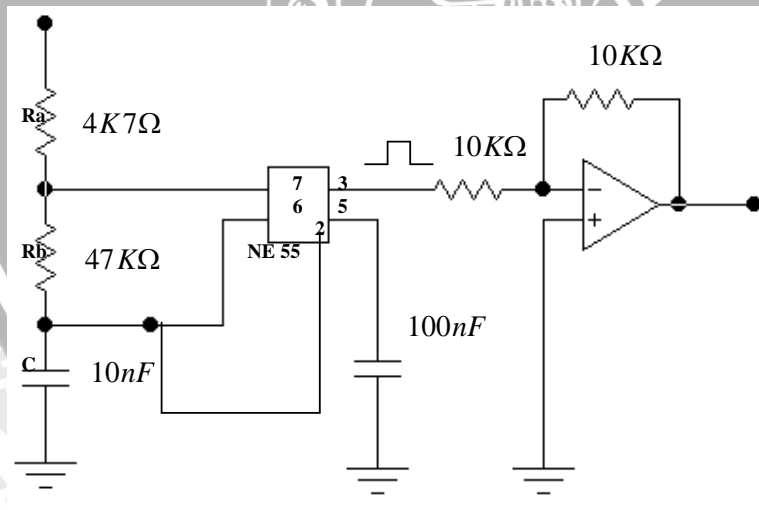
$$2,17 \cdot 10^{-6} = 0,375 \times 10K \times C$$

dan didapatkan $C = 0,6 \text{ nF}$

Nilai R dan C tersebut telah memenuhi syarat minimal untuk dapat me *-reset* mikrokontroler AT89C51. Dalam perancangan menggunakan $10\mu\text{F}$ dan $8,2\text{k}\Omega$.

4.5.3 Perancangan Generator Uji

Generator uji berfungsi untuk membangkitkan sinyal yang digunakan dalam pengujian beda fasa, di mana spesifikasi dari sinyal *output* dari *generator* uji adalah gelombang balok dengan frekuensi = $1,5 \text{ KHz}$, $D = 50\%$, dan A *peak to peak* = 5 volt .



Gambar 4.4 Skema rangkaian *generator* uji.

Sumber: Perancangan.

Perhitungan spesifikasi dari sinyal *output* pada *generator* uji adalah :

- Perhitungan *duty cycle* :

$$D = \frac{Ra + Rb}{Ra + 2Rb} \times 100\%$$

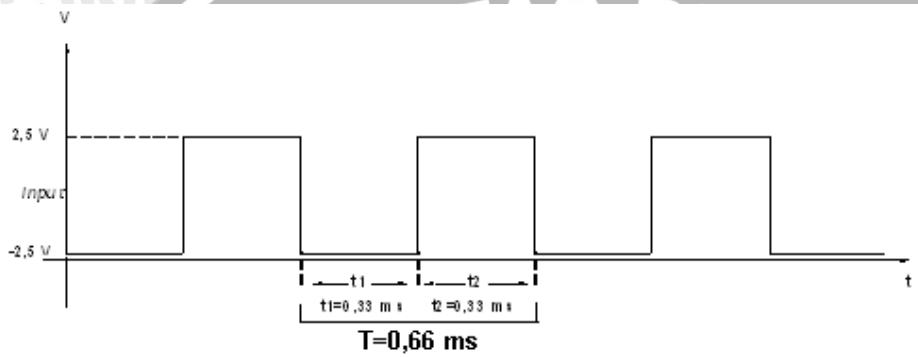
$$D = \frac{4700 + 47000}{4700 + 94000} \times 100\%$$

$$D = 0,52 \times 100\%$$

$$D = 52\%$$

$$D \approx 50\%$$

Dari perhitungan *duty cycle* diperoleh perbandingan antara t_1 dan t_2 adalah 50 %.



Gambar 4.5 Blok periode pada *generator uji*.

Sumber: Perancangan.

- Perhitungan *duty cycle* berdasarkan periode (T) :

$$f = 1,5\text{KHz}$$

$$T = \frac{1}{f}$$

$$T = \frac{1}{1,5\text{KHz}} = 0,66\text{ms.}$$

$$D = \frac{t1}{t1 + t2} \times 100\%$$

$$D = \frac{0,33}{0,33 + 0,33} \times 100\%$$

$$D = 50\%$$

Dari perhitungan *duty cycle* berdasarkan periode (T) didapat jika perbandingan antara t_1 dan t_2 sama dengan perhitungan sebelumnya yaitu 50 %.

- Nilai dari t_1 dan t_2 didapatkan dari perhitungan berikut :

$$T = t1 + t2$$

$$0,66ms = t1 + t2$$

- Nilai dari t1 adalah :

$$t1 = -\frac{0,66ms}{2} = -0,33ms.$$

- Nilai dari t2 adalah :

$$t2 = \frac{0,66ms}{2} = 0,33ms.$$

- Perhitungan frekuensi yang dihasilkan *generator* uji sesuai dengan persamaan berikut ini :

$$f = \frac{1,44}{(Ra + 2Rb)C} \quad (4.5)$$

dan perhitungan frekuensi yang dihasilkan oleh *generator* uji sesuai dengan persamaan 4.5 adalah :

$$f = \frac{1,44}{(4700 + 94000)10 \cdot 10^{-9}}$$

$$f = \frac{1,44}{98700 \cdot 10 \cdot 10^{-9}}$$

$$f = 1,45 \text{ KHz} \cdot$$

$$f \approx 1,5 \text{ KHz} \cdot$$

Dari perhitungan frekuensi didapat jika frekuensi yang dihasilkan oleh *generator* uji adalah sebesar 1,5 KHz.

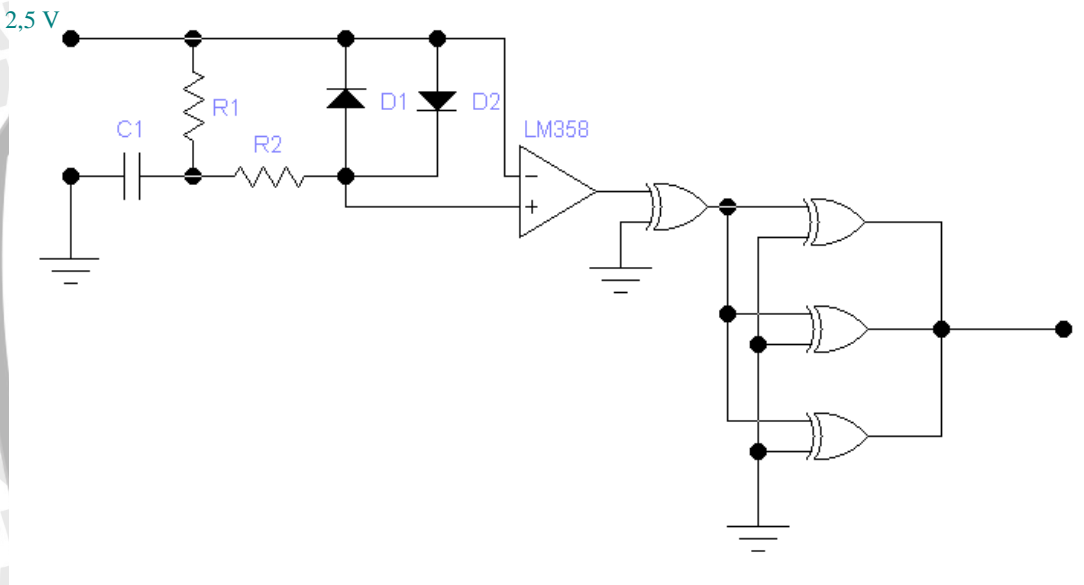
4.5.4 Perancangan Pendeteksi Beda Fasa

Perancangan pendeteksi beda fasa bertujuan untuk mengetahui besarnya selisih fasa antara sinyal informasi yang dikirim dengan sinyal informasi yang diterima. Sinyal informasi dikatakan baik apabila beda fasa yang terukur $\approx 0^\circ$ (mendekati 0°), dan dikatakan tidak baik bila beda fasa yang terukur $\approx 180^\circ$ (mendekati 180°). Di dalam perancangan ini digunakan sinyal informasi berbentuk gelombang balok dengan frekuensi = 1,5 KHz, D = 50%, dan A *peak to*

$peak = 5$ volt. Pada perancangan ini terdapat beberapa bagian yang membentuk pendeteksi beda fasa, antara lain: rangkaian proteksi dan komparator, rangkaian komparator beda fasa berupa gerbang EXOR, dan rangkaian integrator.

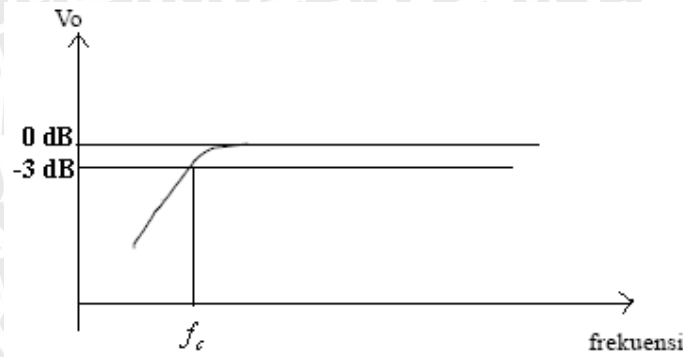
4.5.4.a Perancangan Rangkaian Proteksi dan Komparator

Dalam perancangan seperti yang terlihat dalam Gambar 4.4 terdapat sebuah rangkaian yang terbentuk dari komponen R1, R2, D1, D2, tujuan dari dibentuknya rangkaian tersebut adalah sebagai rangkaian proteksi pengemudian lebih (*over drive*) dengan ambang batas 2,5 volt.



Gambar 4.6 Rangkaian Komparator dalam perancangan.
Sumber: Perancangan.

Pada Gambar 4.6 terdapat rangkaian C1, R1 dan C1, R2, di mana rangkaian tersebut membentuk filter (*High Pass Filter*) dengan *slope* -20 dB/oktaf, yang berfungsi untuk mencegah tegangan DC yang akan masuk (*DC blocking*).



Gambar 4.7 Output HPF pada rangkaian komparator .

Sumber: Perancangan.

Perhitungan untuk mengetahui nilai f_c adalah :

$$f_c = \frac{1}{2\pi \cdot R \cdot C} \quad (4.6)$$

Penjabaran dari persamaan 4.6 pada rangkaian komparator adalah untuk mendapatkan, nilai dari f_c , dan perhitungannya adalah sebagai berikut :

$$f_c = \frac{1}{2(3,14) \cdot 159K\Omega \cdot 100nF}$$

$$f_c = \frac{1}{6,28 \cdot 159 \cdot 10^3 \cdot 100 \cdot 10^{-9}}$$

$$f_c = \frac{1}{6,28 \cdot 159 \cdot 10^{-4}}$$

$$f_c = 10Hz .$$

Dari perhitungan yang sudah dilakukan diperoleh nilai f_c sebesar 10 Hz, dengan asumsi sinyal informasi yang digunakan berupa sinyal *audio* dengan *range* 10 Hz-20 KHz, tetapi karena di pasaran tidak tersedia *resistor* dengan nilai 159K Ω maka yang digunakan adalah *resistor* dengan nilai 150K Ω .

4.5.4.b Perancangan Rangkaian Komparator Beda Fasa

Rangkaian komparator beda fasa berfungsi untuk mendeteksi besarnya beda fasa, di mana rangkaian pengukur beda fasa tersusun dari beberapa gerbang EXOR.

Tabel 4.1 Tabel Kebenaran Gerbang EXOR.

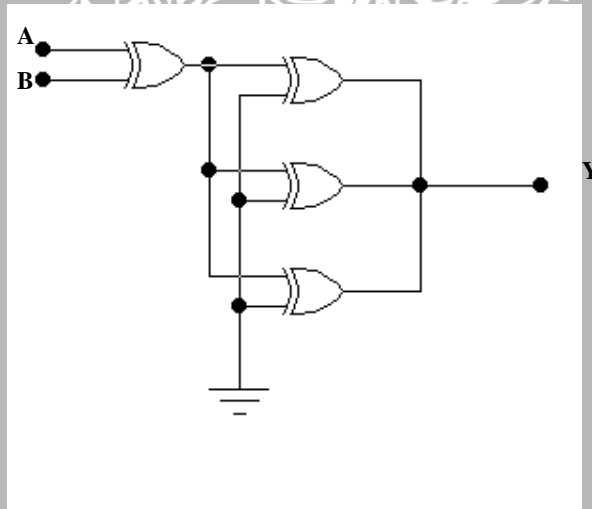
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Sumber: Mikroelektronika, Sistem Digital dan Rangkaian Analog
 Jilid I, Jacob Millman, 1986.

Keterangan pada Tabel 4.1 sama dengan pernyataan berikut, jika A=1 atau B=1 tetapi tidak bersamaan maka Y=1, dimana dalam persamaan *boole* ditulis :

(4.7)

$$Y = (A + B)AB$$



Gambar 4.8 Susunan gerbang EXOR pada rangkaian komparator beda fasa.
Sumber: Perancangan.

Berdasarkan Tabel 4.1 , maka perancangan rangkaian komparator beda fasa bertujuan untuk mengetahui selisih fasa antara *input* pada pemancar dengan *output* pada penerima. Untuk membuktikannya perhatikan analisa berikut, misalkan 0 adalah *input* dan 1 adalah *output* maka akan didapatkan :

Tabel 4.2 Tabel Kebenaran Gerbang EXOR Perancangan.

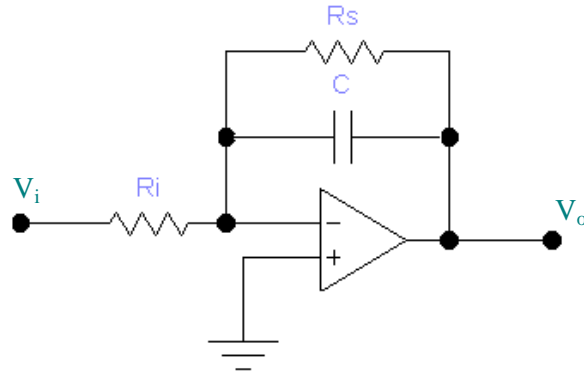
A	B	Y
<i>input</i>	<i>input</i>	0^0
<i>Input</i>	<i>output</i>	180^0
<i>output</i>	<i>input</i>	180^0
<i>output</i>	<i>output</i>	0^0

Sumber: Perancangan.

Dalam Tabel 4.2 dapat disimpulkan bahwa pengukuran yang dilakukan dari titik *input* ke titik *input* atau pengukuran dari titik *output* ke titik *output* tidak terjadi selisih beda fasa atau 0^0 , hal ini karena pengukuran dilakukan pada titik yang sama sehingga tidak terjadi selisih fasa. Namun bila pengukuran dilakukan dari titik *input* ke titik *output* atau sebaliknya maka akan didapatkan selisih fasa 180^0 , karena pengukuran dilakukan dari titik yang berbeda. Analisa ini dilakukan untuk membuktikan bahwa perancangan yang dilakukan sudah sesuai dengan teori yang ada. Sedangkan untuk pengukuran selisih fasa yang sebenarnya akan dilakukan dengan menghubungkan antara *input* pada pemancar dengan *output* dengan penerima, di mana dari pengukuran ini akan didapatkan selisih fasa setelah melalui proses transmisi.

4.5.4.c Perancangan Rangkaian *Integrator*

Rangkaian *integrator* digunakan untuk mencari nilai hasil integrasi dari sinyal input. Rangkaian *integrator* op-amp sendiri berasal dari rangkaian *inverting* dengan tahanan umpan baliknya diganti dengan kapasitor. Pada perancangan ini tujuan utama dibentuknya rangkaian *integrator* adalah untuk mengintegrasikan *output* dari komparator fasa agar menjadi tegangan DC untuk diteruskan ke analog multiplexer (AMUX).



Gambar 4.9 Rangkaian *integrator* yang digunakan.
Sumber: Perancangan.

Untuk mencari nilai tegangan yang keluar dari rangkaian *integrator* seperti yang terlihat pada Gambar 4.9 digunakan persamaan berikut :

$$V_o = -\frac{1}{R_i \cdot C} \int_0^t V_i dt \quad (4.8)$$

Perhitungan pada persamaan 4.8 tersebut berlaku bila frekuensi *input* lebih besar f_c , dan untuk mencari nilai f_c pada rangkaian integrator digunakan persamaan berikut :

$$f_c = \frac{1}{2\pi \cdot R_s \cdot C} \quad (4.9)$$

Dalam perancangan yang dilakukan dijabarkan nilai kapasitor yang digunakan sebesar $2 \mu F$, besarnya nilai kapasitor tersebut didapatkan dari 2 kapasitor yang diparalel, di mana besar masing-masing kapasitor adalah $1 \mu F$. Besar nilai R_s adalah sepuluh kali besarnya R_i ($R_s = 10 \cdot R_i$). Di perancangan ini digunakan R_i sebesar 800Ω , sehingga didapatkan nilai R_s sebesar $8K\Omega$. Dari persamaan 4.8 didapatkan :

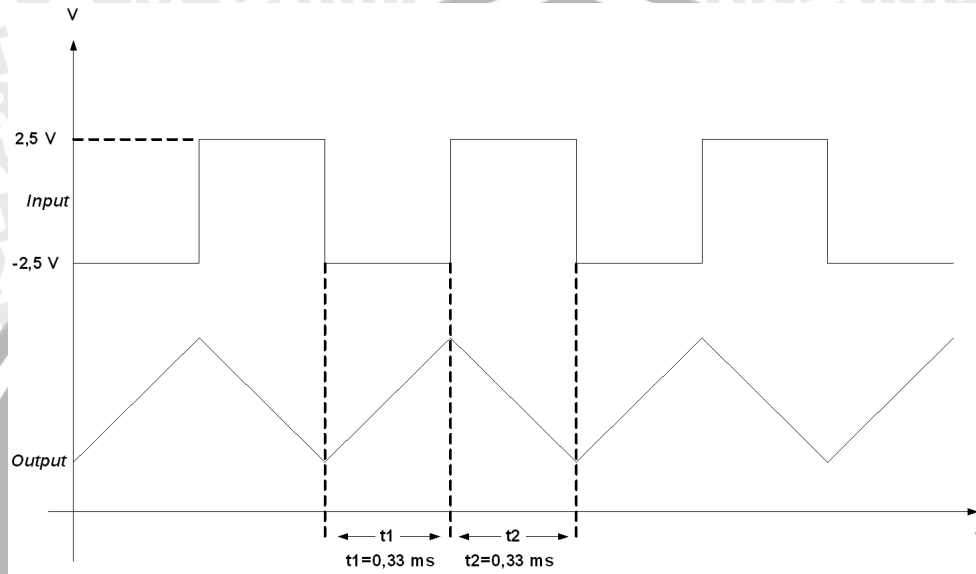
$$f_c = \frac{1}{2(3,14) \cdot 8 \cdot 10^3 \cdot 2 \cdot 10^{-6}}$$

$$f_c = \frac{1}{6,28 \cdot 16 \cdot 10^{-3}}$$

$$f_c = 9,95 \text{ Hz.}$$

Dari perhitungan di atas didapatkan $f_c = 9,95 \text{ Hz}$, pemilihan besarnya nilai R dan C yang dipilih dalam perancangan rangkaian *integrator* ditentukan berdasarkan

besarnya frekuensi yang melalui rangkaian sebelumnya yaitu rangkaian proteksi dan komparator, hal ini perlu dilakukan agar frekuensi yang akan diteruskan ke rangkaian selanjutnya tidak berubah. sedangkan nilai frekuensi $input = 10 \text{ Hz}$, sehingga frekuensi $input$ lebih besar dari f_c . Karena frekuensi $input$ lebih besar dari f_c , maka persamaan 4.9 berlaku.



Gambar 4.10 Estimasi tegangan *output integrator*.
Sumber: Perancangan.

Keterangan Gambar 4.10 adalah sebagai berikut :

$D = \text{Dutycycle} \approx 50\%$

$f = 1,5\text{KHz}$

$T = 0,66 \text{ ms}$

$t1 = -0,33 \text{ ms}$

$t2 = 0,33 \text{ ms}$

Untuk perhitungan tegangan *output* akan dilakukan dua perhitungan dalam mencari nilai tegangan *output*, yaitu meliputi perhitungan pada $t1$ dan $t2$, adapun perhitungannya adalah sebagai berikut :

- Perhitungan pada $t1$

$$V_o |_{t_1} = -\frac{1}{(1 \cdot 10^6)(2 \cdot 10^{-6})} \int_0^{-0,00033} (-2,5) dt$$

$$V_o |_{t_1} = -\frac{1}{2} (-2,5 \cdot -0,00033)$$

$$V_o |_{t_1} = -\frac{1}{2} \cdot 0,000825$$

$$V_o |_{t_1} = -0,00041 \text{ volt.}$$

- Perhitungan pada t2

$$V_o |_{t_2} = \frac{1}{(1 \cdot 10^6)(2 \cdot 10^{-6})} \int_0^{0,00033} (2,5) dt$$

$$V_o |_{t_2} = \frac{1}{2} (2,5 \cdot 0,00033)$$

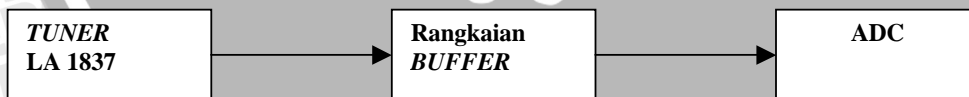
$$V_o |_{t_2} = \frac{1}{2} \cdot 0,000825$$

$$V_o |_{t_2} = 0,00041 \text{ volt.}$$

Dari perhitungan yang sudah dilakukan didapatkan nilai tegangan *output* pada t1 dan t2 sebesar $\pm 0,00041$ volt, namun yang membedakan pada t1 tegangan *output* = $-0,00041$ volt, sedangkan pada t2 tegangan *output* = $0,00041$ volt. Hal ini karena tegangan *input* yang digunakan berbeda kondisinya.

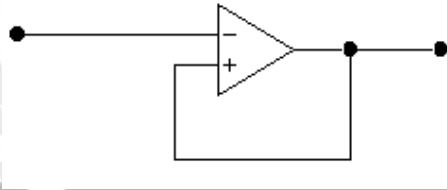
4.5.5 Perancangan Pendeteksi Sinyal / *Strength Meter*

Perancangan pendeteksi sinyal yang dipancarkan / *strength meter*, bertujuan agar alat yang dirancang dapat memberikan besarnya sinyal yang dipancarkan oleh pemancar terhadap penerima. Pada perancangan ini digunakan sebuah rangkaian *buffer* (penyanggah), dimana rangkaian *buffer* sendiri berfungsi agar *tuner* tidak terbebani oleh rangkaian penguat dan ADC.



Gambar 4.11 Blok perancangan rangkaian *buffer*.
Sumber: Perancangan.

Secara teoritis ditunjukkan bahwa Z_{in} *buffer* sama dengan impedansi intrinsik op amp. Untuk lebih jelasnya perhatikan Gambar 4.12 berikut :



Gambar 4.12 Skema rangkaian *buffer*.
Sumber: Perancangan.

Rangkaian *buffer* pada Gambar 4.12 memenuhi :

$$Z_{in} = Z_{in} \text{ intrinsik op amp} = 1M\Omega \text{ (dari datasheet)}$$

$$Z_{out} = Z_{out} \text{ op amp}$$

$$BW = \frac{GBP}{A_v}, \text{ dimana } A_v = 1$$

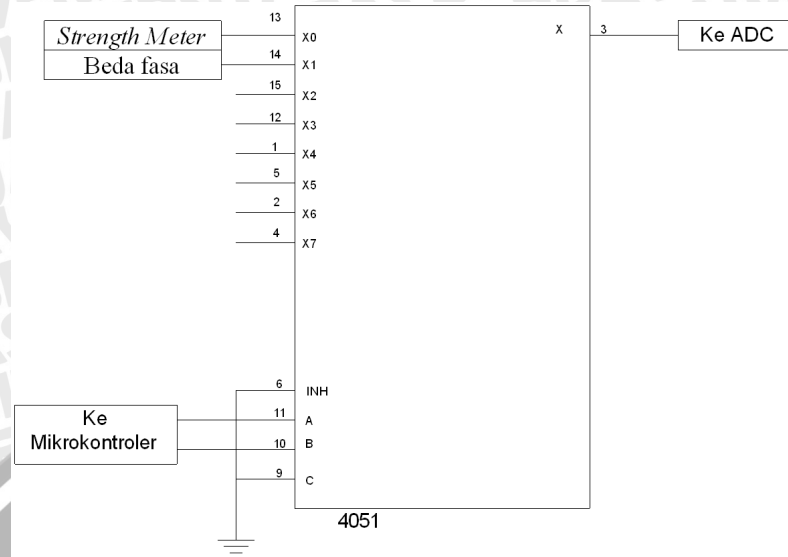
sehingga diperoleh :

$$BW = GBP \text{ (dari datasheet)}$$

diketahui bahwa GBP LM 358 adalah 1 MHz, sehingga besar *bandwidth* sama dengan GBP yaitu sebesar 1 MHz.

4.5.6 Perancangan Analog Multiplekser

Pada perancangan analog multiplekser ini, yang digunakan adalah analog multiplekser 4051 (AMUX 4051). Pada dasarnya secara prinsip AMUX 4051 ini berfungsi sebagai saklar analog untuk menentukan masukan yang akan diteruskan ke mikrokontroler.



Gambar 4.13 Rangkaian AMUX 4051 yang dirancang.
Sumber: Perancangan.

Pada gambar di atas terlihat bahwa fungsi utama dari analog multiplekser adalah untuk memilih jenis masukan yang akan diteruskan ke mikrokontroler, terdapat 2 jenis masukan yang akan dipilih oleh multiplekser, yaitu kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa.

Tabel 4.3 Tabel Kebenaran AMUX 4051

A2	A1	A0	Y0 (Output)
0	0	0	<i>Strength Meter</i>
0	0	1	Beda Fasa
0	1	0	Unused
0	1	1	Unused

Sumber: Perancangan.

Dari tabel kebenaran diatas didapat :

A0, A1, A2 = 0, maka keluaran yang diteruskan ke mikrokontroler adalah *strength meter*.

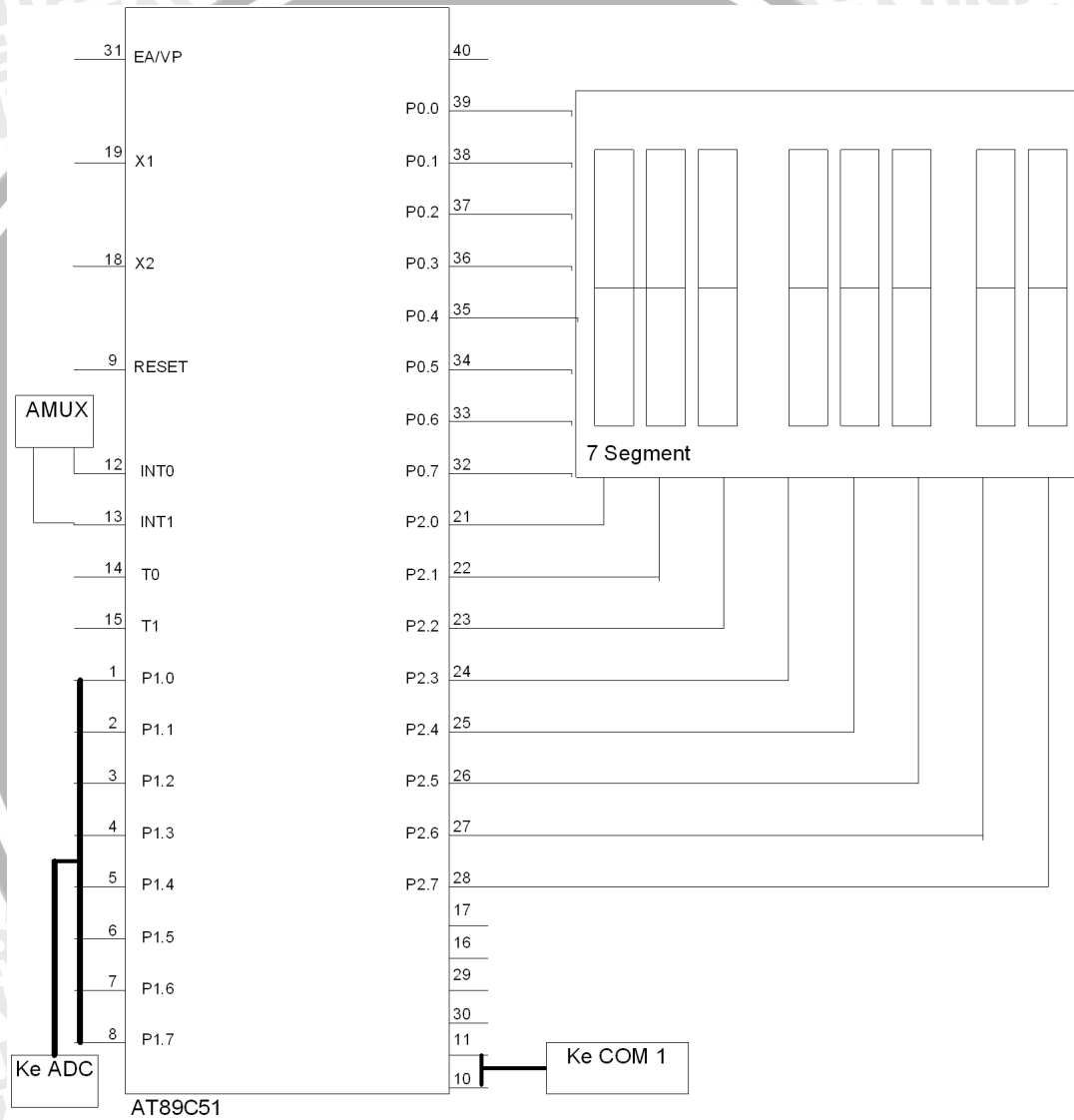
A0 = 1, A1, A2 = 0, maka keluaran yang diteruskan ke mikrokontroler adalah beda fasa.

A1 = 1, A0, A2 = 0, tidak digunakan.

A0, A1 = 1, A2 = 0, tidak digunakan.

4.5.7 Perancangan Tampilan 7 Segment

Perancangan tampilan 7 segment digunakan untuk menampilkan hasil perhitungan yang dilakukan oleh mikrokontroler. Pada perancangan ini digunakan beberapa komponen yang dapat mewujudkan sebuah tampilan yang diinginkan, antara lain transistor, dioda, resistor. Disamping itu akan juga dilakukan perhitungan terhadap hal-hal yang berkaitan dengan dengan komponen tersebut.



Gambar 4.14 Rangkaian interface mikrokontroler dengan 7 segment.

Sumber: Perancangan.

$$R = \frac{V_{in} - V_{out}}{I_{LED}}$$

$$R = \frac{2,4\text{volt} - 1,8\text{volt}}{20\text{mA}}$$

$$R = \frac{0,6\text{volt}}{0,02\text{A}}$$

$$R = 30\Omega.$$

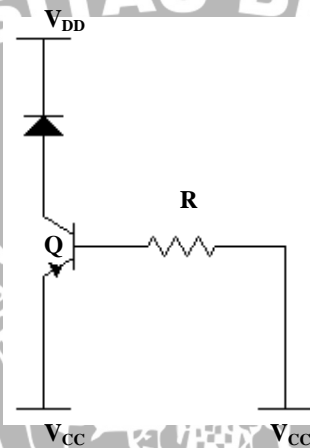
Keterangan :

V_{in} = Tegangan *output port* mikroprosesor P₀-P₇ (2,4 volt).

V_{LED} = Tegangan yang terdapat pada LED (1,8 volt).

I_{LED} = Arus yang mengalir di LED (20 mA).

Sehingga :



Gambar 4.15 Rangkaian pada 7 segment.

Sumber: Perancangan

4.5.8 Perancangan *Interface* Mikrokontroler dengan Komputer

Perancangan *interface* antara mikrokontroler sistem dan komputer sistem, dimana data yang ada pada mikrokontroler sistem akan dikirim ke komputer melalui *serial port* (COM Port 1) menggunakan media RS-232 di mana sebelumnya data perhitungan sinyal yang dipancarkan / *strength* meter, serta beda fasa sudah diproses dengan *serial buffer*. Hal ini bertujuan agar data yang terdapat di mikrokontroler dapat diakses dengan *valid* di komputer sistem. Register yang digunakan untuk mengatur komunikasi serial terdapat pada *serial control* (SCON).

Tabel 4.4 Alokasi Bit SCON.

MSB								LSB
SM0	SM1	SM2	REN	TB8	RB8	TI	RI	

Sumber: Innovative Electronics, Surabaya 2004.

Model pengiriman data yang dilakukan adalah *8-bit shift register*, dimana data dikirimkan dan diterima melalui *pin* RXD sedangkan *clock* dikirimkan dan diterima melalui *pin* TXD. Pengiriman data *8 bit* dilakukan dengan mengirimkan *Least Significant Bit* (LSB) terlebih dahulu. Dalam perancangan ini *baud rate* yang digunakan adalah sebesar 9600 bps, di mana *baud rate* ini didapatkan dari 1/12 frekuensi osilator.

$$\text{Baud_Rate} = \frac{\text{Frekuensi_Osilator}}{12} \quad (4.10)$$

Pada Tabel 4.5 dijelaskan tentang beberapa jenis *baud rate* yang biasa digunakan dalam komunikasi serial.

Tabel 4.5 Nilai dan Konfigurasi *Baud Rate*.

Serial		Timer 1				
Mode	Baud Rate	Frekuensi Osilator	SMOD	C/T	Mode	Reload
0	1,6667 Mbps (max.)	20 MHz	X	X	X	X
2	625 Kbps (max.)	20 MHz	1	X	X	X
1, 3	104,1667 Kbps (max.)	20 MHz	1	0	2	FFh
1, 3	19,2 Kbps	11,0592 MHz	1	0	2	FDh
1, 3	9,6 Kbps	11,0592 MHz	0	0	2	FDh
1, 3	4,8 Kbps	11,0592 MHz	0	0	2	FAh
1, 3	2,4 Kbps	11,0592 MHz	0	0	2	F4h
1, 3	1,2 Kbps	11,0592 MHz	0	0	2	E8h
1, 3	137,5 bps	11,9856 MHz	0	0	2	1Dh
1, 3	110 bps	6 MHz	0	0	2	72h
1, 3	110 bps	12 MHz	0	0	1	FEEBh

Sumber: Innovative Electronics, Surabaya 2004.

4.6 Perancangan Perangkat Lunak

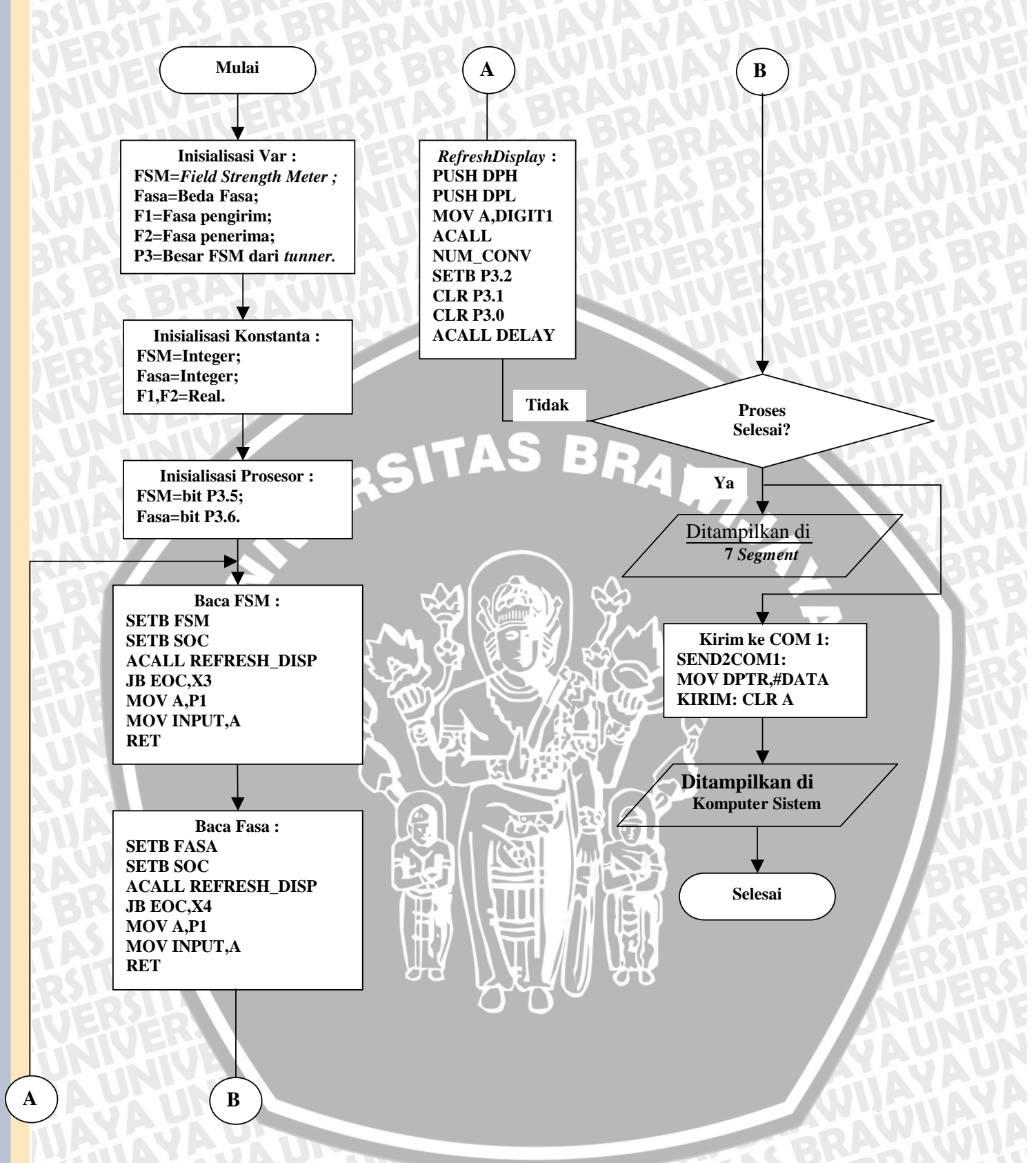
Dalam perancangannya, akan digunakan 2 jenis perangkat lunak pada alat ini. Adapun perangkat lunak yang digunakan adalah program *assembly* untuk mikrokontroler AT89C51 dan program *borland delphi 7* sebagai *data base* di komputer.

4.6.1 Perancangan Program *Assembly* AT89C51

Setelah mengetahui perancangan perangkat kerasnya, maka diperlukan perangkat lunak untuk kemudahan dalam pengendaliannya. Bahasa yang digunakan dalam perancangan ini adalah bahasa *assembly* untuk mikrokontroler AT89C51. Perangkat lunak yang direncanakan untuk menjalankan sistem ini terdiri atas proses deteksi dari penerima dan proses pengiriman data ke komputer.

Diagram alir di bawah ini merupakan tahapan prosedur kerja dari program *assembly* yang digunakan. Saat alat diaktifkan, mikrokontroler akan melakukan inisialisasi *baud rate*, dan alamat memori data. Langkah pertama adalah proses pengiriman sinyal dari pemancar ke penerima, setelah itu sebelum masuk ke mikrokontroler, *input* (masukan) yang terdiri dari 2 bagian yaitu kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa masuk ke analog multiplexer, setelah itu saklar *selector* akan mengirimkan secara bergantian 2 *input* (masukan) tersebut ke mikrokontroler, tetapi sebelum masuk ke mikrokontroler masukan yang diterima akan di konversi terlebih dahulu di ADC (*Analog Digital Converter*), hal ini bertujuan untuk merubah sinyal analog menjadi digital.

Perhatikan Gambar 4.16 dapat dijelaskan bahwa langkah selanjutnya adalah proses perhitungan yang dilakukan oleh mikrokontroler, dalam proses perhitungan akan meliputi perhitungan kuat sinyal yang dipancarkan / *strength meter*, serta beda fasa. Setelah itu maka langkah selanjutnya adalah menampilkan hasil dari perhitungan yang dilakukan oleh mikrokontroler, tampilan pada alat akan menunjukkan besarnya yang diukur dalam bentuk angka dengan menggunakan 7 *segment*. Setelah itu akan data yang sudah diperoleh akan dikirim ke komputer sistem, untuk menampilkan data perhitungan secara *visual* di komputer sistem.

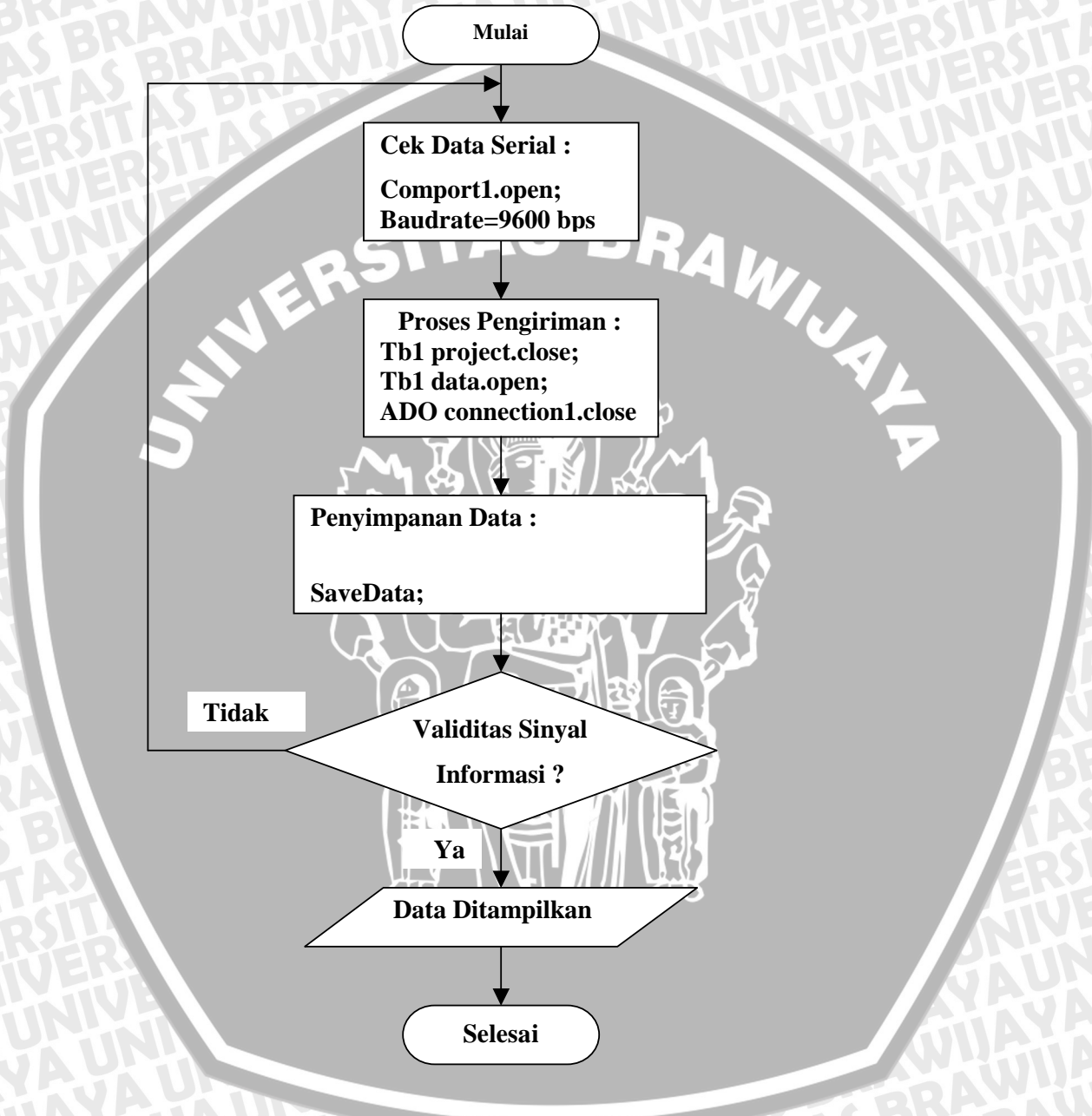


Gambar 4.16 Diagram Alir Proses Perhitungan di Mikrokontroler.

Sumber: Perancangan.

4.6.2 Perancangan Program *Borland Delphi 7*

Program *borland delphi 7* ini digunakan untuk mempermudah dalam proses pembacaan hasil perhitungan yang sudah dilakukan oleh mikrokontroler. Untuk lebih jelasnya perhatikan Gambar 4.17



Gambar 4.17 Diagram alir proses menampilkan data di *borland delphi 7*.

Sumber: Perancangan.

Pertama kali akan dilakukan cek data *serial* yang meliputi *baud rate*, dll. Setelah itu akan dilakukan pengiriman hasil perhitungan, dimana akan dikirim ke komputer dengan dihubungkan melalui *com port 1* komputer, dengan media kabel RS

232 , yang sebelumnya diproses dengan *serial buffer*. Proses selanjutnya adalah penyimpanan data, setelah data tersimpan maka akan ditentukan apakah perlu penambahan data untuk proses keakuratan pengukuran. Bila diperlukan maka akan kembali ke proses awal, jika tidak maka data yang diperoleh akan ditampilkan.



BAB V

PENGUJIAN DAN ANALISA DATA

Setelah merancang keseluruhan sub sistem, selanjutnya dilakukan pengujian untuk mengetahui unjuk kerja masing-masing sub sistem, apakah sudah sesuai dengan yang direncanakan. Setelah dilakukan pengujian terhadap masing-masing sub sistem, selanjutnya seluruh sub sistem tersebut digabungkan untuk membentuk suatu alat pendeteksi kualitas sinyal informasi pada FM dan menguji sistem secara keseluruhan. Bagian-bagian yang diuji adalah:

1. Pengujian komunikasi alat dengan komputer.
2. Pengujian pengukuran beda fasa pada alat.
3. Pengujian berdasarkan jarak antara pemancar dan penerima.
4. Pengujian berdasarkan tinggi antena pemancar.
5. Pengujian pengukuran pada pemancar radio FM di Malang.

Setelah semua bagian di atas diuji, langkah berikutnya adalah pengujian sistem secara keseluruhan. Dalam setiap pengujian dijelaskan tujuan pengujian dan analisa data hasil pengujian.

5.1 Pengujian Komunikasi Alat dengan Komputer

5.1.1 Tujuan

Untuk menguji apakah terjadi sambungan antara alat dengan komputer.

5.1.2 Peralatan yang Digunakan

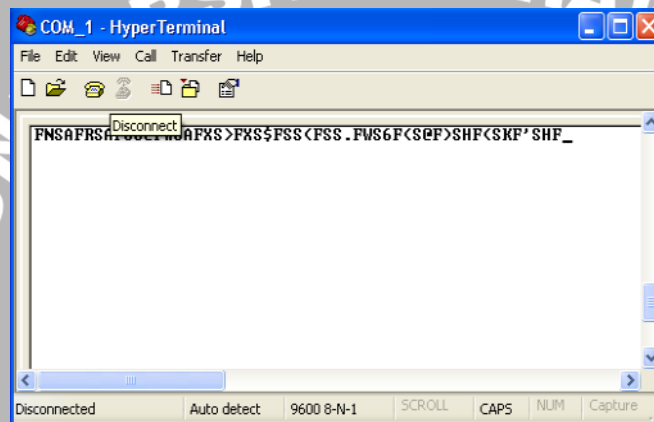
1. Program *hyper terminal* pada komputer sebagai *software* untuk pengujian *koneksi* antara alat dan sistem komputer.
2. Komputer dengan fasilitas COM 1 (DB9), berfungsi sebagai media komunikasi *serial* pada komputer.
3. Kabel data *serial*, yang berfungsi sebagai media untuk *transfer* komunikasi antara alat dengan sistem komputer.

5.1.3 Prosedur Pengujian

1. Hubungkan alat dengan komputer menggunakan kabel data *serial*.
2. Menjalankan program *hyper terminal* pada komputer.

5.1.4 Hasil Pengujian

Hasil pengujian pada *hyper terminal* menunjukkan bahwa data yang dikirimkan oleh sistem mikrokontroler sudah masuk dan dapat diterima oleh sistem komputer, untuk kemudian diproses dan ditampilkan dalam bentuk *visual* oleh *Borland Delphi 7*.



Gambar 5.1 Hasil Pengujian Komunikasi Alat dengan Komputer.

Sumber: Perancangan.

5.1.5 Analisis Hasil Pengujian

Dari hasil pengujian komunikasi antara alat dengan komputer dengan menggunakan *hyper terminal* didapatkan :

1. Simbol huruf F merupakan kuat sinyal / *strength meter* dan selalu diikuti konstanta, hal ini menunjukkan bahwa terjadi pengiriman besarnya kuat sinyal / *strength meter* dari alat ke komputer.
2. Simbol huruf S merupakan beda fasa dan selalu diikuti konstanta, hal ini menunjukkan bahwa terjadi pengiriman besarnya beda fasa dari alat ke komputer.
3. Apabila di *hyper terminal* yang diikuti konstanta hanya salah satu dari antara F atau S, maka hanya terjadi satu jenis pengukuran saja.

5.2 Pengujian Pengukuran Beda Fasa pada Alat

5.2.1 Tujuan

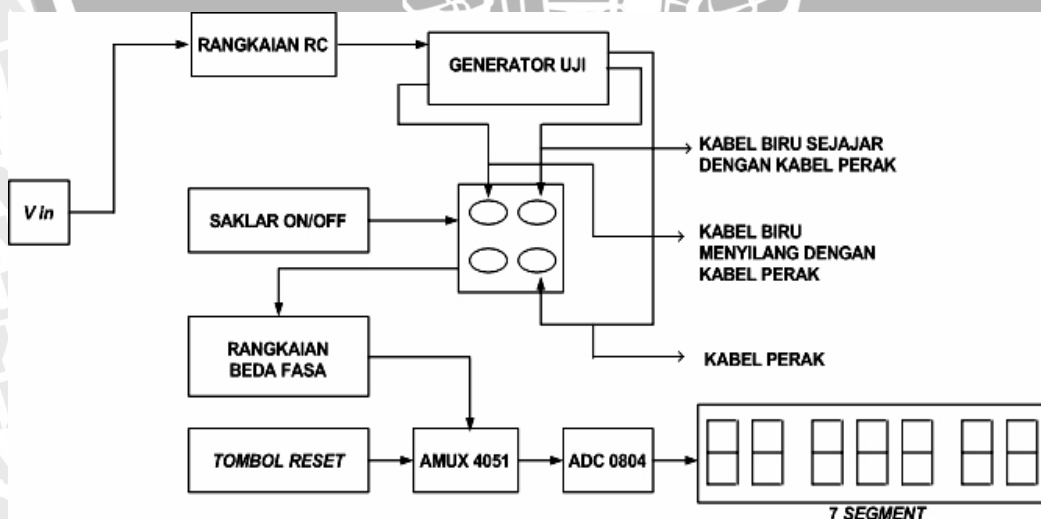
Untuk mengetahui apakah sistem pengukuran beda fasa yang dilakukan oleh alat sudah sesuai dengan yang direncanakan.

5.2.2 Peralatan yang Digunakan

1. Rangkaian sistem alat.
2. Kabel dengan pin berwarna perak adalah kabel yang terhubung pada *output* dari *generator* uji dan berfungsi sebagai media untuk meneruskan sinyal *output* ke rangkaian beda fasa.
3. Kabel dengan pin berwarna biru adalah kabel yang terhubung pada *input* dari *generator* uji dan berfungsi sebagai media untuk meneruskan sinyal *input* ke rangkaian beda fasa, untuk kemudian dibandingkan dengan sinyal *output*.
4. Saklar yang berfungsi untuk menentukan pengukuran alat dengan menggunakan pemancar atau tidak, apabila saklar *on* maka pengukuran dilakukan dengan pemancar mini 10 mW, tetapi bila saklar *off* maka pengukuran dilakukan tanpa pemancar.

5.2.3 Prosedur Pengujian

1. Perhatikan Gambar 5.2 , nyalakan alat yang sudah dirancang.



Gambar 5.2 Blok Diagram Pengujian Pengukuran Beda Fasa.

Sumber: Perancangan.

2. Tentukan saklar pada alat di posisi *off*, untuk memastikan bahwa alat melakukan pengukuran tanpa pemancar.
3. Tekan tombol *reset* pada alat agar sistem mikrokontroler pada alat siap untuk melakukan perhitungan yang baru.
4. Tancapkan kabel pin biru sejajar di atas kabel pin perak, untuk mendapatkan hasil pengukuran beda fasa 0° .



Gambar 5.3 Posisi Kabel Sejajar ke Atas.

Sumber: Perancangan.

5. Setelah langkah 4 dan dilakukan pengukuran, lepaskan kabel pin biru kemudian tancapkan kabel pin biru kembali dengan posisi menyilang dengan kabel pin perak, pengukuran ini bertujuan untuk mendapatkan besar beda fasa 180° .



Gambar 5.4 Posisi Kabel Menyilang.

Sumber: Perancangan.

5.2.4 Hasil Pengujian

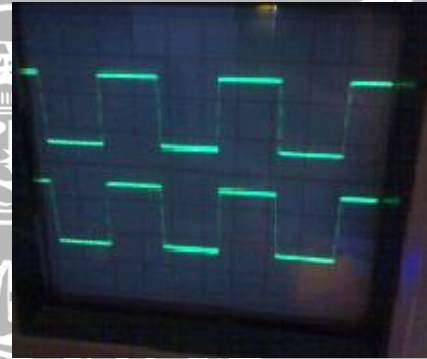
Pengujian dilakukan di ruang tertutup dengan kondisi cuaca yang baik. pengujian sendiri dilakukan pada tanggal 8 Desember 2008, malam hari pukul 19.00 WIB. Dalam pengujian digunakan *osiloskop* sebagai alat pendukung, hal ini bertujuan agar data yang diperoleh lebih jelas, pengukuran dengan *osiloskop* menggunakan ketentuan sebagai berikut : $T / div = 0,2ms / div$,

$$V / div = 2volt / div .$$

- Pengujian pada saat posisi kabel sejajar ke atas (langkah 4).



Gambar 5.5 (a)



Gambar 5.5 (b)

Gambar 5.5 (a) Hasil Pengukuran pada Alat, (b) Gambar Sinyal pada *Osiloskop*.
Sumber: Perancangan.

- Pengujian pada saat posisi kabel menyilang (langkah 5).



Gambar 5.6 (a)



Gambar 5.6 (b)

Gambar 5.6 (a) Hasil Pengukuran pada Alat, (b) Gambar Sinyal pada *Osiloskop*.
Sumber: Perancangan.

5.2.5 Analisis Hasil Pengujian

Berdasarkan pengujian tersebut dapat diambil kesimpulan bahwa sistem telah bekerja sesuai dengan yang direncanakan dan bisa digunakan untuk mendeteksi besarnya beda fasa.

Tabel 5.1 Tabel Perbandingan Posisi Kabel.

No.	Posisi Kabel	Hasil Pengukuran ($^{\circ}$)
1	Sejajar ke atas	0
2	Menyilang	180

Sumber: Perancangan.

Seperti yang terdapat pada Tabel 5.1 diperoleh bahwa pada saat kondisi kabel sejajar ke atas hasil pengukuran adalah 0° , sedangkan pada saat posisi kabel menyilang hasil pengukuran adalah 180° .

5.3 Pengujian Berdasarkan Jarak Antara Pemancar dan Penerima

5.3.1 Tujuan

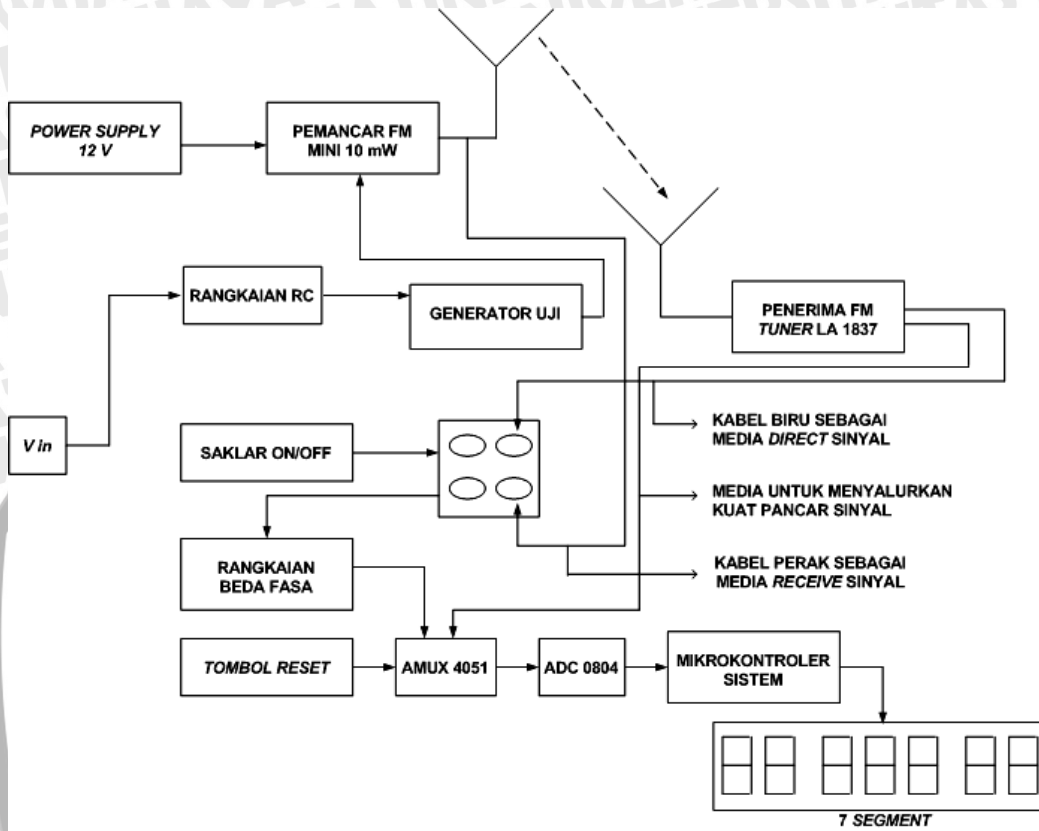
Untuk mendeteksi kuat sinyal yang dipancarkan / *strength meter* dan besarnya beda fasa yang diterima oleh penerima, berdasarkan jarak transmisi antara pemancar FM mini 10 mW dengan radio sebagai penerima.

5.3.2 Peralatan yang Digunakan

1. Rangkaian sistem alat.
2. Pemancar FM mini 10 mW.
3. Komputer sistem sebagai media tampilan secara *visual* dari hasil pengujian dengan menggunakan *Borland Delphi 7*.
4. Kabel data *serial*, yang berfungsi sebagai media untuk *transfer* komunikasi antara alat dengan sistem komputer.

5.3.3 Prosedur Pengujian

1. Nyalakan alat yang sudah dirancang.



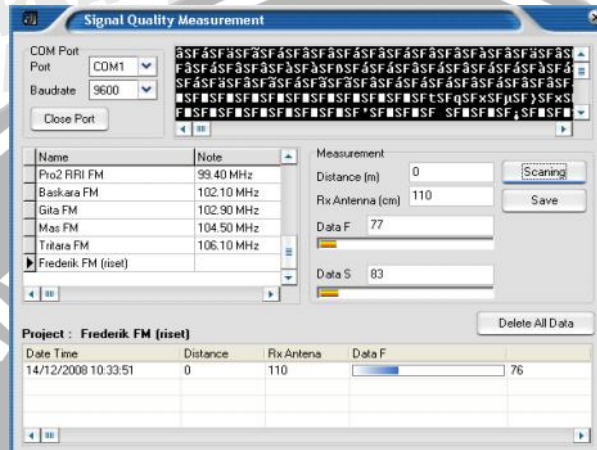
Gambar 5.7 Blok Diagram Pengujian Pengukuran Kuat Sinyal dan Beda Fasa.

Sumber: Perancangan.

2. Tentukan saklar pada alat di posisi *on*, untuk memastikan bahwa alat melakukan pengukuran dengan pemancar FM mini 10 mW.
3. Tekan tombol *reset* pada alat agar sistem mikrokontroler pada alat siap untuk melakukan perhitungan yang baru.
4. Pada penerima pilih frekuensi radio di 89,05 MHz.
5. Tentukan jarak transmisi antara pemancar dan penerima sesuai dengan kebutuhan pada saat proses pengukuran.
6. Hasil pengukuran tampil di 7 *segment* dan hasil pengukuran secara *visual* tampil di komputer sistem.

5.3.4 Hasil Pengujian

Pengujian dilakukan di ruang tertutup dengan kondisi cuaca yang baik. pengujian sendiri dilakukan pada tanggal 8 Desember 2008, malam hari pukul 19.00 WIB. Dalam hasil pengujian ini juga diberikan data dalam bentuk tabel, untuk lebih jelasnya perhatikan Tabel 5.2.



Gambar 5.8 Tampilan pada Borland Delphi 7.
Sumber: Perancangan.



Gambar 5.9 Pengujian pada Alat dengan Jarak Pancar 300 cm.
Sumber: Perancangan.

Tabel 5.2 Tabel Perbandingan Berdasarkan Jarak Pancar Antena.

No.	Jarak Pemancar ke Penerima (cm)	Level Tegangan Kuat Pancar Sinyal / <i>Strength meter</i> (mV)	Beda Fasa ($^{\circ}$)	Kualitas Suara
1	50	226	1	Baik
2	100	223	4	Baik
3	150	220	17	Baik
4	200	210	34	Baik
5	250	154	50	Kurang Baik
6	300	78	79	Kurang Baik

Sumber: Perancangan.

5.3.5 Analisis Hasil Pengujian

Berdasarkan hasil pengujian diperoleh kesimpulan bahwa besarnya kuat sinyal yang dipancarkan / *strength meter* dan besarnya beda fasa sangat tergantung dari jauh atau dekatnya suatu proses transmisi terjadi, artinya semakin jauh jarak antara pemancar dan penerima maka kuat sinyal / *strength meter* dan beda fasa yang dihasilkan kualitasnya akan semakin kurang baik, begitu juga sebaliknya. Disamping itu pengukuran kualitas informasi didasarkan kepada kualitas suara / *audio* yang diterima.

5.4 Pengujian Berdasarkan Tinggi Antena Pemancar

5.4.1 Tujuan

Untuk mendeteksi kuat sinyal yang dipancarkan / *strength meter* dan besarnya beda fasa yang diterima oleh penerima, berdasarkan tinggi antena pemancar.

5.4.2 Peralatan yang Digunakan

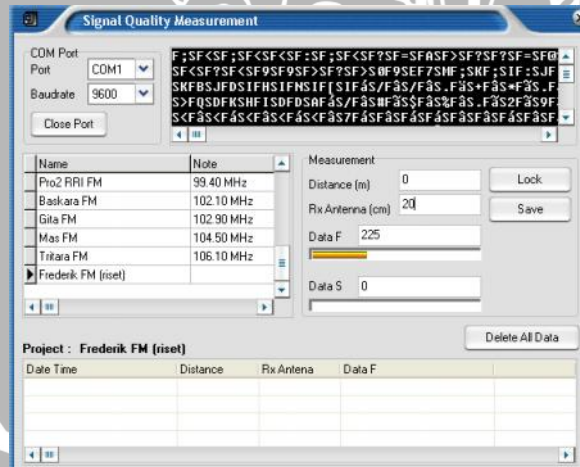
1. Rangkaian sistem alat.
2. Pemancar FM mini 10 mW.
3. Komputer sistem sebagai media tampilan secara *visual* dari hasil pengujian dengan menggunakan *Borland Delphi 7..*
4. Kabel data *serial*, yang berfungsi sebagai media untuk *transfer* komunikasi antara alat dengan sistem komputer.

5.4.3 Prosedur Pengujian

1. Nyalakan alat yang sudah dirancang, dan perhatikan blok diagram pengujian seperti pada Gambar 5.7.
2. Tentukan saklar pada alat di posisi *on*, untuk memastikan bahwa alat melakukan pengukuran dengan pemancar FM mini 10 mW.
3. Tekan tombol *reset* pada alat agar sistem mikrokontroler pada alat siap untuk melakukan perhitungan yang baru.
4. Pada penerima pilih frekuensi radio di 89,05 MHz.
5. Tentukan tinggi antena pemancar pada pemancar FM mini 10 mW dan sesuai dengan kebutuhan pada saat proses pengukuran.
6. Hasil pengukuran tampil di 7 *segment* dan hasil pengukuran secara *visual* tampil di komputer sistem.

5.4.4 Hasil Pengujian

Pengujian dilakukan di ruang tertutup dengan kondisi cuaca yang baik. pengujian sendiri dilakukan pada tanggal 8 Desember 2008, malam hari pukul 19.00 WIB. Dalam hasil pengujian ini juga diberikan data dalam bentuk tabel, untuk lebih jelasnya perhatikan Tabel 5.3.



Gambar 5.10 Tampilan pada *Borland Delphi 7*.

Sumber: Perancangan.



Gambar 5.11 Pengujian pada Alat dengan Tinggi Antena Pemancar 65 cm.
Sumber: Perancangan.

Tabel 5.3 Tabel Perbandingan Posisi Kabel.

No.	Tinggi Antena Pemancar (cm)	Level Tegangan Kuat Pancar Sinyal / <i>Strength meter</i> (mV)	Beda Fasa ($^{\circ}$)	Kualitas Suara
1	20	127	2	Kurang Baik
2	35	225	0	Baik
3	50	227	0	Baik
4	65	226	0	Baik
5	80	143	3	Kurang Baik
6	95	187	1	Kurang Baik
7	110	225	0	Baik

Sumber: Perancangan.

5.4.5 Analisis Hasil Pengujian

Dari hasil pengujian berdasarkan tinggi antena pemancar diperoleh bahwa kuat sinyal dan beda fasa berada pada nilai yang baik saat posisi ketinggian antena pemancar berada di antara 50 cm – 65 cm, hal ini sesuai dengan teori antena bahwa kondisi maksimal daya pancar suatu antena berada pada nilai $0,5 \lambda$. Disamping itu pengukuran kualitas informasi didasarkan kepada kualitas suara / *audio* yang diterima.

5.5 Pengujian Pengukuran pada Pemancar Radio FM di Malang

5.5.1 Tujuan

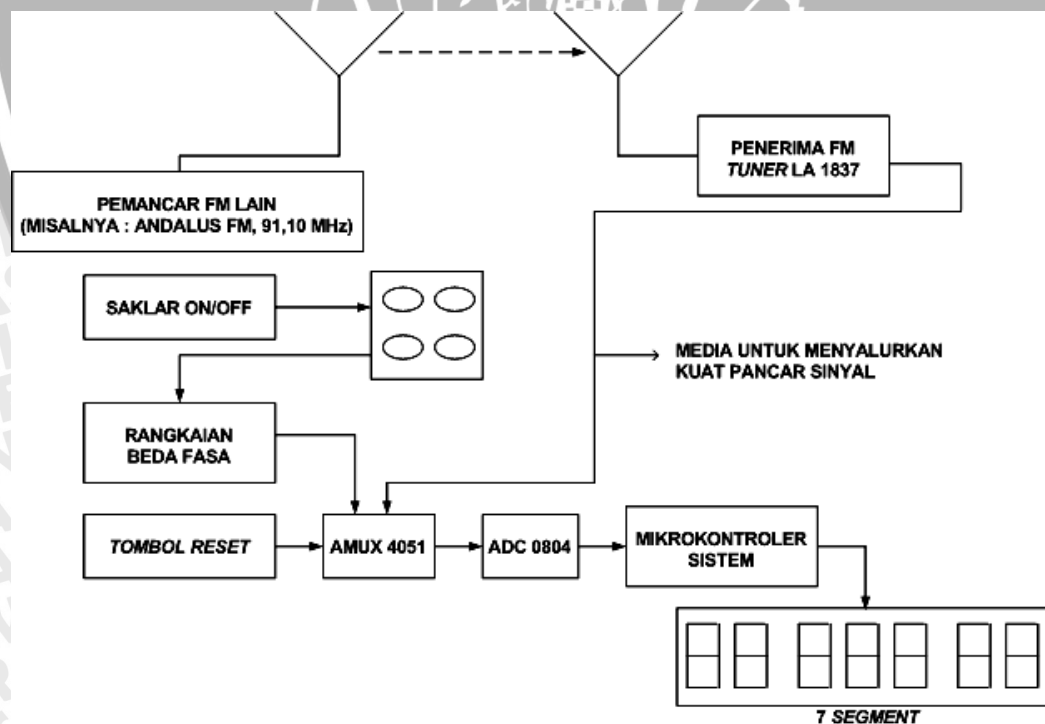
Untuk mendeteksi kuat sinyal yang dipancarkan / *strength meter* oleh setiap pemancar stasiun radio FM yang terdapat di Malang dan sekitarnya. Pada kasus ini tidak diukur besarnya beda fasa karena untuk pemancar lain tidak ada sinkronisasi dengan alat yang dirancang.

5.5.2 Peralatan yang Digunakan

1. Rangkaian sistem alat tanpa pemancar mini 10 mW, karena pengukuran dilakukan ke pemancar FM lain.
2. Komputer sistem sebagai media tampilan secara *visual* dari hasil pengujian dengan menggunakan *Borland Delphi 7..*
3. Kabel data *serial*, yang berfungsi sebagai media untuk *transfer* komunikasi antara alat dengan sistem komputer.

5.5.3 Prosedur Pengujian

1. Nyalakan alat yang sudah dirancang.



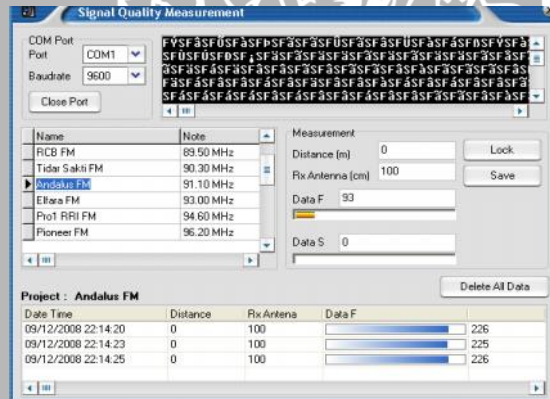
Gambar 5.12 Blok Diagram Pengujian Pengukuran Kuat Sinyal dan Beda Fasa.

Sumber: Perancangan.

2. Tekan tombol *reset* pada alat agar sistem mikrokontroler pada alat siap untuk melakukan perhitungan yang baru.
3. Pilih frekuensi radio FM yang akan diukur, misalnya pilih frekuensi 91,10 MHz untuk Andalus FM.
4. Hasil pengukuran tampil di 7 *segment* dan hasil pengukuran secara *visual* tampil di komputer sistem.

5.5.4 Hasil Pengujian

Pada hasil pengujian akan ditampilkan besarnya kuat sinyal yang dipancarkan / *strength meter* oleh 12 pemancar stasiun radio FM yang terdapat di Malang dan sekitarnya. Pengujian dilakukan di ruang tertutup dengan kondisi cuaca yang baik. pengujian sendiri dilakukan pada tanggal 8 Desember 2008, malam hari pukul 19.00 WIB. Dalam hasil pengujian ini juga diberikan data dalam bentuk tabel, untuk lebih jelasnya perhatikan Tabel 5.4.



Gambar 5.13 Besar Kuat Sinyal Andalus FM pada Tampilan *Borland Delphi 7*.
Sumber: Perancangan.



Gambar 5.14 Besar Kuat Sinyal Andalus FM pada Alat.
Sumber: Perancangan.

Tabel 5.4 Tabel Perbandingan Kuat Sinyal Pemancar Radio di Malang.

No.	Nama	Frekuensi (MHz)	Level Tegangan Kuat Pancar Sinyal / <i>Strength Meter</i> (mV)	Kualitas Suara
1	RCB FM	89,50	229	Baik
2	Tidar Sakti FM	90,30	221	Baik
3	Andalus FM	91,10	228	Baik
4	Elfara FM	93,00	229	Baik
5	RRI PRO 1	94,60	228	Baik
6	Pioneer FM	96,20	226	Baik
7	Kencana FM	98,60	228	Baik
8	RRI PRO 2	99,40	95	Tidak Baik
9	Baskara FM	102,10	226	Baik
10	Gita FM	102,90	225	Baik
11	Mas FM	104,50	130	Kurang Baik
12	Tritara FM	106,10	122	Kurang Baik

Sumber: Perancangan.

5.5.5 Analisis Hasil Pengujian

Dari hasil pengujian diperoleh bahwa besarnya sinyal yang dipancarkan berbanding lurus dengan kualitas suara / *audio*, semakin besar kuat sinyal yang dipancarkan maka semakin jelas dan baik juga kualitas suara / *audio* yang diterima, disamping hal tersebut besarnya kuat sinyal yang dipancarkan juga dida sarkan jarak antara pemancar dengan penerima, serta keadaan cuaca, dan ada atau tidaknya gangguan pada sistem pemancar.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari perancangan dan pembuatan sistem serta dilanjutkan dengan proses pengujian, maka diperoleh kesimpulan sebagai berikut :

1. Perancangan sistem deteksi sinyal suara (*audio*) pada FM dengan menggunakan mikrokontroler AT89C51 dapat memberikan kemudahan dalam proses pendeteksian kuat sinyal yang dipancarkan / *strength meter* serta beda fasa.
2. Sistem pendeteksian kuat sinyal yang dipancarkan / *strength meter* serta beda fasa hanya dilakukan pada modulasi frekuensi (FM).
3. Pendeteksian sinyal suara (*audio*) yang diterima berdasarkan kualitas suara / *audio* yang diterima oleh penerima.
4. Pengukuran kuat sinyal yang dipancarkan / *strength meter* merupakan besarnya kuat sinyal yang diterima oleh penerima, sedangkan pengukuran beda fasa berdasarkan selisih antara sinyal yang dipancarkan dan sinyal yang diterima.
5. Proses pengujian pada radio lain yang berada di Malang dan sekitarnya hanya terbatas untuk mengukur kuat sinyal yang dipancarkan / *strength meter*.

6.2 Saran

Saran-saran untuk penyempurnaan kinerja alat dan pengembangan lebih lanjut adalah sebagai berikut :

1. Sistem pemancar FM yang digunakan hanya terbatas pada pemancar mini 10 mW, sehingga perlu dikembangkan pada sistem pemancar FM yang memiliki daya lebih besar.
2. Perlu diadakan kerja sama dengan pihak perusahaan radio, sehingga sistem ini dapat digunakan untuk menunjang kinerja perusahaan dalam menjaga kualitas suara (*audio*) yang akan disampaikan pada penerima.
3. Pengembangan sistem pendeteksi sinyal suara (*audio*), sehingga sistem yang dirancang tidak hanya terbatas pada modulasi frekuensi (FM).

Daftar Pustaka

Anonymous, Owner manual, " FM-500 M/E Broadcast Transmitter, LYS Electronic LTSA", Rio De Janeiro, Brasilia 1997.

Atmel 2003. AT 89C51. <http://www.atmel.com/dyn/resources/doc0310.pdf>, diakses 21 Mei 2008.

D.Chattopadhyay. 1989. Dasar Elektronika. Calcutta University.

Robert F. Coughlin, Frederick F. Driscoll. 1992. Penguat Operasional dan Rangkaian Terpadu Linear.

Soclof. 1971. *Design and Applications of Analog Integrated Circuits*. Sidney: Prentice Hall.

M. Morris Mano, Charles R. Kime. 2008. *Logic and Computer Design Fundamentals*. Sidney: Prentice Hall.

Jacob Millman. 1986. Mikroelektronika, Sistem Digital dan Rangkaian Analog 1.

Inge Martina. 2002. *Database Menggunakan Delphi*. Yogyakarta: ANDI Yogyakarta.

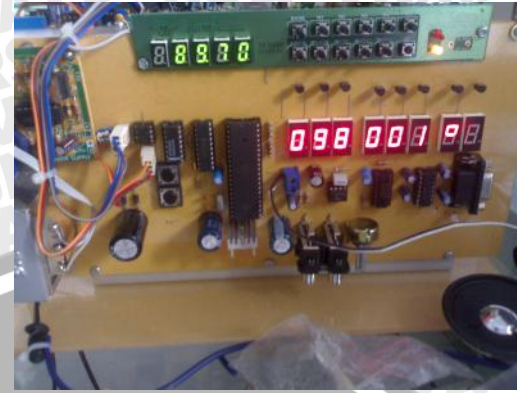
Serial Port pada PC. [http://www.elexmedia.co.id/konfigurasi/serial port/RS -232](http://www.elexmedia.co.id/konfigurasi/serial%20port/RS-232), diakses 3 Juni 2008.



LAMPIRAN A FOTO ALAT



Alat Pada Proses Pengukuran



Alat Tampak Depan



Alat Tampak Atas



Pemancar FM Mini 10 mW



LAMPIRAN B
GAMBAR SKEMATIK
ALAT



LAMPIRAN C
LISTING PROGRAM
MIKROKONTROLER



```
DIGIT1 EQU 21H
DIGIT2 EQU 22H
DIGIT3 EQU 23H
DIGIT4 EQU 24H
DIGIT5 EQU 21H
DIGIT6 EQU 22H
DIGIT7 EQU 23H
COUNTER EQU 26H

SOC BIT P3.3
EOC BIT P3.4
FSM BIT P3.5
FASA BIT P3.6
```

```
ORG 0H
AJMP MAIN

ORG 001BH
AJMP TIMER_1
```

```
MAIN: SETB EA
      SETB ETO
      SETB ET1
      SETB P3.7
```

```
CLR RS0
CLR RS1

MOV SP, #50H
```

```
INIT_COM1: MOV SCON, #52H
          MOV TMOD, #20H
          MOV TH1, #0FDH
          MOV TCON, #40H
          MOV 0C8H, #00H
          ORL 87H, #80H
```

```
START: ACALL READ_FSM
        ACALL HEX2BCD
        ACALL SEND2COM
```

```
X1: ACALL REFRESH_DISP
     JB READY, X1
```

```
ACALL READ_FASA
ACALL HEX2BCD
ACALL SEND2COM
X2: ACALL REFRESH_DISP
     JB READY, X2
```

```
;
```

```
READ_FSM: SETB FSM
          SETB SOC
```

```
X3: ACALL REFRESH_DISP
     JB EOC, X3
     MOV A, P1
     MOV INPUT, A
     RET
```

```
;
```




```

READ_FASA:      SETB FASA
                SETB SOC
X4:             ACALL REFRESH_DISP
                JB EOC,X4
                MOV A,P1
                MOV INPUT,A
                RET
;-----
NUM_CONV:      PUSH DPH
                PUSH DPL
                MOV DPTR,#NUMERIK
NEXTDATA:      MOVC A,@A+DPTR
                MOV P1,A
                POP DPL
                POP DPH
                RET
;-----
SEND2COM1:     MOV DPTR,#DATA
KIRIM:         CLR A
                MOVC A,@A+DPTR
                MOV SBUF,A
                JNB TI,$
                CLR TI
                INC DPTR
                ACALL DELAY_5S
                DJNZ R2,KIRIM
                RET
;-----
RUN_TIMER:     PUSH DPH
                PUSH DPL
                MOV TL1,#17H ; Set Sampling Time 1000uS(FC17)
                MOV TH1,#0FCH
                MOV TL0,#00H ; Preset Counter
                MOV TH0,#00H
                MOV TMOD,#00010101B ; T1 = Timer T0 = Counter
                ORL TCON,#01010000B
AGAIN1:        JBC F0,AGAIN2
                SJMP AGAIN1
AGAIN2:        ANL TCON,#10101111B
UP_DATE:       MOV A,TL0
                MOV INPUT,A
EXIT2:         POP DPL
                POP DPH
                RET
;-----
DELAY_5S:      PUSH DPH
                PUSH DPL

```

```

DEL2:      MOV R7,#0FH
DEL3:      MOV A,#0FFH
           MOV B,#0FFH
           DJNZ B,$
           DJNZ ACC,DEL3
           DJNZ R7,DEL2
    
```

```

POP DPL
POP DPH
RET
    
```

```

-----
DELAY_2S:  PUSH DPH
           PUSH DPL
    
```

```

           MOV R7,#0FH
DEL4:      MOV A,#0FH
DEL5:      MOV B,#0FFH
           DJNZ B,$
           DJNZ ACC,DEL5
           DJNZ R7,DEL4
    
```

```

POP DPL
POP DPH
RET
    
```

```

-----
DELAY:     PUSH DPH
           PUSH DPL
    
```

```

           MOV R7,#2
DEL6:      MOV A,#5
DEL7:      MOV B,#255
           DJNZ B,$
           DJNZ ACC,DEL7
           DJNZ R7,DEL6
    
```

```

POP DPL
POP DPH
RET
    
```

```

-----
REFRESH_DISP:  PUSH DPH
               PUSH DPL
    
```

```

           MOV A,DIGIT1      ;DIGIT1 IS LSD
           ACALL NUM_CONV
           SETB P3.2
           CLR P3.1
           CLR P3.0
           ACALL DELAY
    
```

```

           MOV A,DIGIT2
           ACALL NUM_CONV
           SETB P3.1
           CLR P3.0
           CLR P3.2
           ACALL DELAY
    
```

```

           MOV A,DIGIT3
    
```





```
ACALL NUM_CONV
SETB P3.0
CLR P3.1
CLR P3.2
ACALL DELAY
```

```
MOV A,DIGIT4
ACALL NUM_CONV
SETB P3.2
CLR P3.1
SETB P3.0
ACALL DELAY
```

```
MOV A,DIGIT5
ACALL NUM_CONV
SETB P3.1
SETB P3.0
CLR P3.2
ACALL DELAY
```

```
MOV A,DIGIT6
ACALL NUM_CONV
SETB P3.0
CLR P3.1
CLR P3.2
ACALL DELAY
```

```
POP DPL
POP DPH
RET
```

HEX2BCD:

```
PUSH DPH
PUSH DPL
```

```
MOV R2,#9
ACALL CONV1
MOV A,B
ANL A,#0FH
MOV DIGIT1,A
MOV A,B
ANL A,#0F0H
SWAP A
MOV DIGIT2,A
MOV R2,#9
MOV A,B
ANL A,#0FH
MOV DIGIT3,A
```

```
POP DPL
POP DPH
RET
```

CONV1:

```
PUSH DPH
PUSH DPL
```

```
CLR C
CLR A
MOV B,A
```

CONV2:

```
DJNZ R2,CONT
```

```

CONT:          SJMP EXIT
              XCH A,INPUT
              RLC A
              XCH A,INPUT
              MOV A,B
              ADDC A,ACC
              DA A
              MOV B,A
              JNC CONV2
              XCH A,INPUT
              ADDC A,#0H
              XCH A,INPUT
              SJMP CONV2
    
```

```

EXIT:         POP DPL
              POP DPH
              RET
;
-----
    
```

```

TIMER_1:     PUSH DPH
              PUSH DPL

              SETB F0

              POP DPL
              POP DPH
              RETI
;
-----
    
```

```

ISI_BUFFER:  PUSH DPH
              PUSH DPL
              MOV 7CH,#05H
              MOV R0,#70H
    
```

```

ULANG:       MOV A,7EH
              MOVC A,@A+DPTR
              MOV @R0,A
              INC DPTR
              INC R0
              DJNZ 7CH,ULANG
              RET
    
```

```

DIS8SEG:     MOV 7BH,#5
              MOV R1,#70H
              MOV 78H,#07H
    
```

```

ULANG2:      MOV A,@R1
              MOV P1,A
              INC R1
              MOV A,78H
              POP DPL
              POP DPH
              RET
;
-----
    
```

```

NUMERIK:     DB 3FH,30H,5BH,4FH,66H,6DH,7CH,07H,7FH,6FH
              END
    
```



**LAMPIRAN D
LISTING PROGRAM
BORLAND DELPHI 7**



unit fUtama;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, CPortCtl, CPort, IniFiles, DB, DBTables, ADODB, Grids,
DBGrids, NxColumnClasses, NxColumns, NxScrollControl,
NxCustomGridControl, NxCustomGrid, NxGrid, WinSkinData, ComCtrls, ImgList ;

type

```
TfrmUtama = class(TForm)
  ComPort1: TComPort;
  ComDataPacket1: TComDataPacket;
  GroupBox1: TGroupBox;
  ComComboBox1: TComComboBox;
  ComComboBox2: TComComboBox;
  Label1: TLabel;
  Label2: TLabel;
  ADOConnection1: TADOConnection;
  tblProject: TADOTable;
  DBGrid1: TDBGrid;
  DataSource1: TDataSource;
  DataSource2: TDataSource;
  GroupBox2: TGroupBox;
  Grid2: TNextGrid;
  colWaktu: TNxTextColumn;
  colJarak: TNxNumberColumn;
  colDataF: TNxNumberColumn;
  colProgF: TNxProgressColumn;
  colDataS: TNxNumberColumn;
  colProgS: TNxProgressColumn;
  lblProject: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  edtF: TEdit;
  edtS: TEdit;
  btnScan: TButton;
  btnSave: TButton;
  SkinData1: TSkinData;
  tblData: TADOTable;
  ProgressBar1: TProgressBar;
  ProgressBar2: TProgressBar;
  ComTerminal1: TComTerminal;
  edtJarak: TEdit;
  Label5: TLabel;
  Label6: TLabel;
  btnPort: TButton;
  Button1: TButton;
  NxImageColumn1: TNxImageColumn;
  ImageList1: TImageList;
  ADOQuery1: TADOQuery;
  Label7: TLabel;
  edtHeight: TEdit;
  colHeight: TNxNumberColumn;
  procedure SaveData(ProjectID:String;Jarak,Tinggi,DataF,DataS:Double;Waktu:TDateTime);
  procedure ShowDataProject(ProjectID : String );
  procedure FormCreate(Sender : TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure btnScanClick(Sender: TObject);
```




```

procedure ComDataPacket1Packet(Sender: TObject; const Str: String);
procedure edtFChange(Sender: TObject);
procedure edtSChange(Sender: TObject);
procedure DBGrid1DbClick(Sender: TObject);
procedure btnSaveClick(Sender: TObject);
procedure btnPortClick(Sender: TObject);
procedure Grid2CellClick(Sender: TObject; ACol, ARow: Integer);
procedure Button1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frmUtama: TfrmUtama;
  IniMeasure : TIniFile ;
  IsScan      : Boolean;
  IsAverage   : Boolean;
  IsAutoSave  : Boolean;
  TempF,TempS : Integer;
  MaxF        : Integer;
  MaxS        : Integer;
  StrData     : String;
implementation

{$R *.dfm}
procedure
TfrmUtama.SaveData(ProjectID:String;Jarak,Tinggi,DataF,DataS:Double;Waktu:TDateTime);
var
  i:Integer;
begin
  tblData.Append;
  tblData['project_id']:=ProjectID;
  tblData['waktu']:=FormatDateTime('dd/mm/yyyy hh:nn:ss',waktu);
  tblData['jarak']:=Jarak;
  tblData['tinggi']:=Tinggi;
  tblData['data_f']:=DataF;
  tblData['data_s']:=DataS;
  tblData.Post;
  if lblProject.Caption=ProjectID then begin
    Grid2.AddRow;
    i:=Grid2.RowCount-1;
    Grid2.Cell[0,i].AsString:=FormatDateTime('dd/mm/yyyy hh:nn:ss',waktu);
    Grid2.Cell[1,i].AsFloat:=Jarak;
    Grid2.Cell[2,i].AsFloat:=Tinggi;
    Grid2.Cell[3,i].AsInteger:=round(DataF/MaxF*100);
    Grid2.Cell[4,i].AsFloat:=DataF;
    Grid2.Cell[5,i].AsInteger:=round(DataS/180);
    Grid2.Cell[6,i].AsFloat:=DataS;
  end;
end;

procedure TfrmUtama.ShowDataProject(ProjectID:String);
var
  i,j,k,l,m,n:Integer;
  s1,s2,s3,s4,s5:String;
  d1,d2:Double;
begin
  tblData.Filtered:=False;

```

```

tblData.Filter:='project_id='+QuotedStr(tblProject['id']);
tblData.Filtered:=True;
tblData.First;
Grid2.ClearRows;
i:=0;
while not tblData.EOF do begin
  Grid2.AddRow;
  Grid2.Cell[0,i].AsString:=tblData['waktu'];
  Grid2.Cell[1,i].AsFloat:=tblData['jarak'];
  Grid2.Cell[2,i].AsFloat:=tblData['tinggi'];
  Grid2.Cell[3,i].AsInteger:=round(tblData['data_f']/MaxF*100);
  Grid2.Cell[4,i].AsFloat:=tblData['data_f'];
  Grid2.Cell[5,i].AsInteger:=round(tblData['data_s']/180*100);
  Grid2.Cell[6,i].AsFloat:=tblData['data_s'];
  tblData.Next;
  i:=i+1;
end;
end;

procedure TfrmUtama.FormCreate(Sender: TObject);
var
  i,j,k:Integer ;
begin
  MaxF:=255;
  MaxS:=255;
  IsScan:=True;
  ADOConnection1.ConnectionString:='Provider=Microsoft.Jet.OLEDB.4.0;Data Source='+
  ExtractFilePath(Application.ExeName)+'dbpengukuran.mdb;Persist Security Info=False';
  ADOConnection1.Open;
  tblProject.Open;
  tblData.Open;
  DBGrid1.Columns.Items[0].Width:=150;
  DBGrid1.Columns.Items[1].Width:=200;
  DBGrid1.Columns.Items[0].Title.caption:='Name';
  DBGrid1.Columns.Items[1].Title.caption:='Note';
  //for i:=1 to 10 do begin
  // Grid2.AddRow;
  // Grid2.Cell[0,i-1].AsString:=IntToStr(i);
  // Grid2.Cell[2,i-1].AsInteger:=i*10;
  //end;
end;

procedure TfrmUtama.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  tblProject.Close;
  tblData.Open;
  ADOConnection1.Close;
end;

procedure TfrmUtama.btnScanClick(Sender: TObject);
begin
  if btnScan.Caption='Scaning' then begin
    IsScan:=True;
    btnScan.Caption:='Lock';
  end else begin
    IsScan:=False;
    btnScan.Caption:='Scaning';
  end;
end;
end;

```



```

procedure TfrmUtama.ComDataPacket1Packet(Sender: TObject;
const Str: String);
var
i,j,k,l,m,n : Integer;
d1,d2,d3 : Integer ;
s1,s2,s3,s4,s5 : String;
IsValid : Boolean;
label
STR_OK;
begin
StrData:=StrData + Str;
StrData := Trim(StrData);
if length(StrData)>=0 then begin
j:=0;
k:=0;
for i:=length(StrData)-1 downto 1 do begin
if (StrData[i]='S') then begin
j:=i;
end;
if (StrData[i]='F') then begin
k:=i;
end;
if (j*k<>0) then begin//and(Abs(j-k)=2))then begin
if strdata[j+1]<>'F' then d1:=ord(strdata[j+1])else d1:=0;
if strdata[k+1]<>'S' then d2:=ord(strdata[k+1])else d2:=0;
IsValid:=True;
StrData:="";
goto STR_OK;
end;
end;

end;

if not isvalid then exit;
STR_OK:
if IsScan then begin
edtS.Text:=intToStr(d1);
edtF.Text:=intToStr(d2);
end;
end;

procedure TfrmUtama.edtFChange(Sender: TObject);
var
d1 :Double;
begin
if TryStrToFloat(edtF.Text,d1)then ProgressBar1.Position:=round(d1/MaxF*100);
end;

procedure TfrmUtama.edtSChange(Sender: TObject);
var
d1 :Double;
begin
if TryStrToFloat(edtS.Text,d1)then ProgressBar2.Position:=round(d1/MaxS*100);
end;
procedure TfrmUtama.DBGrid1DbClick(Sender: TObject);
begin
lblProject.Caption:=tblProject['id'];
ShowDataProject(tblProject['id']);
end;

```



```

procedure TfrmUtama.btnSaveClick(Sender: TObject);
var
  i,j,k,l,m,n : Integer;
  s1,s2,s3,s4,s5 : String;
  d1,d2,d3,d4,d5 : Double;
begin
  if TryStrToFloat(edtF.Text,d1) and
    TryStrToFloat(edtS.Text,d2)and
    TryStrToFloat(edtJarak.Text,d3) and
    TryStrToFloat(edtHeight.Text,d4)
  then begin
    SaveData(tblProject['id'],d3,d4,d1,d2,Now);
  end;
end;

procedure TfrmUtama.btnPortClick(Sender: TObject);
begin
  if btnPort.Caption='Open Port' then begin
    ComPort1.Open;
    btnPort.Caption:='Close Port';
  end else begin
    ComPort1.Close;
    btnPort.Caption:='Open Port';
  end;
end;

procedure TfrmUtama.Grid2CellClick(Sender: TObject; ACol, ARow: Integer);
var
  i,j,k,l,m,n : Integer ;
  s1,s2,s3,s4,s5 : String ;
  d1,d2,d3 : Double;
  tg1,tg2 : TDateTime;
begin
  if ACol=7 then begin
    s1:=Grid2.Cell[0,ARow].AsString;
    ADOQuery1.SQL.Text:='delete from tdata where project_id="'+lblProject.Caption+"' and
    waktu='"+s1+"'";
    ADOQuery1.ExecSQL;
    tblData.Close;
    tblData.open;
    ShowDataProject(lblProject.Caption);
  end;
end;

procedure TfrmUtama.Button1Click(Sender: TObject);
begin
  ADOQuery1.SQL.Text:='delete from tdata where project_id="'+lblProject.Caption+"'";
  ADOQuery1.ExecSQL;
  tblData.Close;
  tblData.open;
  ShowDataProject(lblProject.Caption);
end;

end.

```




**LAMPIRAN E
DATA SHEET
MIKROKONTROLER
AT89C51**



8-Bit Microcontroller with 4K Bytes Flash


AT89C51

Features

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
 - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4Kbytes of Flash Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The ATMEL logo is displayed in a bold, black, sans-serif font. The letters 'A', 'M', and 'E' are significantly larger than the 'T' and 'L'. The logo is positioned on the right side of the page, partially overlapping a vertical black line that runs down the page.

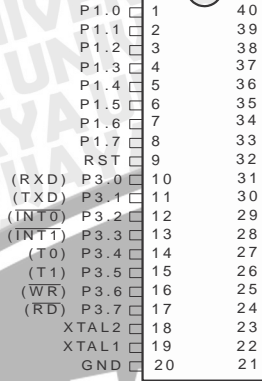
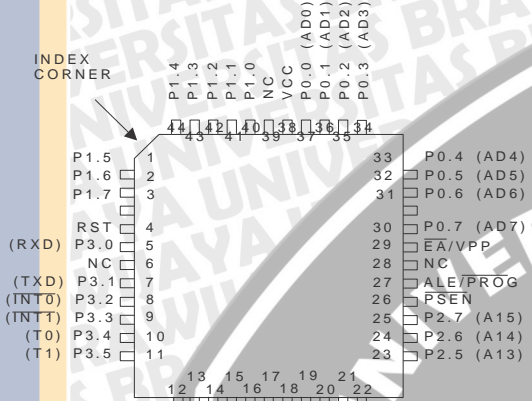
Pin Configurations

(continued)

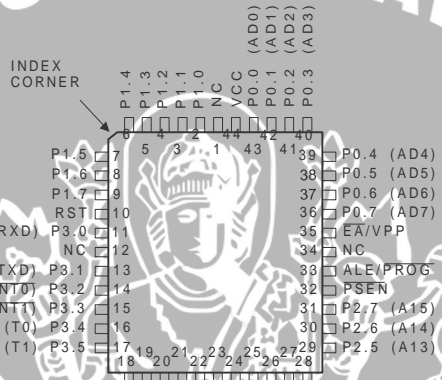
PQFP/TQFP

PDIP

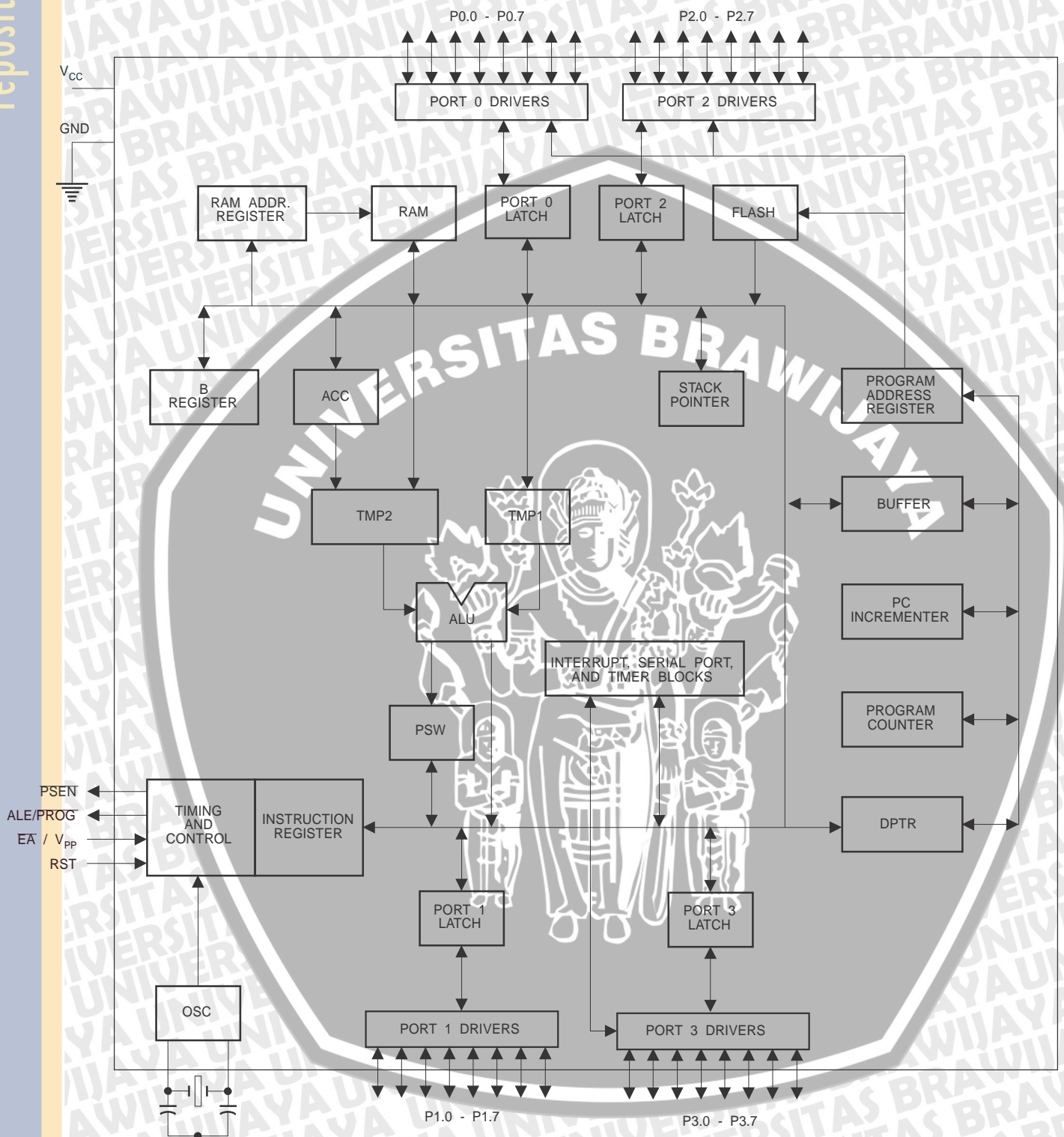
INDEX CORNER



PLCC



Block Diagram



The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Description

V_{CC}

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application it uses strong internal pullups

when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/PROG

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

PSEN

Program Store Enable is the read strobe to external program memory.

When the AT89C51 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.

EA/V_{PP}

External Access Enable. $\overline{\text{EA}}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{\text{EA}}$ will be internally latched on reset.

$\overline{\text{EA}}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming, for parts that require 12-volt V_{PP}.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

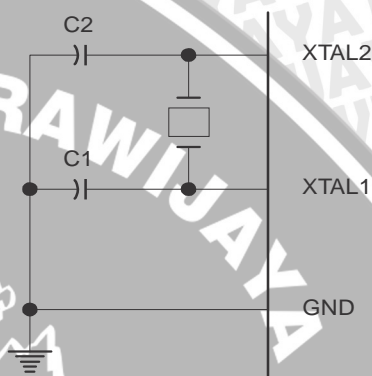
In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

Status of External Pins During Idle and Power Down Modes

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

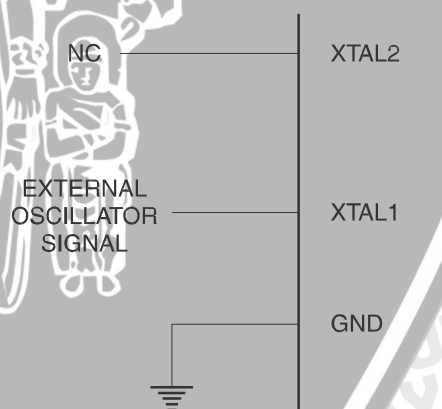
It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Figure 1. Oscillator Connections



Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



Power Down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Lock Bit Protection Modes

Program Lock Bits			Protection Type	
LB1	LB2	LB3		
1	U	U	U	No program lock features.
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, \overline{EA} is sampled and latched on reset, and further programming of the Flash is disabled.
3	P	P	U	Same as mode 2, also verify is disabled.
4	P	P	P	Same as mode 3, also external execution is disabled.

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	$V_{PP} = 12V$	$V_{PP} = 5V$
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H)=1EH (031H)=51H (032H)=FFH	(030H)=1EH (031H)=51H (032H)=05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

When lock bit 1 is programmed, the logic level at the \overline{EA} pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of \overline{EA} be in agreement with the current logic level at that pin in order for the device to function properly.

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figures 3 and 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise \overline{EA}/V_{PP} to 12V for the high-voltage programming mode.
5. Pulse ALE/ \overline{PROG} once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

Data Polling: The AT89C51 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The progress of byte programming can also be monitored by the RDY/ \overline{BSY} output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.



Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/PROG low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H,

031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel

(031H) = 51H indicates 89C51

(032H) = FFH indicates 12V programming

(032H) = 05H indicates 5V programming

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

Mode	RST	PSEN	ALE/PROG	EA/V _{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data	H	L		H/12V	L	H	H	H
Read Code Data	H	L		H	L	L	H	H
Write Lock	Bit - 1	L		H/12V	H	H	H	H
				H/12V	H	H	L	L
				H/12V	H	L	H	L
Chip Erase	H	L	(1)	H/12V	H	L	L	L
Read Signature Byte	H	L		H	L	L	L	L

Note: 1. Chip Erase requires a 10-ms PROG pulse.

Figure 3. Programming the Flash

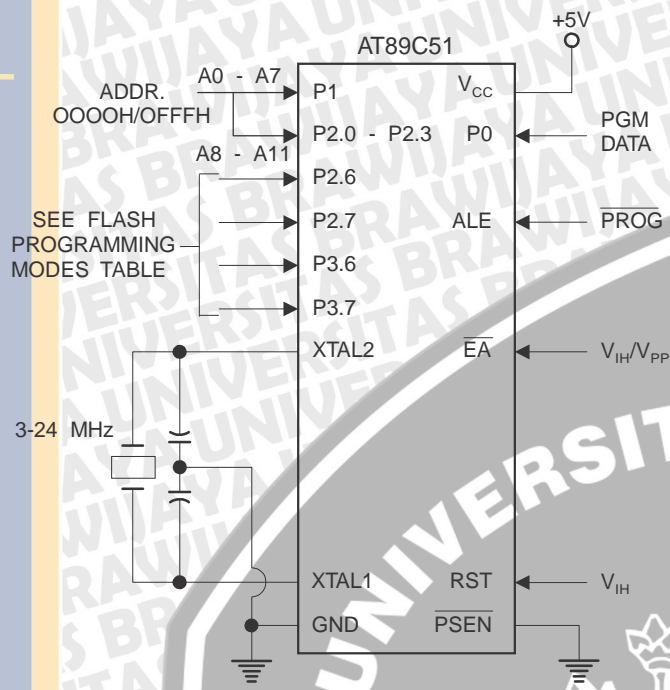
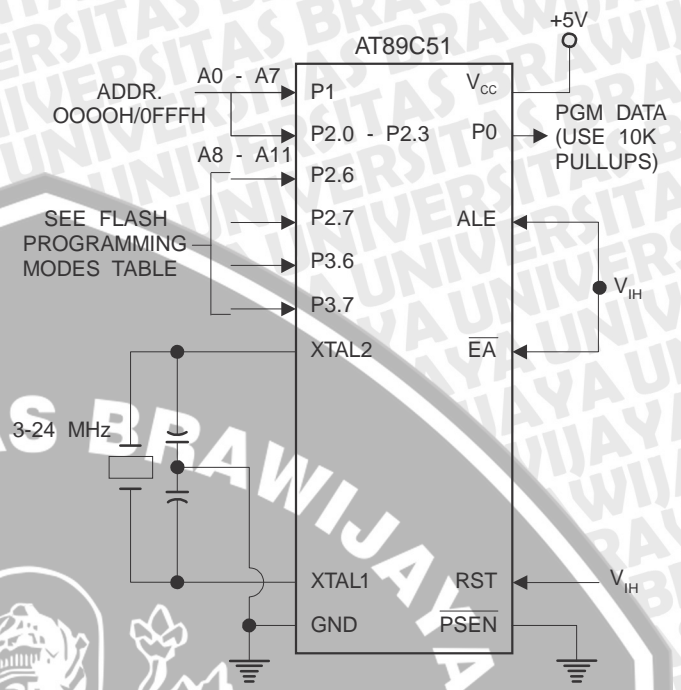


Figure 4. Verifying the Flash



Flash Programming and Verification Characteristics

$T_A = 0^\circ\text{C to } 70^\circ\text{C}$, $V_{CC} = 5.0 \pm 10\%$

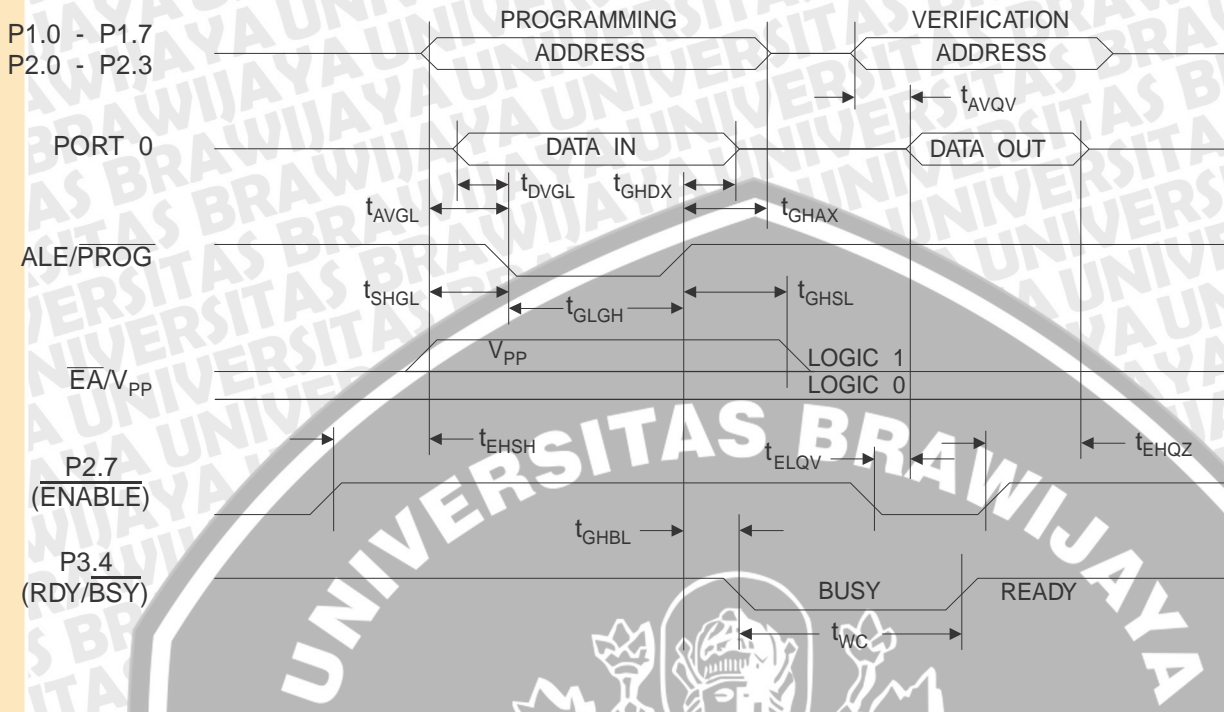
Symbol	Parameter	Min	Max	Units
$V_{PP}^{(1)}$	Programming Enable Voltage	11.5	12.5	V
$I_{PP}^{(1)}$	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to PROG Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After PROG	$48t_{CLCL}$		
t_{DVGL}	Data Setup to PROG Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After PROG	$48t_{CLCL}$		
t_{EHSB}	P2.7 (ENABLE) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to PROG Low	10		µs
$t_{GHSL}^{(1)}$	V_{PP} Hold After PROG	10		µs
t_{GLGH}	PROG Width	1	110	µs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	ENABLE Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float After ENABLE	0	$48t_{CLCL}$	
t_{GHBL}	PROG High to BUSY Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

Note: 1. Only used in 12-volt programming mode.

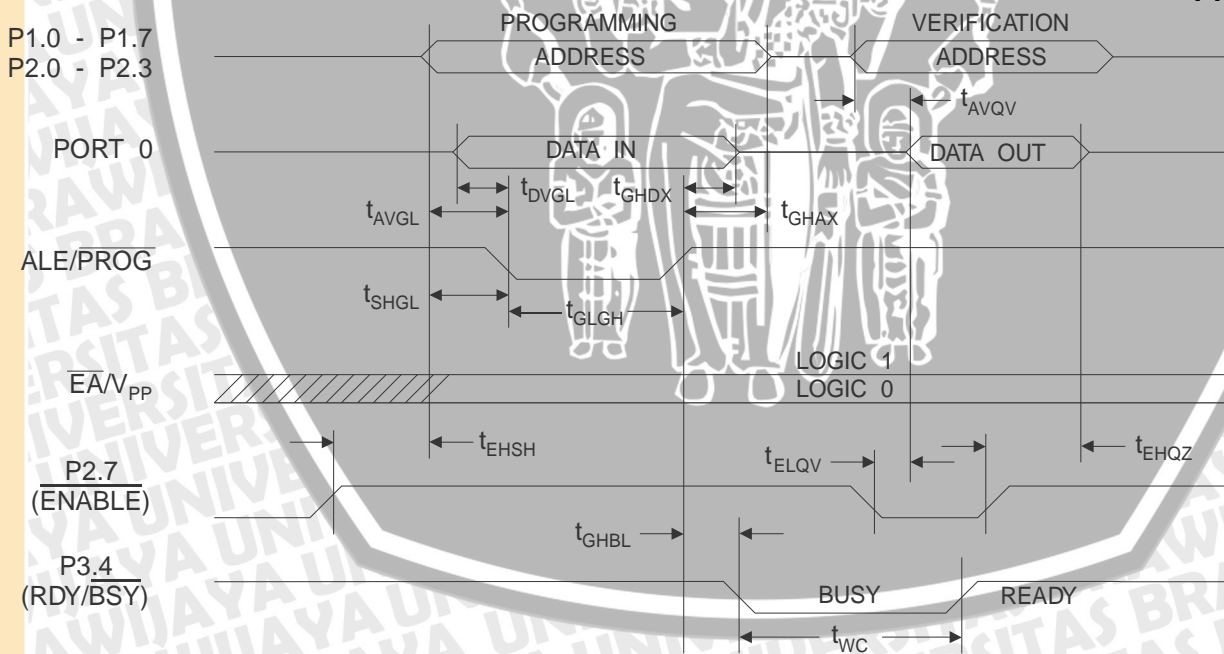




Flash Programming and Verification Waveforms - High Voltage Mode ($V_{PP} = 12V$)



Flash Programming and Verification Waveforms - Low Voltage Mode ($V_{PP} = 5V$)



Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage.....	6.6V
DC Output Current.....	15.0 mA

*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 5.0V ± 20% (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V _{IL}	Input Low Voltage	(Except EA)	-0.5	0.2 V _{CC} - 0.1	V
V _{IL1}	Input Low Voltage (EA)		-0.5	0.2 V _{CC} - 0.3	V
V _{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V _{CC} + 0.9	V _{CC} + 0.5	V
V _{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V _{CC}	V _{CC} + 0.5	V
V _{OL}	Output Low Voltage ⁽¹⁾ (Ports 1,2,3)	I _{OL} = 1.6 mA		0.45	V
V _{OL1}	Output Low Voltage ⁽¹⁾ (Port 0, ALE, PSEN)	I _{OL} = 3.2 mA		0.45	V
V _{OH}	Output High Voltage (Ports 1,2,3, ALE, PSEN)	I _{OH} = -60 μA, V _{CC} = 5V ± 10%	2.4		V
		I _{OH} = -25 μA	0.75 V _{CC}		V
		I _{OH} = -10 μA	0.9 V _{CC}		V
V _{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I _{OH} = -800 μA, V _{CC} = 5V ± 10%	2.4		V
		I _{OH} = -300 μA	0.75 V _{CC}		V
		I _{OH} = -80 μA	0.9 V _{CC}		V
I _{IL}	Logical 0 Input Current (Ports 1,2,3)	V _{IN} = 0.45V		-50	μA
I _{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V _{IN} = 2V, V _{CC} = 5V ± 10%		-650	μA
I _{LI}	Input Leakage Current (Port 0, EA)	0.45 < V _{IN} < V _{CC}		±10	μA
RRST	Reset Pulldown Resistor		50	300	KΩ
C _{IO}	Pin Capacitance	Test Freq. = 1 MHz, T _A = 25°C		10	pF
I _{CC}	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power Down Mode ⁽²⁾	V _{CC} = 6V		100	μA
		V _{CC} = 3V		40	μA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:

Maximum I_{OL} per port pin: 10 mA

Maximum I_{OL} per 8-bit port: Port 0: 26 mA

Ports 1, 2, 3: 15 mA

Maximum total I_{OL} for all output pins: 71 mA

If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum V_{CC} for Power Down is 2V.





AC Characteristics

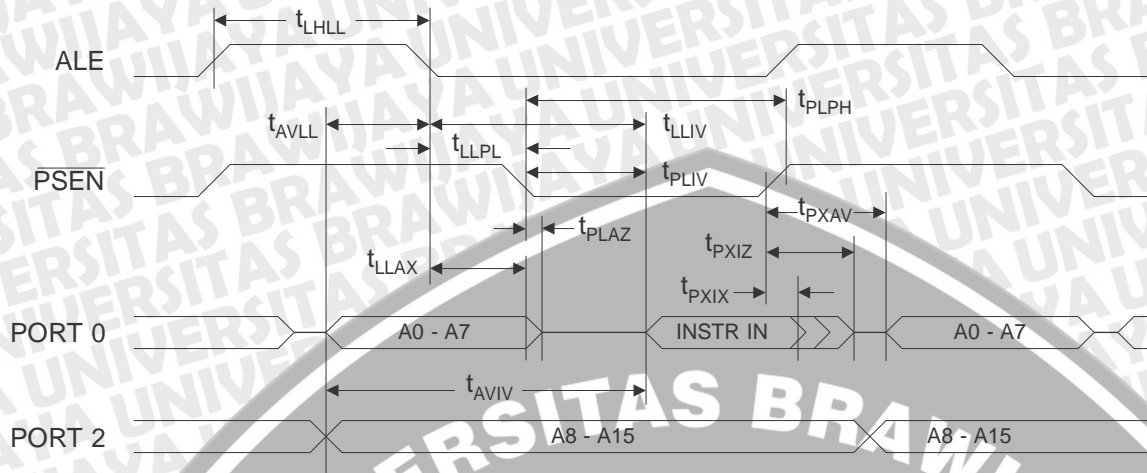
(Under Operating Conditions; Load Capacitance for Port 0, ALE/PROG, and PSEN = 100 pF; Load Capacitance for all other outputs = 80 pF)

External Program and Data Memory Characteristics

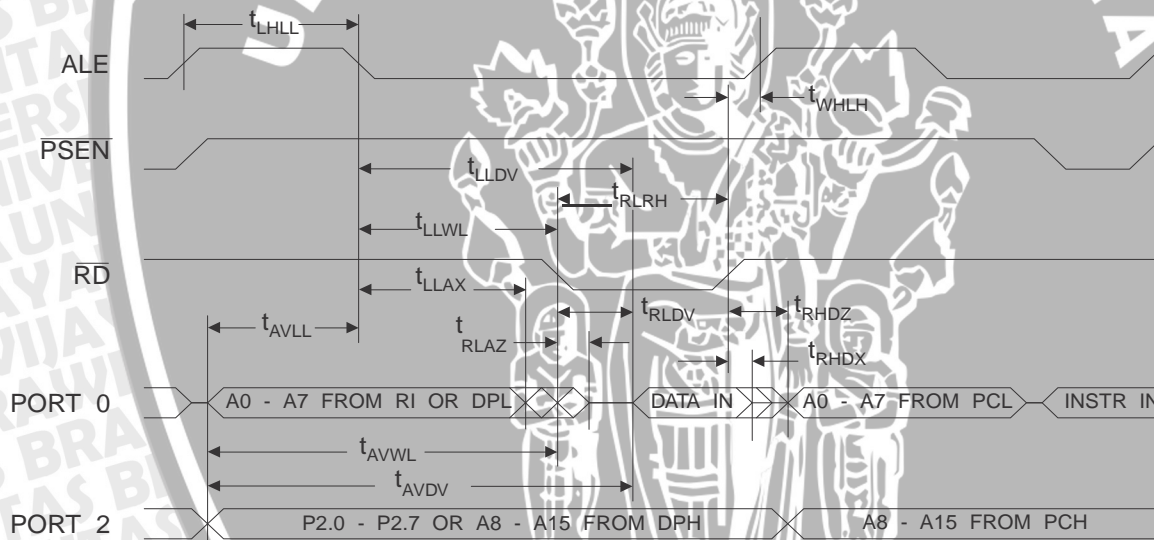
Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}-40$		ns
t_{AVLL}	Address Valid to ALE Low	43		$t_{CLCL}-13$		ns
t_{LLAX}	Address Hold After ALE Low	48		$t_{CLCL}-20$		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}-65$	ns
t_{LLPL}	ALE Low to PSEN Low	43		$t_{CLCL}-13$		ns
t_{PLPH}	PSEN Pulse Width	205		$3t_{CLCL}-20$		ns
t_{PLIV}	PSEN Low to Valid Instruction In		145		$3t_{CLCL}-45$	ns
t_{PXIX}	Input Instruction Hold After PSEN	0		0		ns
t_{PXIZ}	Input Instruction Float After PSEN		59		$t_{CLCL}-10$	ns
t_{PXAV}	PSEN to Address Valid	75		$t_{CLCL}-8$		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}-55$	ns
t_{PLAZ}	PSEN Low to Address Float		10		10	ns
t_{RLRH}	RD Pulse Width	400		$6t_{CLCL}-100$		ns
t_{WLWH}	WR Pulse Width	400		$6t_{CLCL}-100$		ns
t_{RLDV}	RD Low to Valid Data In		252		$5t_{CLCL}-90$	ns
t_{RHDX}	Data Hold After RD	0		0		ns
t_{RHDZ}	Data Float After RD		97		$2t_{CLCL}-28$	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}-150$	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}-165$	ns
t_{LLWL}	ALE Low to RD or WR Low	200	300	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
t_{AVWL}	Address to RD or WR Low	203		$4t_{CLCL}-75$		ns
t_{QVWX}	Data Valid to WR Transition	23		$t_{CLCL}-20$		ns
t_{QVWH}	Data Valid to WR High	433		$7t_{CLCL}-120$		ns
t_{WHQX}	Data Hold After WR	33		$t_{CLCL}-20$		ns
t_{RLAZ}	RD Low to Address Float		0		0	ns
t_{WHLH}	RD or WR High to ALE High	43	123	$t_{CLCL}-20$	$t_{CLCL}+25$	ns



External Program Memory Read Cycle

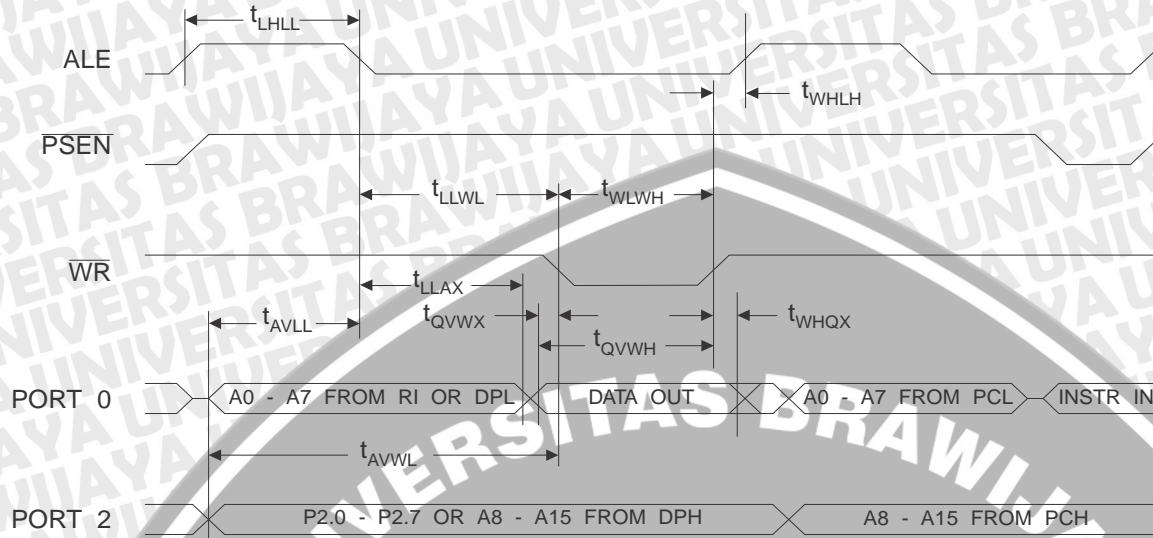


External Data Memory Read Cycle

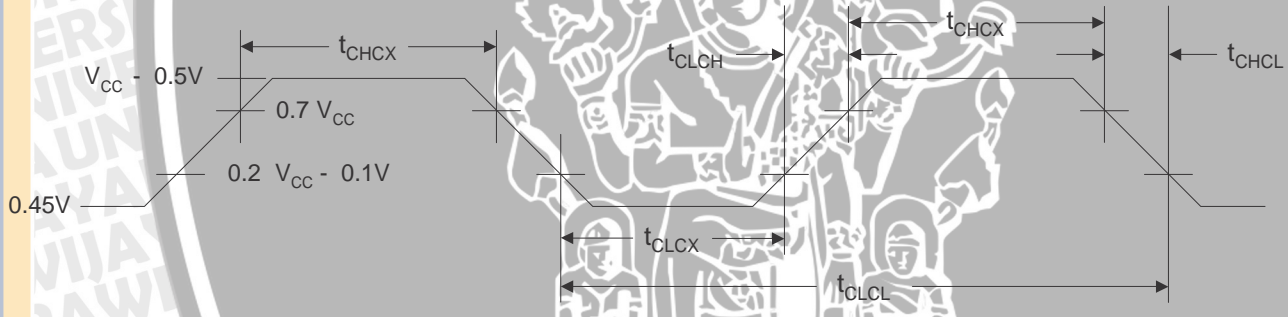




External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

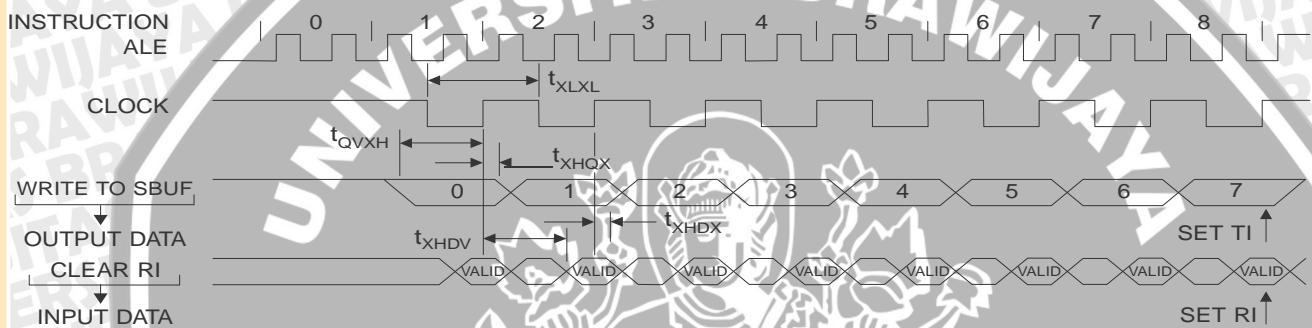
Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

Serial Port Timing: Shift Register Mode Test Conditions

($V_{CC} = 5.0\text{ V} \pm 20\%$; Load Capacitance = 80 pF)

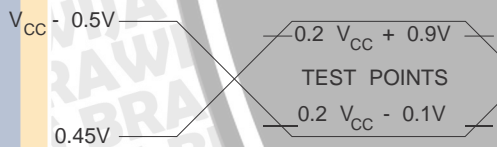
Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		μs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}-133$		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}-117$		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHdV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}-133$	ns

Shift Register Mode Timing Waveforms

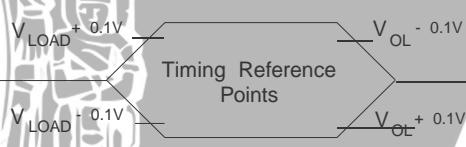


AC Testing Input/Output Waveforms (1)

Float Waveforms (1)



Note: 1. AC Inputs during testing are driven at $V_{CC} - 0.5\text{V}$ for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded V_{OH}/V_{OL} level occurs.



Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range	
12	5V ± 20%	AT89C51-12AC	44A	Commercial (0°C to 70°C)	
		AT89C51-12JC	44J		
		AT89C51-12PC	40P6		
		AT89C51-12QC	44Q		
		5V ± 20%	AT89C51-12AI	44A	Industrial (-40°C to 85°C)
			AT89C51-12JI	44J	
			AT89C51-12PI	40P6	
			AT89C51-12QI	44Q	
		5V ± 20%	AT89C51-12AA	44A	Automotive (-40°C to 105°C)
			AT89C51-12JA	44J	
			AT89C51-12PA	40P6	
			AT89C51-12QA	44Q	
16	5V ± 20%	AT89C51-16AC	44A	Commercial (0°C to 70°C)	
		AT89C51-16JC	44J		
		AT89C51-16PC	40P6		
		AT89C51-16QC	44Q		
		5V ± 20%	AT89C51-16AI	44A	Industrial (-40°C to 85°C)
			AT89C51-16JI	44J	
			AT89C51-16PI	40P6	
			AT89C51-16QI	44Q	
		5V ± 20%	AT89C51-16AA	44A	Automotive (-40°C to 105°C)
			AT89C51-16JA	44J	
			AT89C51-16PA	40P6	
			AT89C51-16QA	44Q	
20	5V ± 20%	AT89C51-20AC	44A	Commercial (0°C to 70°C)	
		AT89C51-20JC	44J		
		AT89C51-20PC	40P6		
		AT89C51-20QC	44Q		
		5V ± 20%	AT89C51-20AI	44A	Industrial (-40°C to 85°C)
			AT89C51-20JI	44J	
			AT89C51-20PI	40P6	
			AT89C51-20QI	44Q	

**AT
89
C5
1**

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Ran
24	5V ± 20%	AT89C51-24AC	44A	Commercial (0°C to 70°C)
		AT89C51-24JC	44J	
		AT89C51-24PC	44P6	
		AT89C51-24QC	44Q	
		AT89C51-24AI	44A	Industrial (-40°C to 85°C)
		AT89C51-24JI	44J	
		AT89C51-24PI	44P6	
		AT89C51-24QI	44Q	

Package Type	
44A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)

