SISTEM MONITORING SUHU MENGGUNAKAN

MIKROKONTROLLER AT89S51 DENGAN TAMPILAN DI PC

Diajukan Guna Melengkapi Sebagai Syarat
Dalam Mencapai Gelar Sarjana Strata Satu (S1)

Di susun Oleh :

Nama	:	**Andy Ihza Mahendra**
NIM	:	4140412-088
Jurusan	:	Teknik Elektro
Peminatan	:	Elektronika Industri
Pembimbing	:	Ir. Eko Ihsanto M,Eng

PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS MERCU BUANA
JAKARTA
2007

LEMBAR PENGESAHAN

Sistem Monitoring Suhu Menggunakan Mikrokontroller AT89S51 Dengan Tampilan Di PC

Disusun oleh

Nama	: Andy Ihza Mahendra
NIM	: 4140412-088
Jurusan	: Teknik Electro
Peminatan	: Elektronika

Disetujui dan disahkan oleh :

Pembimbing,

(Ir. Eko Ihsanto, MEng)

Mengetahui,

Ketua Jurusan,	Koordinator Tugas Akhir,
(Ir. Budi Yanto Husodo, MSc)	(Yudhi Gunardi, ST, MT)

ABSTRAK

Aplikasi monitoring suhu banyak ditemui diberbagai bidang industri diantaranya penyolderan pada pabrik perakitan elektronika karena temperature panas sangat membahayakan manusia maka sebaiknya monitoring suhu dilakukan dari jarak jauh sehingga mampu memberikan rasa aman bagi manusia

Tujuan dari aplikasi suhu ini adalah merancang dan merealisasikan instrumen monitoring suhu secara digital.dan membuat user interface untuk memudahkan user dalam memonitoring suhu pada suatu ruangan. Sistem aplikasi ini berbasis mikrokontroller AT 89S51 dan menggunakan pemrograman borland delphi

Berdasarkan hasil pengujian keseluruhan, suhu yang dideteksi oleh sensor dapat tampil pada PC dengan resolusi setara termometer air raksa, yaitu 1 derajat celcius.Suhu yang ditampilkan di PC berupa angka dan grafik. Alat ini mampu mendeteksi suhu antara 20 ° celcius sampai 100 ° celcius

Kata kunci : suhu, mikrokontroller

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAAN

1.1 Latar Belakang

Pengunaan sistem elektronika bagi dunia Industri dewasa ini sudah tidak dapat di pisahkan lagi, hampir semua Industri baik Industri besar maupun kecil menggunakan sistem elektronika sebagai alat bantunya, karena dengan menggunakan sistem elektronika dirasakan mampu mempermudah pengguna dalam melakukan pekerjaan, menekan biaya produksi bahkan mampu memberikan rasa aman bagi para penggunanya karena sistem elektronika dapat diletakan ditempat – tempat yang ekstrem, salah satu penggunaan sistem elektronika yang banyak digunakan adalah sistem elektronika yang menggunakan sensor temperature sebagai aplikasinya misalnya penyolderan (soldering) dipabrik perakitan alat- alat elektronika, karena temperature panas sangat membahayakan manusia maka sebaiknya monitoring temperature dilakukan dari jarak jauh sehingga mampu memberikan rasa aman bagi manusia.

Salah satu sistem elektronika yang banyak digunakan adalah sistem elektronika yang menggunakan mikrokontroller. Hal ini di karenakan mikrokontroller merupakan sistem mikroprosesor yang didalamnya terdapat CPU, ROM, RAM dan IO, sehingga penggunaan mikrokontroller menjadi sangatlah luas tidak hanya untuk akuisi data melainkan juga dapat di gunakan sebagai pengendali pada sebuah sistem

Berdasarkan keadaan tersebut diatas maka berkembanglah suatu gagasan untuk menciptakan sebuah alat yang dapat memonitoring keadaan suhu pada suatu ruangan dan memberikan informasi tentang berapa besar suhu pada ruangan tersebut. Sistem monitoring temperature ini berbasis mikrokontroller AT89S51 dengan tampilan grafik secara visual pada layar monitor PC, berdasarkan ide tersebut diatas dan teknologi yang ada, maka penulis memilih judul **SISTEM MONITORING SUHU MENGGUNAKAN MIKROKONTROLLER AT89S51 DENGAN TAMPILAN DI PC,** sebagai judul tugas akhir kami untuk melengkapi syarat kelulusan strata 1 program studi teknik elektronika universitas mercubuana

1.2 Tujuaan Penulisan

Adapun tujuaan dari penulisan ini adalah sebagai berikut

1. Merancang dan merealisasikan instrumen monitoring suhu secara digital.
2. Membuat user interface pada PC untuk mempermudah monitoring dan recording suhu.

1.3 Pembatasan Masalah

Sistem Monitoring Suhu Menggunakan Mikrokontroller AT89S51 dengan Tampilan di PC meliputi rangkaian elektronika Hardware dan Software, namun dalam pembatasan masalah pada penulisaan tugas akhir ini dititikberatkan pada pembahasan rangkaiaan Mikrokontroller,

rangkaiaan Temperature, rangkaiaan ADC, rangkaian Komunikasi Serial dan bahasa pemrograman Assembly dan Delfi

1.4 Metode Penulisaan

Dalam penulisan tugas akhir, penulis menggunakan berbagai cara untuk dapat menyelesaikan dengan baik dan benar, sebagai berikut

1. Metode observasi
2. Study literature
3. Percobaan
4. Konsultasi dengan dosen pembimbing

1.5 Sistematika Penulisaan

Untuk memberikan gambaran yang jelas mengenai urutan atau sistematika pembahasan dalam penyusunan tugas akhir ini, maka susunan penulisan tugas akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Berisi tentang latar belakang masalah, tujuan penulisan, pembatasan masalah, metode penulisan dan sistematika penulisan pada laporan tugas akhir ini.

BAB II TEORI DASAR

Membahas tentang teori dasar yang digunakan dan rancang bangun dalam membuat alat, diantaranya membahas arsitektur mikrokontroler AT89C51, Analog to Digital Converter (ADC), Sensor Suhu dan komponen pendukung lainnya.

BAB III PERANCANGAN

Merupakan penjelasan mengenai perancangan dan cara kerja rangkaian meliputi perancangan perangkat keras dan perangkat lunak

BAB IV PENGUJIAN ALAT

Merupakan bagian dari pengujian alat dan hasil dari analisa uji pada alat.

BAB V PENUTUP

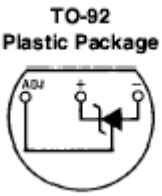
Berisikan kesimpulan yang didapat dari pembuatan alat yang telah dicapai selama perancangan dan pembuatan sistem/alat. Selain itu juga berisi saran-saran yang dapat dipakai untuk acuan pada pengembangan lebih lanjut.

BAB II

LANDASAN TEORI

2.1 Sensor Suhu (LM35)

Sensor temperatur LM35 ini merupakan sensor yang presisi, mudah untuk dikalibrasi, dan merupakan *Integrated Circuit Temperature Sensor* dimana output yang dihasilkan dari sensor ini sudah merupakan tegangan yang dapat langsung dihubungkan ke ADC ataupun lainnya untuk mendapatkan nilai yang kita inginkan. Prinsip bekerjanya hanya berdasarkan output yang dihasilkan oleh dua buah terminal dari Zener.



Gambar 2.1. Bagian dalam dari LM35

Sensor ini memiliki tegangan breakdown yang sama dengan temperatur absolut pada 10mV/°C. Dengan kurang dari 1Ω Impedansi Dynamicnya alat ini bekerja dengan range 400 μA sampa 5 mA. Sensor ini bekerja antara -40°C sampai 150°C dan cukup murah serta banyak kita dapatkan di pasaran.

Dalam pemakaiannya sensor temperatur LM35 diletakkan dalam ruangan untuk dapat langsung berinteraksi dengan kondisi suhu pada ruangan yang akan dikontrol. Setiap perubahan suhu ruangan yang terjadi akan dideteksi secara langsung oleh LM35 yang akan menyampaikan kondisi tersebut pada Analog to Digital Converter (ADC) dan selanjutnya ADC akan mengolah data yang didapatkan dari LM35 untuk menampilkan perubahan dalam setiap derajat celciusnya.

Sensor suhu seri LM35 ini terdiri atas 4 penguat operasi yang masing-masing berdiri sendiri, mempunyai penguatan tinggi dan secara intern terkompensasi terhadap frekuensi. Ini dirancang untuk dioperasikan dari pencatu daya tunggal dalam range tegangan yang lebar. Pemakaian arus yang kecil tidaklah bergantung pada besar tegangan catu daya, penerapannya meliputi sensor (tranduser), blok-blok penguatan DC dan rangkaian Op-Amp konvensional yang kini dapat dengan mudah dilengkapi pada sistem-sistem yang menerapkan pencatu daya tunggal. Perubahan dari temperature menjadi tegangan pada lm 35 di karenakan koefesian suhu pada zener yang ada didalam lm 35 yaitu perubahan tegangan zener per derajat celcius yang didefinisikan dengan rumus

$$\Delta Vz = Tc \times \Delta T \times Vz$$

Dimana : ΔVz = Perubahan tegangan zener

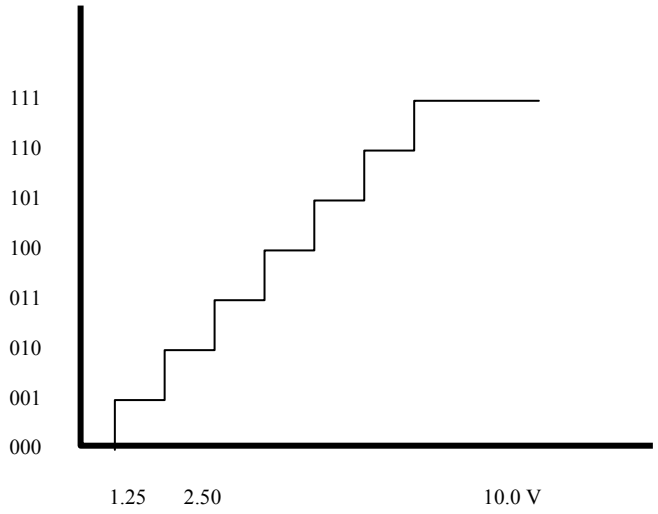
Tc = Koefisien suhu

Vz = Tegangan zener

2.2 Analog to Digital Converter (ADC)

Pengubah data dari analog ke digital merupakan salah satu alat yang berguna untuk mengkonversi data analog (berupa tegangan) ke dalam bentuk data digital yaitu data biner. Konsep dasar dari komponen ini terbagi atas dua buah proses yaitu Pencacahan dan pengkodean. Pencacahan merupakan sebuah proses untuk mentransformasikan sinyal analog ke dalam satu set kondisi output diskrit. Pengkodean merupakan sebuah proses untuk menentukan suatu kalimat kode digital menjadi beberapa kondisi-kondisi output.

Fungsi transfer nonlinear yang terlihat pada gambar 2.8 ialah suatu pencacah ideal dengan 8 buah kondisi output; dengan kalimat-kalimat kondisi output yang telah ditentukan, biasa juga disebut sebagai A/D Konverter 3 bit. 8 buah kondisi output ditentukan oleh urutan dari kode biner 000 sampai 111. Dengan masukan analognya 0 sampai 10V.



Gambar 2.2. Fungsi transfer dari A/D konverter 3 bit

Di dalam suatu komponen ADC, resolusi merupakan hal yang paling menentukan untuk mendapatkan data yang paling akurat. Resolusi dari ADC ditentukan oleh jumlah bit output yang dikeluarkan oleh ADC tersebut. Sebagai contoh untuk ADC 3 bit, seperti gambar diatas dengan ADC 8 bit memiliki resolusi yang berbeda. Jika ADC memiliki resolusi sebesar $1/2^n$, maka ADC 3 bit memiliki akurasi 1/8 atau 0,125% dari skala maksimum bila skala maksimumnya 5 Volt maka resolusinya sebesar 0.625 V untuk tiap kenaikan 1 angka desimal. Pada ADC 8 bit mempunyai tingkat resolusi sebesar 1/256 atau 0,0039% dari skala maksimum. Jadi bila skala maksimumnya 5 Volt maka resolusinya sebesar 0.019 V. Dengan kata lain kode biner 00000000 (0 desimal) akan bernilai 0V, kode 00000001 menjadi 0.019 V , 00000010 menjadi 0.039 V dan seterusnya hingga 11111111 menjadi +5 Volt. Perhatikan contoh Tabel konversi untuk ADC 8 bit dengan tegangan referensinya +5 V.

Tabel 2.1 Konversi ADC 8 Bit

Konversi ADC 8 bit								
Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Volts	2.5	1.25	0.625	0.3125	0.156	0.078	0.039	0.0195
Output Value	0	0	1	0	1	1	0	0

Dengan menjumlahkan tegangan yang pada tiap kode yang memiliki nilai ‘1’ dalam 00101100, maka kita akan mendapatkan :

0.625 + 0.156 + 0.078 = 0.859 Volts

Jadi tegangan yang wakilkan oleh kode biner tadi adalah 0.859 Volt.

ADC juga membutuhkan waktu yang cukup cepat untuk melakukan proses pencacahan dan pengkodean ini. Waktu yang dibutuhkan untuk melakukan proses ini tergantung atas beberapa faktor diantaranya resolusi dari konverter, teknik konversi dan kecepatan dari komponen-komponen yang bekerja didalam konverter tersebut. Kecepatan konversi yang dibutuhkan untuk aplikasi tertentu tergantung dari variasi waktu dari sinyal yang akan di konversi dan akurasi yang diinginkan.

ADC terdiri dari bermacam-macam jenis kecepatan, Interface yang berbeda, dan juga berbagai macam derajat akurasi. Jenis-jenis ADC yang paling umum digunakan diantaranya :

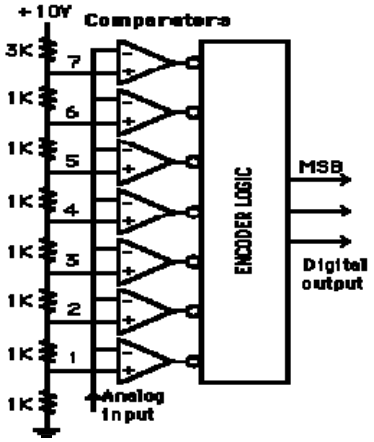
2.2.1. Flash ADC

Jenis ini merupakan jenis yang paling cepat saat ini. Flash ADC menggunakan banyak komparator, satu untuk setiap step tegangan, dan juga sederetan resistor. ADC 4 bit akan memiliki 16 buah komparator, dan ADC 8 bit akan memiliki 256 komparator. Seluruh output dari komparator akan terhubung ke sebuah blok logika yang nantinya akan menentukan mana komparator yang ‘low’ dan mana yang ‘high’.

Kecepatan konversi dari Flash ADC ini merupakan penjumlahan dari delay tiap-tiap komparator dan delay dari blok logika yang ada (umumnya waktu delay dari Blok logika ini dapat diabaikan). Flash ADC ini memang sangat cepat akan tetapi jenis ini banyak membutuhkan IC

yang sangat bagus dalam jumlah yang sangat besar. Juga, karena jumlah jumlah komparator yang digunakan, maka akan cenderung untuk memonopoli daya, menarik arus secara signifikan. Flash ADC 10 bit akan mungkin membutuhkan arus sampai setengah ampere.

Variasi lain dari konverter Flash ini adalah *half-flash*, yang menggunakan Digital-to-Analog Converter (DAC) dan proses pengurangan untuk mengurangi jumlah komparator yang ada didalam. Konverter *Half-flash* memang lebih lambat dari Konverter Flash yang sebenarnya tetapi lebih cepat dari ADC jenis lain. Jenis ini tetap dimasukan kedalam kategori konverter flash.

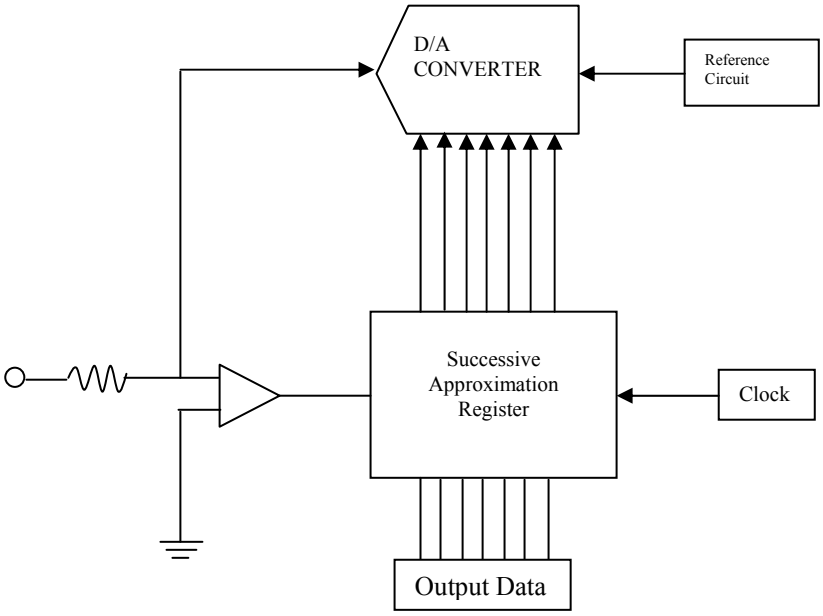


Gambar 2.3. Rangkaian dalam dari Flash ADC

2.2.2. Converter Successive Approximation

Jenis ini menggunakan komparator dan logika pencacahan untuk menghasilkan konversi yang diinginkan. Langkah pertama dalam konversi ini adalah untuk melihat jika masukan lebih besar dari setengah tegangan referensi. Jika benar, maka nilai MSB (most significant bit) dari

keluarannya menjadi ‘1’ (SET). Nilai ini kemudian dikurangi nilai masukan, dan hasilnya akan di cek selama seperempat dari tegangan referensi. Proses ini akan berlangsung terus hingga semua bit keluarannya telah menjadi ‘1’ semua atau ‘0’ semua. Jenis ini banyak memakan *clock cycle* yang dikarenakan bit-bit yang dikeluarkan untuk melakukan konversi.

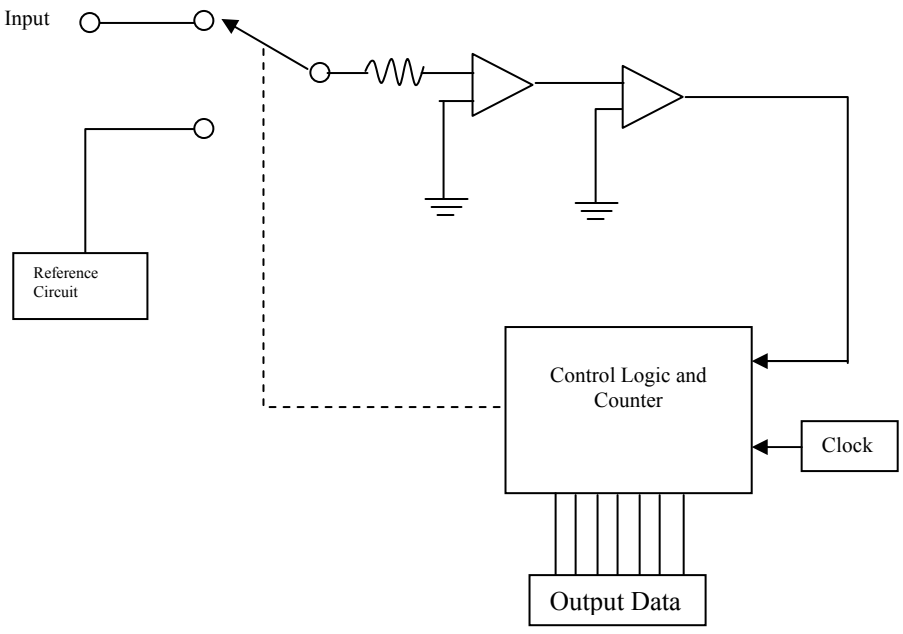


Gambar 2.4. Rangkaian Converter Successive Approximation.

2.2.3. Sigma-delta/Dual Slope

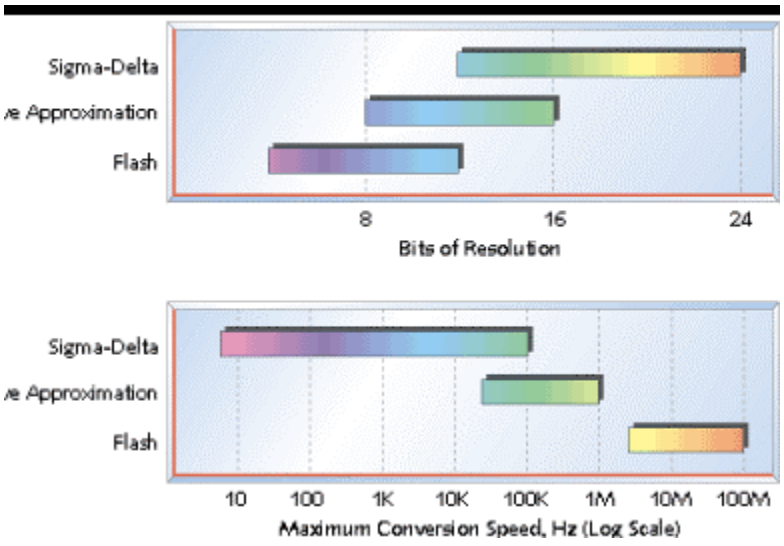
Jenis ini menggunakan DAC 1-bit, filterisasi, dan *oversampling* untuk memperoleh konversi yang sangat akurat. Akurasi dari proses konversi ini dikontrol oleh referensi masukan dan juga masukan *clock* rata-rata. Keuntungan utama dari konverter ini yaitu memilik resolusi yang

tinggi. Pada dua jenis konverter yang sebelumnya banyak menggunakan resistor. Permasalahannya adalah akurasi dari resistor-resistor tersebut akan langsung mempengaruhi akurasi dari hasil konversi. Sedangkan konverter sigma-delta ini tidak menggunakan rangkaian resistor tetapi mengumpulkan jumlah sampel yang ada ke dalam suatu hasil. Kekurangan dari konverter ini adalah kecepatannya. Karena konverter bekerja dengan menggunakan pengambilan kembali data masukan (oversampling), maka konversi akan menggunakan banyak *clock cycle.*



Gambar 2.5. Rangkaian konverter sigma-delta

Digambar 2.6 ditunjukan batasan-batasan dari resolusi yang dimiliki oleh sigma-delta, successive approximation, dan converter flash.



Gambar 2.6. Perbandingan antar konverter

Dari perbandingan diatas maka kita dapat menentukan pilihan yang terbaik untuk digunakan di alat yang ingin dibuat. Dan jelas, jika ingin memilih ADC dengan kecepatan tinggi maka akan dipilih flash ADC, tetapi jika kita lebih memilih resolusi yang tinggi maka lebih baik untuk memilih Sigma-Delta. Pada umumnya Successive Approximation ADC merupakan pilihan yang paling banyak digunakan karena kecepatannya yang cukup dan resolusi yang juga cukup baik.

2.3 Mikrokontroler AT89S52

2.3.1 Keluarga MCS-51

Atmel merupakan salah satu perusahaan *IC* yang mengembangkan mikrokontroler keluarga MCS-51. Atmel banyak memproduksi *Flash Programmable and Erasable ROM (flash PEROM*) yaitu salah satu EPROM yang sangat murah dan mudah pemakaiannya. Flash PEROM

adalah ROM *(Read Only Memory)* yang dapat dihapus dan ditulis kembali dengan teknologi *flash.* Kelebihan *flash* ini adalah mikrokontroler dapat menyimpan program secara *internal*, tidak membutuhkan ROM *eksternal.* Program dapat langsung ditulis sehingga menimpa program yang lama, apabila program yang lama akan diganti program yang baru. Maka pemakaian mikrokontroler menjadi sederhana, murah dan pemakaiannya menjadi cepat. *Flash* PEROM sejenis inilah yang dipakai untuk menyimpan BIOS semua PC untuk saat ini.

Atmel mengaplikasikan teknologi *Flash* PEROM ini ke dalam keluarga MCS-51 buatannya sendiri yang diberi nama AT89C51, AT89C52, AT89C2051 AT89S51, AT89S52, dan sebagainnya.

Bila dilihat dari segi arsitekturnya, mikrokontroler keluarga MCS-51 dapat dibagi menjadi dua jenis, yaitu mikrokontroler menggunakan EPROM dan tanpa menggunakan EPROM. Untuk mikrokontroler yang tidak menggunakan ROM/EPROM didalam maka harus menambahkan EPROM luar yang dihubungkan dengan mikrokontroler tersebut melalui *port* pararel (*port* 0 dan *port* 2). Perbandingan mikrokontroler berbagai jenis dari keluarga MCS-51 ditunjukkan Tabel 2.2.

Tabel 2.2 Perbandingan mikrokontroler keluarga MCS-51

Device	*Type*	Memori *Internal*				*Type*(16 bit)
		ROM	EPROM	RAM	PEROM	
8031AH	NMOS	-	-	128 *byte*	-	2

8051AH	NMOS	4 K*byte*	-	128 *byte*		2
80C31BH	CMOS	-	-	128 *byte*	-	2
80C51BH	CMOS	4 K*byte*	-	128 *byte*	-	2
87C51	CMOS	-	4 K*byte*	128 *byte*	-	2
89C51	CMOS	-	-	128 *byte*	4 K*byte*	2

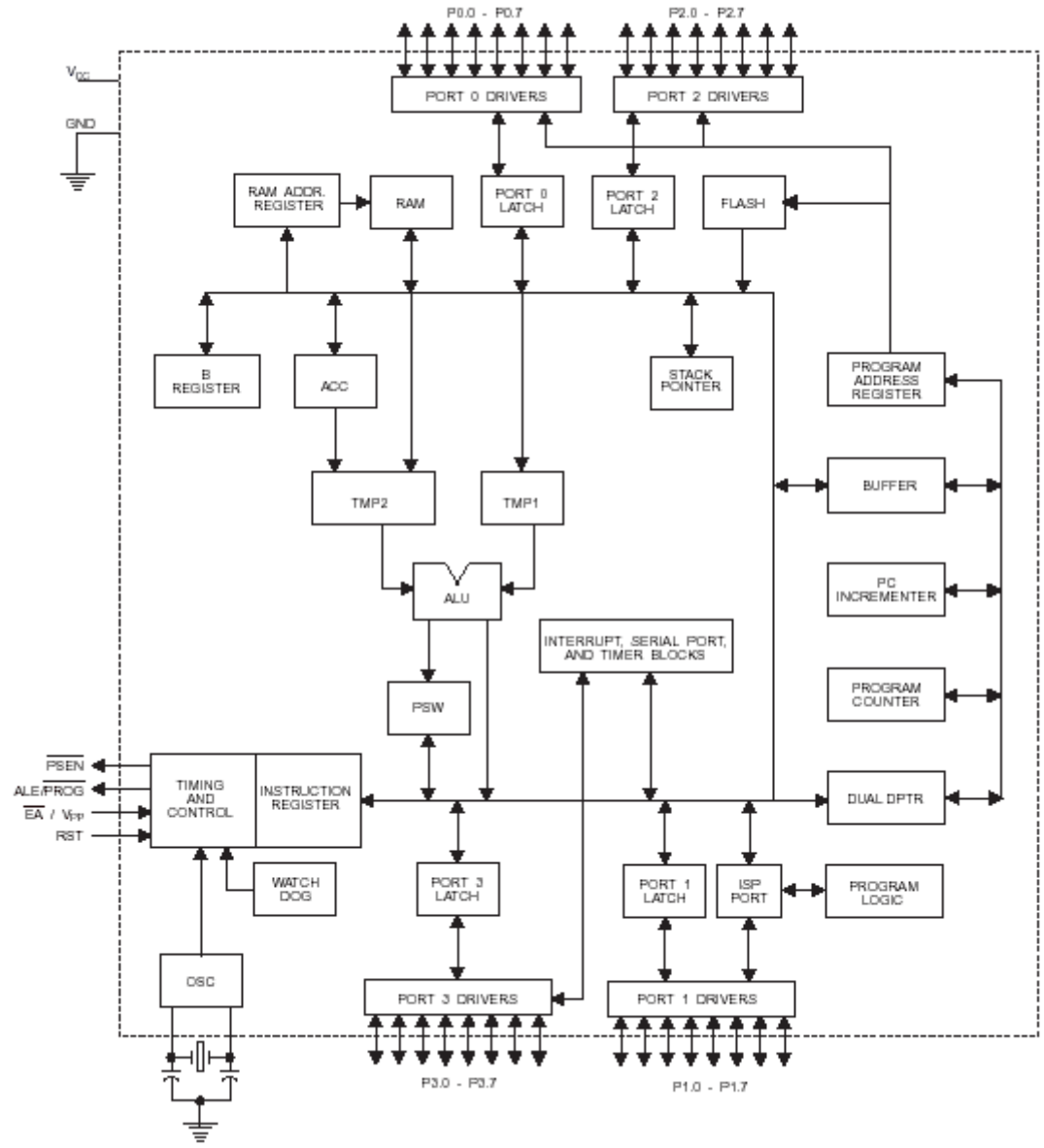
Prototype dari mikrokontroler AT89C51 adalah 89C51. Kedua tipe ini memiliki persamaan yaitu berisi RAM, *Timer/Counter*, *port* pararel, dan *port* serial. Mikrokontroler 87C51 memiliki EPROM, sedangkan AT89C51 menggunakan Flash PEROM *(Programmable Erasable Read Only Memory)*. Pada PEROM inilah program dapat ditulis dihapus dan ditulis kembali sehingga lebih efisien karena tidak membutuhkan ROM *eksternal*. Sedangkan untuk mikrokontroler tipe 8031 tidak memililki ROM/EPROM, sehingga harus menggunakan EPROM *eksternal* untuk menyimpan program.

2.3.2 Arsitektur Mikrokontroler AT89S51

Mikrokontroler AT89S51 menggunakan *Flash Programmable Erasable Read Only Memory (Flash PEROM)* yang mempunyai bayak kepraktisan sehingga penghapusan data dapat dilakukan dengan cepat atau serentak (tidak byte demi byte seperti pada EPROM) yang dapat dilakukan secara elektris.

Mikrokontroler AT89S51 merupakan mikrokomputer CMOS 8 bit dengan 8 *Kbytes Flash Programmable Memory*. Arsitektur AT89S51 ditunjukkan Gambar 2.7

.



Gambar 2.7 Arsitektur mikrokontroler AT89S51

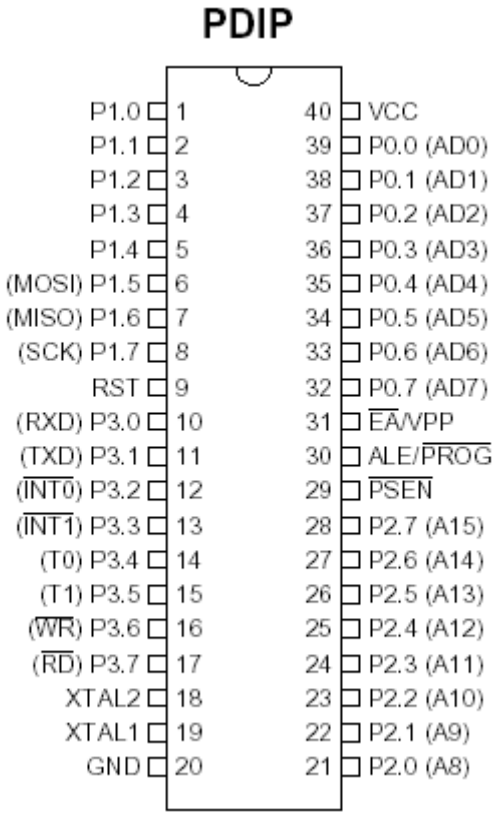
Mikrokontroler AT89S51 memiliki karakteristik-katakteristik yang cukup menguntungkan dan memudahkan dalam merancang suatu alat. Karakteristik-karakteristik itu diantaranya adalah:

1. Kompatibel dengan mikrokontroler MCS-51
2. *8 Kbytes Flash PEROM.*
3. Tahan 1000 kali pengulangan penulisan dan penghapusan.
4. Frekuensi kerja antara 0 sampai 24MHz.
5. Memiliki tiga tingkat penguncian memori.
6. Memiliki 128 X 8-bit memori internal (RAM).
7. Memiliki 2 unit 16-bit *Timer/Counter*.
8. Memiliki 6 sumber interupsi.
9. Memiliki serial port yang dapat diprogram.
10. Tegangan operasi 4,0V sampai 5,5V.
11. *Programmable Watchdog Timer*.
12. SPI *Serial Interface.*

Dengan karakteristik diatas menjadikan pembuatan suatu sistem menjadi lebih efisien, baik dalam proses pembuatan rancangan perangkat keras maupun perangkat-lunaknya.

Mikrokontroler AT89S51 mempunyai 40 kaki, 32 kaki diantaranya adalah kaki untuk keperluan *port* parallel. Satu *port* parallel terdiri atas 8 kaki, dengan demikian 32 kaki tersebut membentuk 4 buah *port* parallel, yang masing-masing dikenal dengan *Port0*, *Port1*, *Port2*, *Port3*. Nomor

masing-masing jalur (kaki) *port* parallel mulai dari 0 sampai 7, jalur (kaki) pertama sebagai P0.0 dan jalur terakhir untuk *Port3* adalah *Port3.7*.



Gambar 2.8 Susunan pin mikrokontroler AT89S51

Gambar 2.8 diatas menunjukkan pin-pin dari mikrokontroler AT89S51, berikut adalah penjelasan serta fungsi masing-masing pin mikrokontroler AT89S51.

1. Vcc (pin 40)

Catu daya 4,0V sampai 5,5V DC.

2. GND (pin 20)

Ground.

3. Port 0 (pin32-39)

Port 0 merupakan *port I/O* 8-bit yang bersifat bidireksional. Masing-masing pin dapat dihubungkan secara langsung dengan 8 input TTL. Untuk desain sistem yang minimum, *port* ini digunakan sebagai I/O untuk tujuan umum. Namun aplikasi yang menggunakan memori eksternal, maka port 0 mengeluarkan *"low order byte"* alamat memori *eksternal* (A0-A7), yang dimultipleks dengan data 8-bit yang dibaca dan ditulis,sehingga *port* ini juga ditulis sebagai AD0, AD1, …, AD7.

4. *Port* 1 (pin 1-8)

Port 1 bersifat bidireksional. Masing-masing pin diberi nama P1.0, P1.1, …, P1.7 yang digunakan untuk berhubungan dengan alat diluarnya. Pin ini memiliki fungsi alternatif yaitu sebagai jalur MOSI (P1.5), MISO (P1.6), dan SCK (P1.7) pada pemrograman secara serial.

5. *Por*t 2 (pin 21-28)

Port 2 adalah port yang memiliki dua fungsi, yaitu sebagai port I/O 8-bit untuk tujuan umum dan juga dapat berfungsi sebagai bus alamat untuk *byte* yang lebih tinggi jika didesain menggunakan memori *eksternal*. *Port* 2 *mengeluarkan "high order byte"* alamat memori eksternal (A8-A15).

6. *Port 3* (pin10-17)

Selain sebagai *port* 8-bit I/O untuk tujuan umum, pin-pin pada *port* ini juga bersifat multifungsi, salah satu fungsi alternatifnya berhubungan untuk fungsi yang spesial dari mikrokontroler ini. Fungsi alternatif dari *port* 3 ini ditunjukkan secara lengkap pada Tabel 2.3.

Tabel 2.3 Fungsi alternative *port* 3

Pin	Nama	Fungsi Alternatif
P3.0	RXD	*Port input serial*
P3.1	TXD	*Port output serial*
P3.2	INT0	*Interupsi eksternal*
P3.3	INT1	*Interupsi eksternal*
P3.4	T0	*Input Timer/Counter 0 eksternal*
P3.5	T1	*Input Timer/Counter 1 eksternal*
P3.6	WR	Sinyal tulis memori data *eksternal*
P3.7	RD	Sinyal baca memori data *eksternal*

7. RST (pin9)

RST adalah masukan *reset* yang aktif tinggi yang akan me-*reset* mikrokontroler AT89S52 yaitu ketika sinyal tinggi diberikan selama kurang lebih dua siklus mesin, sistem akan me-*reset* untuk kembali ke keadaan awal.

8. ALE/PROG *(Address Latch Enable)* (pin 30)

Pulsa keluaran ALE digunakan untuk menahan alamat byte rendah (*low byte*) selama mengakses memori *eksternal*. Pin ini juga sebagai masukan pulsa program selama pemrograman *Flash*.

9. PSEN *(Program Store Enable)* (pin 29)

Program Store Enable (PSEN) merupakan suatu sinyal *output* yang biasanya merupakan suatu sinyal kontrol untuk memori program *eksternal*. PSEN biasanya dihubungkan dengan pin *Output Enable* (OE) suatu EPROM untuk mengizinkan pembacaan *byte-byte* program.

10. EA/Vpp *(External Access Enable)* (pin 31)

EA harus dihubungkan dengan ground jika akan mengakses memori program *eksternal* yang dimulai di alamat 0000H sampai FFFFH. Sebaliknya, EA seharusnya dihubungkan ke Vcc untuk mengeksekusi program *internal*.

11. XTAL1 (pin 19)

Input untuk *inverting oscillator amplifier* dan masukan untuk pengoperasian rangkaian detak *(clock) internal*.

12. XTAL2 (pin 20)

Output dari *inverting oscillator amplifier*.

2.3.3 *Special Function Registers (SFR)*

SFR merupakan register dengan tugas khusus yang terdapat pada alamat 80H sampai FFH. Dengan demikian mikrokontroler AT89S52 memiliki 128 lokasi alamat untuk SFR, namun demikian pada mikrokontroler ini tidak berarti memiliki SFR sebanyak 128 buah. Untuk lebih jelasnya dapat dilihat Gambar 2.9.

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H									0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000								0AFH
0A0H	P2 11111111		AUXR1 XXXXXXX0				WDTRST XXXXXXXX		0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX00XX0		8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000		PCON 0XXX0000	87H

Gambar 2.9 Pemetaan *Special Fungsi Register (SFR)* pada AT89S51

Adanya alamat kosong yang tidak ditempati oleh SFR, memungkinkan vendor-vendor lain untuk dapat mengembangkan lebih baik.

Fungsi masing-masing register dijelaskan sebagai berikut:

- ***Accumulator***

  ACC merupakan register akumulator. Pada program ditulis dengan A.

- **Register B**

  Register B digunakan pada operasi perkalian dan pembagian. Pada instruksi-instruksi yang lain berfungsi seperti register umumnya.

- ***Program Status Word*** **(PSW)**

  PSW berisi informasi status program

- ***Stack Pointer*** **(SP)**

  Stack Pointer terdiri dari 8 bit. Alamat SP ditambah/dinaikkan sebelum data disimpan pada eksekusi instruksi PUSH dan CALL. SP dapat diletakkan pada alamat mana pun di *on-chip* RAM, SP diinisialisasi pada alamat 07H setelah *reset*. Hal ini mengakibatkan *stack* dimulai pada lokasi 08H.

- ***Data Pointer*** **(DPTR)**

  DPTR terdiri atas *high byte* (DPH) dan *low byte* (DPL). Fungsi utamanya adalah sebagai tempat alamat 16 bit. Register ini bisa juga dimanipulasi sebagai sebuah register 16 bit atau 2 buah register 8 bit yang berdiri sendiri.

- ***Port*** **0 - 3**

P0, P1, P2, dan P3 adalah *SFR latch* dari *Port* 0, 1, 2, dan 3.

- ***Serial Data Buffer***

Serial Data Buffer sebenarnya merupakan 2 register yang terpisah, *transmit buffer* (untuk mengirim data serial) dan *receive buffer* (untuk menerima data serial). Ketika data dipindahkan ke SBUF, maka data akan menuju ke *transmit buffer* tempat data ditampung untuk pengiriman serial. Memindahkan data ke SBUF berarti menginisialisasi/memulai transmisi data secara serial. Sebaliknya bila data dipindahkan dari SBUF, data tersebut berasal dari *receive buffer*.

- **Register *Timer***

Pasangan register (TH0 & TL0), (TH1 & TL1), serta (TH2 & TL2) adalah register 16 bit untuk proses perhitungan *Timer/Counter* 0, 1, dan 2.

- **Register Control**

FR IP, IE, TMOD, TCON, T2CON, SCON, dan PCON berisi bit kontrol dan status untuk sistem interupt, *timer/counter*, dan *port* serial.

2.3.3 Struktur dan Cara Kerja Port

AT89S51 mempunyai 4 *port bidirectional* (*Port 0 - Port 3*), yang masing-masing terdiri atas 8 bit. Setiap *port* terdiri atas satu unit *latc*h (*Special Function Registers* P0 sampai P3),sebuah *output driver*, dan satu *input buffer*. *Output driver port 0* dan *port 2*, serta *input buffer Port 0* digunakan untuk mengakses memori *eksternal*. Untuk aplikasi yang menggunakan memori *eksternal,* maka p*ort* 0 mengeluarkan *'low order*

byte' alamat memori *eksternal* (A0-A7), yang dimultipleks dengan data (1 *byte*) yang dibaca atau ditulis. *Port* 2 mengeluarkan *'high order byte'* alamat memori *eksternal* (A8-A15) bila alamat yang diperlukan sebanyak 16 bit. Bila alamat yang diperlukan hanya A0-A7 maka *output port* 2 sama dengan isi *SFR (Special Function Registers).* Semua pin *Port* 3 mempunyai fungsi alternatif selain sebagai *port*.

2.3.4 Konfigurasi I/O

Gambar 2.10 menunjukkan diagram *latch* dan *I/O buffer* tiap bit pada p*ort 0 - Port 3. Port 1,*2, dan 3 mempunyai *pull-up internal*. Sedangkan *port 0*, konfigurasi outputnya adalah *open drain*. Setiap bit I/O ini berdiri sendiri, jadi dapat berfungsi sebagai *input* atau *output* tanpa tergantung satu sama lain. *port 0* dan 2 tidak dapat dipakai sebagai I/O bila digunakan sebagai jalur alamat/data. Bila *port-port* tersebut ingin difungsikan sebagai *inpu*t, maka *bit latch* harus berisi '1',yang akan mematikan *output driver FET*. Maka pin-pin *port* 1,2, dan 3 akan 'ditarik' ke *high* oleh *pull-up internal*, tetapi bila diinginkan dapat juga 'ditarik' ke *low* dengan sumber *eksternal*. *Port 0* agak berbeda, karena tidak menggunakan *pull-up interna*l. *FET pull-up* pada *output driver* P0 hanya digunakan pada saat *Port* mengeluarkan '1' selama akses memori *eksternal*, selain keadaan ini *FET pull-up* tidak aktif. Akibatnya bila bit-bit P0 berfungsi sebagai *output* maka bersifat *open drain*. Penulisan logika '1'

ke bit latch menyebabkan kedua FET tidak bekerja, sehingga pin dalam keadaan mengambang (*floating*). Pada kondisi ini pin dapat berfungsi sebagai *high impedance input*. p*ort* 1,2, dan 3 sering disebut dengan *'quasibidirectional'* karena mempunyai *pull-up internal*. Saat berfungsi sebagai *input* maka mereka akan 'ditarik' ke tinggi (*high)* dan akan bersifat sebagai sumber arus bila 'ditarik' ke rendah (*low)* secara eksternal. *port* 0 sering disebut sebagai *'true-bidirectional'*, karena bila dikonfigurasikan sebagai input maka pin-pinnya akan mengambang. Pada saat *reset* semua *port latch* akan berlogika '1' .

2.3.5 Akses Memori

Mengakses memori *eksternal* ada 2 macam:

- Akses Program *Memory eksternal*
- Akses Data *Memory eksternal*.

Mengakses program *memory eksternal* menggunakan sinyal PSEN (*Program Store Enable*) sebagai sinyal baca. Sedangkan untuk mengakses data memori *eksternal* digunakan RD dan WR (fungsi alternatif P3.7 dan P3.6) untuk membaca dan menulis ke memori. Membaca program *memory eksternal* selalu menggunakan alamat 16 bit. Sedangkan untuk mengakses data *memory eksternal* dapat menggunakan alamat 16 bit (MOVX @DPTR) atau alamat 8 bit (MOVX @Rx). Pada saat alamat 16 bit digunakan, *high byte* dari jalur alamat dihasilkan oleh *port* 2, yang dipertahankan selama siklus pembacaan atau penulisan. p*ort* 2 mempunyai

pull-up yang kuat selama mengeluarkan bit alamat '1' (pada saat eksekusi instruksi MOVX @DPTR). Pada saat ini latch *port* 2 (SFR) tidak selalu berisi '1', dan isi SFR *port* 2 tidak berubah. Bila siklus memori *eksternal* tidak segera diikuti siklus memori *eksternal* yang lain maka isi SFR *port* 2 yang tidak berubah tersebut akan muncul kembali pada siklus berikutnya. Bila menggunakan alamat 8 bit (MOVX @Ri), isi SFR *port* 2 tetap sama dengan pin *port* 2 selama siklus memori *eksternal*. Karakteristik ini memberikan kemampuan *paging* memori. Bit rendah dari alamat bersifat *time* m*ultiplexed* dengan data byte *port* 0, artinya data dan alamat dihasilkan oleh pin yang sama secara bergantian dengan selang waktu tertentu. Sinyal alamat/data mengaktifkan kedua FET pada *output buffer port* 0. Jadi dalam aplikasi ini pin-pin *port* 0 tidak bersifat sebagai *output opendrain*, dan tidak memerlukan *pull-up eksternal*. Sinyal ALE (*Address Latch Enable*) digunakan untuk menyimpan bit alamat ke sebuah *latch eksternal*. Bit alamat valid pada saat transisi negatif ALE. Pada siklus penulisan, data yang akan dituliskan muncul pada *port* 0 tepat sebelum WR aktif, dan data ini tetap ada sampai WR dinonaktifkan. Pada siklus pembacaan, data bit diterima oleh *port* 0 sesaat sebelum sinyal RD dinonaktifkan.

2.3.6 *Timer / Counter*

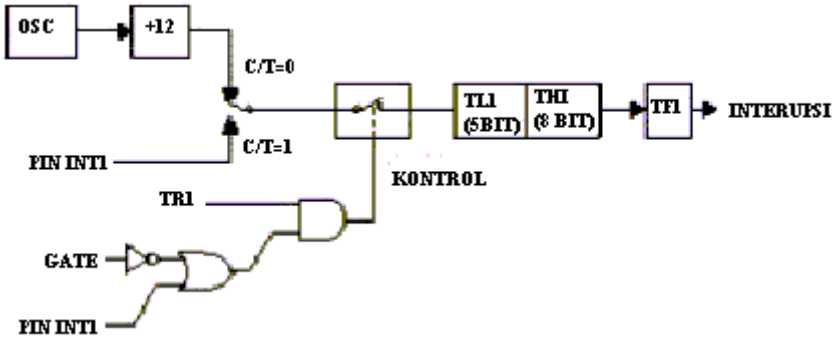
AT89S51 mempunyai 2 unit register *timer /counter* 16 bit : *Timer* 0 dan *Timer* 1. Keduanya dapat beroperasi sebagai *timer atau counter*. Pada fungsi *'timer'*, isi register ditambah satu setiap siklus mesin. Jadi,

seperti menghitung siklus mesin. Karena satu siklus mesin terdiri atas 12 periode osilator, maka kecepatannya t = 1/12 frekuensi osilator. Pada fungsi *'counter'*, isi register ditambah satu setiap terjadi transisi 1 ke 0 pada pin *input eksternal* yang bersesuaian T0 atau T1. Untuk mengenali transisi 1 ke 0 ini dibutuhkan 2 siklus mesin (24 periode osilator), maka input maksimum ialah 1/24 frekuensi osilator. Tidak ada batasan untuk *duty cycle* sinyal input. t*imer 0* dan t*imer 1* mempunyai 4 mode operasi yang bisa dipilih. Fungsi *timer* dan *counter* dipilih dengan bit kontrol C/T pada SFR TMOD. Kedua *timer/counter* ini mempunyai 4 mode operasi yang dipilih dengan sepasang bit M1 dan M0

- *Mode* 0

Ttimer 0 dan *timer 1* pada mode ini berfungsi sebagai *counter* 8 bit dengan *divided-by-32 prescaler*. Operasi mode 0 pada *timer* 1, sehingga konfigurasi register *timer* menjadi 13 bit. Ketika perhitungan berubah dari nilai maksimum (semua bit = 1) menjadi 0 maka *flag interupt timer* TF1 akan aktif. Masuakan akan dihitung oleh *timer* bila TR1=1 dan salah satu *GATE*=0 atau INT1=1. Bila *GATE* di-*set* = 1 maka *timer* dikontrol oleh *input eksternal* INT1, dan dapat digunakan untuk mengukur lebar pulsa. TR1 adalah bit kontrol pada SFR TCON, sedangkan *GATE* ada pada TMOD. Men-*set* TR1 Register 13 bit terdiri atas 8 bit TH1 dan 5 bit TL1. 3 bit TL1 bagian atas dapat diabaikan. Men-*set* TR1 tidak menghapus isi register. Mode 0 untuk t*imer* 0 sama seperti t*imer* 1. Substitusi TR1, TF1

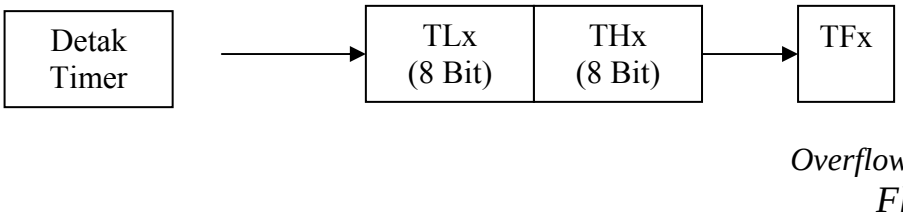
dan INT1 dengan TR0, TF0, dan INT0. Ada 2 bit *GATE* yang berbeda, yaitu TMOD.7/TMOD bit ke 7 untuk t*imer* 1 dan TMOD.3/TMOD bit ke 3 untuk *timer* 0. *Mode* 0 ditunjukkan dengan Gambar 2.8.



Gambar 2.10 *Timer/Counter 1 Mode 0* : *Counter 13 Bit*

- *Mode* 1

Mode 1 sama dengan mode 0, kecuali register *timer* berjalan dengan 16 bit. Jadi semua bit pada TH1/TL1 (*timer 1*) atau TH0/TL0 (t*imer* 0) berfungsi. Untuk lebih jelasnya lihat Gambar 2.11.

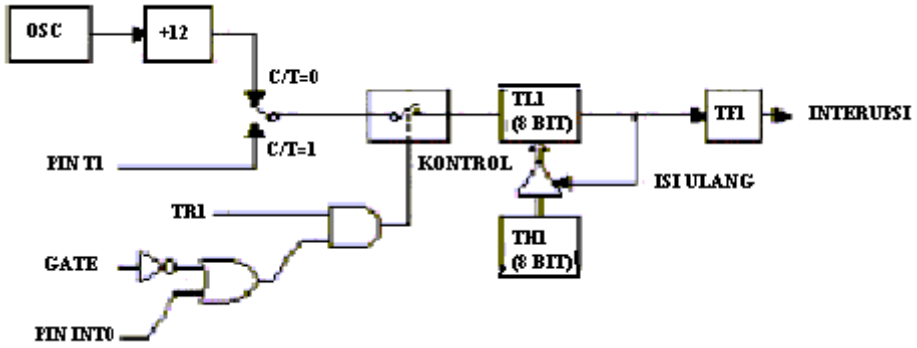


Gambar 2.11 *Timer/Counter 1 Mode 1 : Counter 16 bit*

- *Mode* 2

Pada mode ini *register timer* berfungsi sebagai *counter* 8 bit (TL1) dengan isi ulang otomatis. *Overflow* dari TL1 tidak hanya men-*set* TF1, tetapi juga mengisi-ulang TL1 dengan isi TH1, yang ditentukan dengan

software. Proses isi- ulang ini tidak mengakibatkan isi TH1 berubah. Mode 2 untuk *timer/counter 0* sama seperti *timer/counter 1*. Untuk lebih jelasnya lihat Gambar 2.12.



Gambar 2.12 *Timer/Counter 1 Mode 2* : 8 Bit dengan isi ulang otomatis (*Auto Reload*)

bit 7	bit6	bit 5	bit 4	bit3	bit 2	bit 1	bit 0
GATE	C/~T	M1	M0	GATE	C/~T	M1	M0
		Timer 1				*Timer 2*	

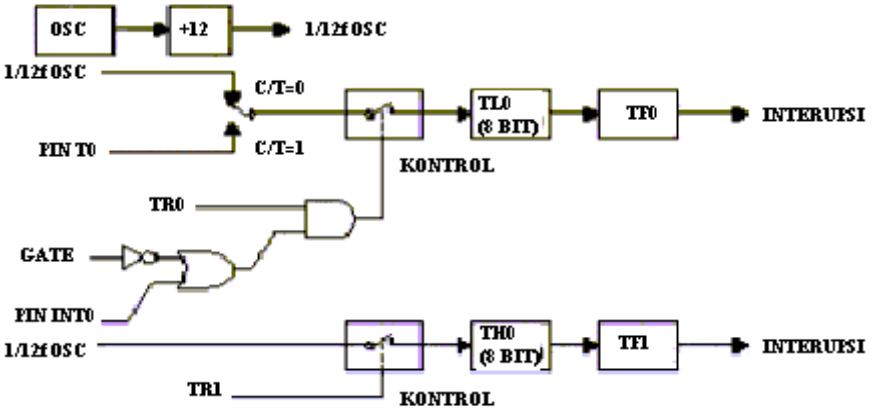
Gambar 2.13 Susunan *bit* dalam *register* TMOD.

Pada Gambar 2.13 menunjukkan susunan *bit register* TMOD pada *timer* 1 dan timer 2 yang menentukan fungsi dan jenis kerja dari *counter* atau *timer*.

- *Mode* 3

Timer 1 pada Mode 3 tidak menghitung sama sekali, sama seperti men-set TR1 = 0. *Timer 0* pada mode 3 menjadikan TL0 dan TH0 sebagai 2 *counter* yang terpisah. Cara kerja t*imer 1* pada mode 3. TL0

menggunakan bit kontrol t*imer 0* :C/T, GATE, TR0, INT0, dan TF0. TH0 berfungsi sebagai *timer* yang menghitung siklus mesin dan mengambil alih kontrol TR1 dan TF1 dari *timer* 1. Jadi TH0 sekarang mengontrol *interupt timer 1*. Mode 3 ini digunakan untuk aplikasi yang membutuhkan satu *timer* atau *counter* 8 bit tambahan. Dengan *timer 0* pada Mode 3, AT89S52 seolah-olah mempunyai 3 unit *timer /counter*. Ketika t*imer 0* bekerja pada mode 3, t*imer 1* dapat diaktifkan pada mode yang lain. Sebagai contoh *timer* 1 dapat digunakan sebagai *baud rate* generator atau aplikasi apapun yang tidak memerlukan interupsi. Untuk lebih jelasnya lihat Gambar 2.14.



Gambar 2.14 *Timer/Counter 1* Mode 3, 2 Counter 8 Bit

2.3.7 *Serial Interface*

Port serial AT89S52 bersifat *full duplex*, jadi dapat mengirim dan menerima data (byte) secara simultan. Selain itu ada *buffer* penerima, sehingga port serial dapat bersiap menerima data kedua sebelum data pertama dibaca dari register penerima. Namun bila data pertama belum

dibaca juga sampai data kedua diterima lengkap, maka salah satu data tersebut akan hilang. Register penerima dan pengirim *port* serial diakses melalui SFR SBUF. Menulis ke SBUF berarti mengisi register pengirim, dan membaca SBUF berarti mengakses register penerima yang terpisah. *Port* serial dapat bekerja dalam 4 mode berikut.

- **Mode 0:** data serial masuk dan keluar melalui RXD. TXD mengeluarkan sinyal *clock*. 8 bit data dikirim/diterima dengan bit *LSB (Least Significant Bit)* yang pertama. *Baud rate* tetap pada 1/12 frekuensi osilator.

- **Mode 1:** 10 bit dikirim melalui TXD atau diterima melalui RXD yang terdiri atas satu *start* bit (0), 8 bit data (LSB pertama), dan satu *stop* bit (1). Pada penerimaan, *stop* bit menuju RB8 pada SFR SCON. *Baud rate variabel*.

- **Mode 2:** 11 bit dikirim melalui TXD atau diterima melalui RXD, sebuah start bit (0), 8 bit data (LSB pertama), bit data ke 9 yang terprogram, dan sebuah *stop* bit (1). Pada saat pengiriman, bit data ke 9 (TB8 pada SCON) dapat diberi nilai 0 atau 1. Sebagai contoh bit paritas (P pada PSW) dapat dipindahkan ke TB8. Pada penerimaan, bit data ke 9 masuk ke RB8 pada SCON sedangkan *stop* bit diabaikan. *Baud rate* dapat diprogram 1/32 atau 1/64 frekuensi osilator.

- **Mode 3 :** 11 bit dikirim melalui TXD atau diterima melalui RXD, satu *start bit* (0), 8 *bit* data (pertama LSB), *bit* data ke-9 yang terprogram dan satu *stop bit* (1). Sebenarnya mode 3 sama seperti mode 2, namun *baud rate* Mode 3 variabel. Pada semua mode di atas, pengiriman diinisialisasi

dengan instruksi yang menggunakan SBUF sebagai register tujuan. Penerimaan diinisialisasi pada mode 0 dengan kondisi RI = 0 dan REN = 1. Pada mode lain penerimaan diinisialisasi dengan diterimanya *start bit* dengan syarat REN = 1.

2.3.8 *Serial Port Control Register*

Kontrol dan status *port* serial terdapat pada SFR SCON ditunjukkan Gambar 2.13. Register ini berfungsi *bit* untuk memilih mode operasi, *bit* data ke-9 untuk penerimaan dan pengiriman (TB8 dan RB8), serta *bit interrupt port serial* (TI dan RI). Untuk lebih jelasnya lihat Tabel 2.4. Pada AT89S52, *baud rate* untuk mode 1 dan mode 3 ditentukan oleh *overflow rate timer 1*.

MSB LSB

SM0	SM1	SM2	REN	TB8	RB8	T1	R1

Gambar 2.15 Susunan SCON *port serial*

Tabel 2.4 Fungsi bit-bit SCON

Bit	Simbol	Alamat	Keterangan
SCON.7	SM0	9FH	Mode *port* serial bit 0
SCON.6	SM1	9EH	Mode *port* serial bit 1
SCON.5	SM2	9DH	Mode *port* serial bit 2
SCON.4	REN	9CH	*Receive Enable*.Harus diset untuk menerima karakter
SCON.3	TB8	9BH	*Transmit bit 8*.Bit ke-9 dikirimkan dalam mode 2 dan 3,di *set/clear* oleh software
SCON.2	RB8	9AH	*Receive bit 8*.Bit ke-9 diterima
SCON.1	TI	99H	*Transmit Interrupt flag*.Di *set* pada akhir pengiriman karakter,di *clear* oleh

			software
SCON.0	RI	98H	*Receive Interrupt flag*.Di set pada akhir penerimaan karakter,di *set* oleh *software*

Nilai dari SMOD adalah sebagai berikut :

$$Baud\,Rate\,Mode1,3 = \frac{2^{SMOD}}{32} X (Timer1\,Overflow\,Rate) \qquad (2.2)$$

Untuk aplikasi ini interupsi *timer 1* harus dinonaktifkan. *Timer* bisa dioperasikan sebagai *'timer'* atau *'counter'*, dan bisa menggunakan salah satu dari 3 mode ketika *timer* 1 aktif. Pada umumnya *timer* 1 dikonfigurasikan sebagai *'timer'* dengan mode *auto-reload* (mode 2) di mana *high nibble/4 bit upper* TMOD = 0010B. Dalam kondisi ini, rumus untuk *baud rate.*

$$Baud\,Rate\,Mode1,3 = \frac{2^{SMOD}}{32} X \frac{frekuensi\,oscilator}{12X[256-(TH1)]} \qquad (2.3)$$

Bila diperlukan *baud rate* yang rendah, maka interupsi t*imer 1* diaktifkan, dan menginisialisasi t*imer 1* sebagai *timer* 16 bit (*high nibble TMOD* = 0001B). Interupsi t*imer 1* ini digunakan untuk mengisi ulang nilai 16 bit pada TH1 dan TL1 secara software.

Tabel 2.5 menunjukkan daftar *baud rate* yang umum digunakan dan bagaimana cara menghasilkan *baud rate* tersebut dengan t*imer 1*.

Tabel 2.5 *Baud Rate* yang sering dipakai yang dihasilkan t*imer 1*

Baud rate	Fosc	SMOD	*Timer 1*		
			C/~T	*Mode*	Nilai isi ulang

Mode 0 Max:1 MHz	12 MHz	X	X	X	X
Mode 2 Max:375K	12 MHz	1	X	X	X
Mode 1&3 Max:62,5K	12 MHz	1	0	2	FFH
19,2K	11,059 MHz	1	0	2	FDH
9,6K	11,059 MHz	0	0	2	FDH
4,8K	11,059 MHz	0	0	2	FAH
2,4K	11,059 MHz	0	0	2	F4H
1,2K	11,059 MHz	0	0	2	E8H
137,5	11,986 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FBH

2.3.9 Pengendali Interupsi

Interupsi adalah kejadian atau kondisi yang menyebabkan penghentian sementara suatu program pada saat kondisi tersebut dilayani oleh program lain. Program yang menanggapi sebuah interupsi disebut rutin pelayanan interupsi atau *Interrupt Service Routine* (ISR) atau *Interrupt Handler*. ISR mengeksekusi tanggapan terhadap interupsi dan biasanya menyajikan operasi *input* atau *output* ke suatu piranti.

Ketika interupsi terjadi, program utama menghentikan eksekusi sejenak dan melompat ke ISR, ISR mengeksekusi, selanjutnya melaksanakan operasi dan akhirnya menghentikan operasi tersebut dengan perintah *Return from Interrupt*. Dengan instruksi terakhir tersebut maka program utama akan melanjutkan proses pelaksanaan instruksi pada urutan yang telah ditinggalkan.

Ada 2 unit register yang bertugas mengontrol interupsi, yaitu IE *(Interrupt Enable)* yang digunakan sebagai sumber interupsi yang dapat diaktifkan maupun dinonaktifkan secara individual dengan mengatur satu

bit SFR yang bernama IE. Sedangkan register yang kedua adalah IP *(Interrupt Priority)* digunakan sebagai sumber interupsi yang dapat diprogram per bit menjadi satu atau dua tingkat prioritas dengan mengatur bit pada SFR yang bernama IP.

Tabel susunan register IE dan IP dapat dilihat pada Tabel 2.6 dan tabel 2.7 sebagai berikut:

Tabel 2.6 Susunan Register IE

Simbol	Alamat	Keterangan(1=aktif,0=non aktif)
EA	Afh	Mengaktifkan / menonaktifkan intrupsi secara umum
-	AEh	Tidak terdefinisi
ET2	ADh	Mengaktifkan *Timer 2*
ES	Ach	Mengaktifkan interupsi kanal serial
ET1	ABh	Mengaktifkan interupsi *Timer 1*
EX1	AAh	Mengaktifkan interupsi eksternal 1
ET0	A9h	Mengaktifkan interupsi *Timer 0*
EX0	A8h	Mengaktifkan interupsi eksternal 0

Tabel 2.7 Susunan Register IP

Bit	Simbol	Alamat	Keterangan (1=aktif,0=non aktif)
IP.7	-	-	Tidak terdefinisi
IP.6	-	-	Tidak terdefinisi
IP.5	PT2	BDh	Prioritas interupsi *Timer 2*
IP.4	PS	BCh	Prioritas interupsi kanal serial
IP.3	PT1	BBh	Prioritas interupsi *Timer 1*
IP.2	PX1	Bah	Prioritas interupsi eksternal 1
IP.1	PT0	B9h	Prioritas interupsi *Timer 0*

IP.0	PX0	B8h	Prioritas interupsi eksternal 0

2.3.10 Pemrograman Mikrokontroler

Program pengendali mikrokontroler disusun atas kumpulan instruksi, instruksi yang setara dengan kalimat bahasa manusia yang hanya terdiri atas predikat dan obyek. Obyek dalam pemrograman mikrokontroler adalah data yang tersimpan di dalam memori, register dan *input/output*. Sedangkan kata kerja yang dikenal pun secara umum dikelompokkan menjadi perintah untuk perpindahan data, aritmetika, operasi logika, dan pengaturan alur program. Pada penyusunan program untuk alat yang penulis buat, menggunakan bahasa *assembler*.

Data bisa berada di berbagai tempat yang belainan, dengan demikian dikenal beberapa cara untuk pengalamatan data (*addressing mode*). Mode-mode pengalamatan data tersebut antara lain sebagai berikut.

2.3.10.1 Pengalamatan langsung *(direct addressing)*

Cara ini dipakai untuk menunjuk data yang berada dalam memori dengan cara menyebut alamat memori tempat data tersebut berada.

Contoh: MOV A, 7Fh

2.3.10.2 Pengalamatan tidak langsung *(indirect addressing)*

Cara ini dipakai untuk menunjuk data yang berada dalam memori, kalau memori penyimpan data ini letaknya berubah-ubah sehingga alamat memori tidak disebut secara langsung tapi melalui register (R0 atau R1).

Contoh: MOV A, @R1

2.3.10.3 Pengalamatan register *(register addressing)*

Cara ini digunakan untuk menunjuk data yang berada dalam register.

Contoh: MOV A, R7

2.3.10.4 Pengalamatan segera *(immediate constant)*

Cara ini digunakan untuk menunjuk data konstan yang berada di dalam instruksi.

Contoh: MOV A, #20h

Secara keseluruhan AT89S52 mempunyai sebanyak 255 macam instruksi, yang dikelompokan sebagai berikut:

1. Instruksi transfer data

Kode dasar untuk kelompok ini adalah **MOV**, singkatan *move* yang artinya memindahkan, meskipun demikian lebih tepat dikatakan perintah ini mempunyai makna peng-*copy*-an data.

Contoh lain perintah transfer data: **MOVC, MOVX, PUSH, POP, XCH, XCHD**

.

2. Instruksi aritmetika

Operasi aritmatika terdiri dari: penjumlahan, penambahan satu, pengurangan satu, perkalian, dan pembagian.

Contoh perintah ini adalah: **INC, DEC, SUBB, ADD, MUL, DIV**.

3. Instruksi logika

Kelompok perintah ini dipakai untuk melakukan operasi logika mikrokontroler MCS51, operasi logika yang bias dilakukan adalah AND (kode operasi **ANL**), OR (kode operasi **ORL**), dan *Exvlusive-OR* (kode operasi **XRL**).

4. Instruksi percabangan

Urutan pelaksanaan program dapat dikendalikan oleh instruksi percabangan bersyarat maupun tidak bersyarat.

Contoh instruksi percabangan: **ACALL, RET, AJMP, JNZ, JNC, DJNZ**.

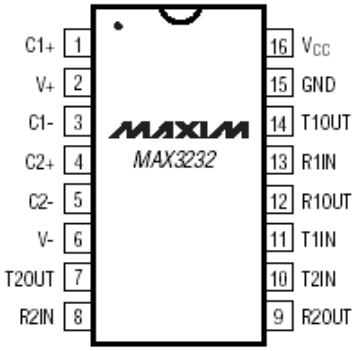
5. Kelompok operasi bit

Kelompok ini menangani instruksi yang banyaknya 1 bit. Register yang dipakai adalah register *carry*.

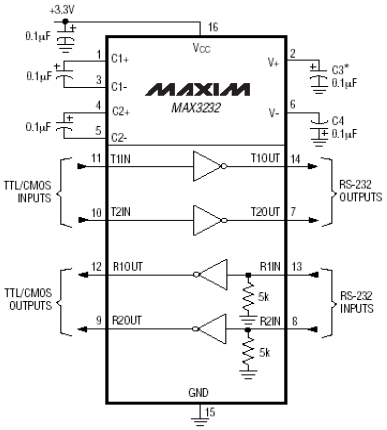
Contoh instruksi operasi bit: **CLR, SETB, CPL, JC, JBC, JNB**.

2.4 RS-232

RS232 merupakan metode pengubahan level TTL/CMOS ke level RS232. Contoh *IC* yang dapat digunakan adalah MAX-232 (lihat Gambar 2.16). *IC* ini terdiri atas dua *driver/receiver*, yang berisi penyesuai tegangan ke level tegangan EAI232 dari catu 5V. Setiap penerima mengubah masukan EIA232 ke level TTL/CMOS 5V. Gambar 2.15 menunjukkan rangkaian aplikasi *IC* MAX-232.



Gambar 2.16 *IC* MAX-232



Gambar 2.17 Contoh aplikasi *IC* MAX-232

2.5 Pemograman Borland Delphi

Borland delphi atau biasa di sebut Delphi merupakan sarana aplikasi *visual*. Bahasa pemograman yang digunakan adalah bahasa Pascal, Delphi merupakan pengembangan dari Turbo Pascal. Turbo Pascal yang diluncurkan pada tahun 1983 dirancang untuk sistem operasi DOS sedangkan Delphi dirancang untuk beroperasi dibawah sistem operasi windows. Delphi merupakan bahasa pemrograman yang menyediakan fasilitas untuk aplikasi dengan antarmuka *visual*

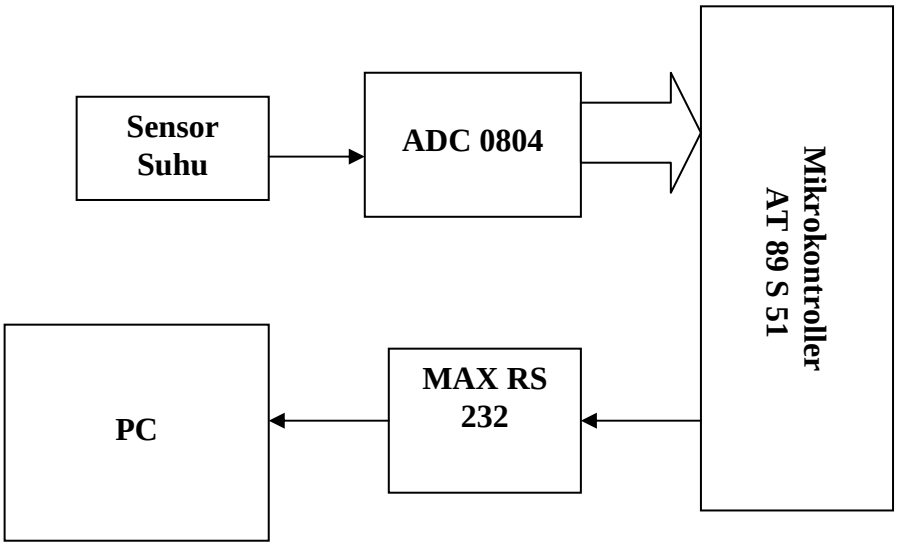
BAB III

PERANCANGAN ALAT

Perancangan dalam tugas akhir ini terdiri dari beberapa komponen – komponen yang akan dirangkaikan menjadi sebuah diagram blok yang membentuk sebuah sistem yang akan di rancang, adapun perancangan alat keseluruhan terdiri dari beberapa rangkaian utama yaitu : rangkaiaan sensor *temperature* lm 35, rangkaiaan ADC 0804, rangkaiaan *mikrokontroller*, dan rangkaian *converter serial* RS 232 serta didukung dengan perangkat lunak *(software)* berupa pemograman bahasa *assembly* dan pemograman bahasa delphi sebagai *interface* program

3.1 Diagram Blok

Dalam perancangan suatu alat diagram blok berfungsi untuk mempermudah dalam memahami suatu rangkaiaan. Adapun sistem secara keseluruhan, diagram blok dan hubungan tiap bagian dapat dilihata pada gambar 3.1 berikut



Gambar 3.1. Blok diagram rangkaiaan

Keterangan blok diagram

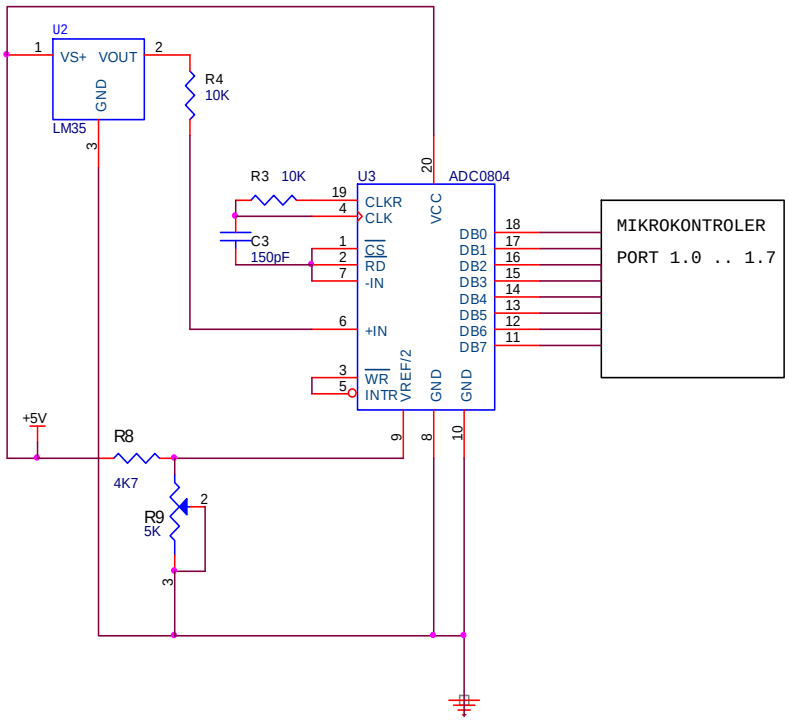
1. Sensor : Dimana rangkaiaan ini berfungsi sebagai alat untuk merubah besaran suhu menjadi besaran listrik
2. Rangkaian ADC : Rangkaian ini berfungsi untuk merubah data analog menjadi data digital
3. Rangkaian Mikrokontroller : Rangkaian ini berfungsi untuk megolah dan memproses data juga sebagai media penyimpan program *assembly*

4. Rangkaian RS 232 : Rangkaian ini berfungsi sebagai rangkaian *driver* komunikasi serial

5. Personal PC : mengolah dan memproses data yang diterima com serial dan di tampilkan di layar monitor, juga berfungsi sebagai media penyimpan program delphi

3.1 Sistem Elektonik

3.2.1 Rangkaian sensor Suhu dan ADC



Gambar 3.2 Rangkaiaan Suhu dan ADC

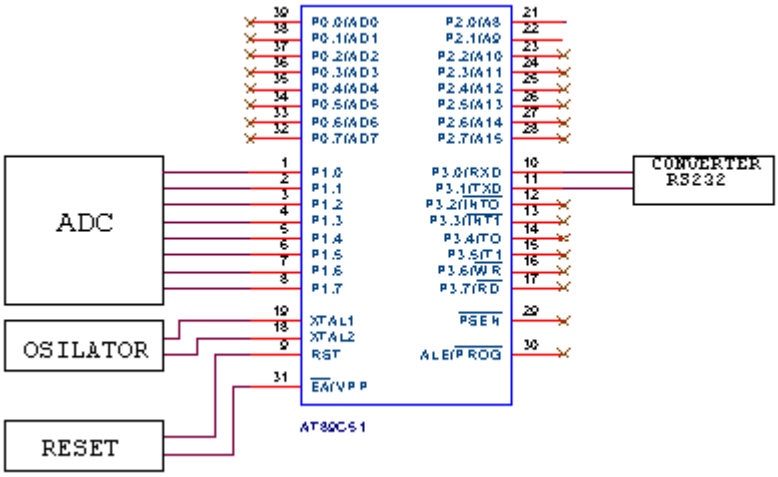
Dalam rancangan alat ini menggunakan sensor suhu IC LM 35 yang berfungsi merubah besaran suhu menjadi besaran listrik, IC LM 35 akan mendeteksi setiap berubahan suhu yang terjadi dan secara langsung akan menyampaikan kondisi tersebut kepada *Analog Digital Converter* (ADC) dan selanjutnya ADC akan Mengolah data yang diterima dari IC LM 35 untuk menampilkan perubahan dalam setiap derajat celciusnya. *Input* yang diterima ADC dari IC LM 35 berupa data analog dan kemudiaan ADC akan mengolah data tersebut sehingga *output* yang di hasilkan berupa data digital

Setiap perubahan temperatur sebesar 1°C, maka akan menghasilkan keluaran 10 mV dan perubahan ini adalah *linier*. Keluaran yang dihasilkan oleh LM 35 kemudian dimasukkan ke ADC ke pin +IN. Dengan *output* 1°C adalah sama dengan 10 mV maka dengan dikalikan 255, maka didapat tegangan referensinya adalah 2.55 V.

Proses yang terjadi di ADC adalah menggunakan prinsip *Successive Approximation* ADC yaitu dengan cara membandingkan tegangan yang masuk dengan DAC. DAC mendapatkan output dari *Successive approximation register*. Dengan pendekatan yang dilakukan adalah dengan dimulai dari bit yang terbesar atau MSB dengan bit 1, sampai mendapatkan biner yang sesuai dengan tegangan yang masuk. Setelah mendapatkan tegangan yang sesuai dengan tegangan yang masuk

maka data biner tersebutlah yang keluar dari ADC yang kemudian dikirimkan ke *mikrokontroler*.

3.2.2 Rangkaiaan Mikrokontroller



Gambar 3.3 Rangkaiaan Mikrokontroller

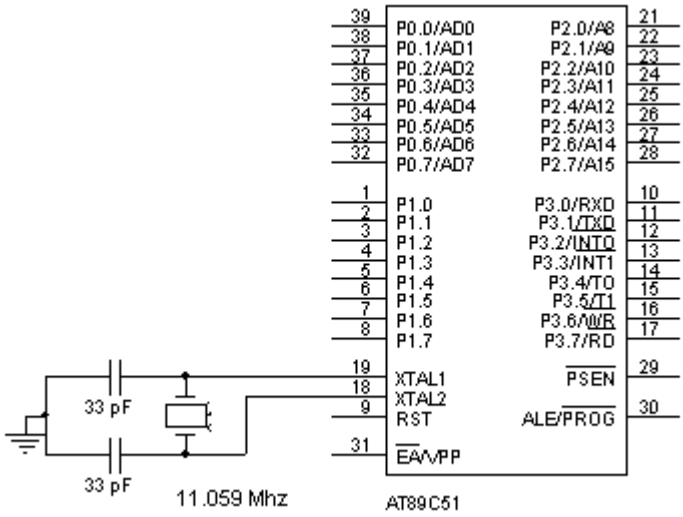
Dalam rancangan alat ini digunakan *mikrokontroller* Tipe AT89S51, *mikrokontroller* bertindak sebagai *procesor* yang mengendalikan maupun mengolah data serta sebagai media penyimpan bahasa *assembly*, hampir semua sistem ini terkendali pada bagian ini sehingga bagian ini dapat dikatakan sebagai pengendali utama dari sistem yang dirancang

Masukan yang *temperature* dari sensor suhu IC LM 35 keluarannya akan menghasilkan tegangan analog yang masuk ke ADC dan kemudiaan keluaranya akan dijadikan masukan ke *mikrokontroller*

melalui *port* 1.0 …..1.7. *Port* ini memang dapat digunakan sebagai masukan atau keluaran. Dalam sistem ini digunakan sebagai *port* masukan. *Mikrokontroller* akan memproses setiap perubahan data yang diterima dari ADC dan menyimpanya pada memori internal *mikrokontroller*, kemudian data yang telah di proses tersebut akan dikirim ke com serial sebagai jembatan komunikasi antara *mikrokontroller* dan PC

Jalur yang menghubungkan *mikrokontroler* dengan *converter* RS232 adalah melalui *port serial* input yaitu *port* 3.0 dan *serial output* adalah *port* 3.1. Dari *port* inilah tempat dimana terjadinya komunikasi serial antara *mikrokontroler* dan PC

Osilasi untuk clock di sini menggunakan kristal dengan frekuensi 11.059 MHz yang dihubungkan ke port XTAL1 (input) dan XTAL2 (output). Pemilihan kristal ini sudah didasarkan pada perhitungan untuk mendapatkan komunikasi *serial baud rate* 2400 bps.

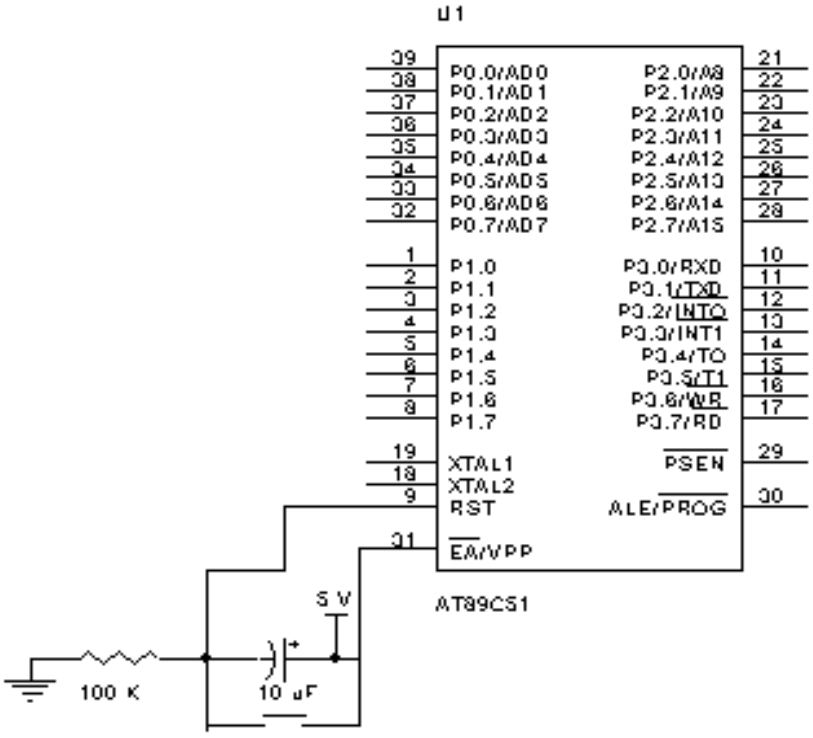


Gambar 3.4 Rangkaiaan Clock

Tombol reset dihubungkan melalui pin RESET (pin9) C1 dan R1 merupakan rangkaian power On Reset yang akan mereset rangkaian catu daya pertama kali dihidupkan. Hal ini penting untuk meyakinkan *mikrokontroler* akan bekerja dari awal program. Keadaan reset diperoleh apabila pin RESET pada *mikrokontroler* diberi logika tinggi dalam beberapa milidetik setelah catu daya dihidupkan. Hal ini diberikan untuk memberikan waktu pada rangkaian *osilator* agar mencapai keadaan stabil

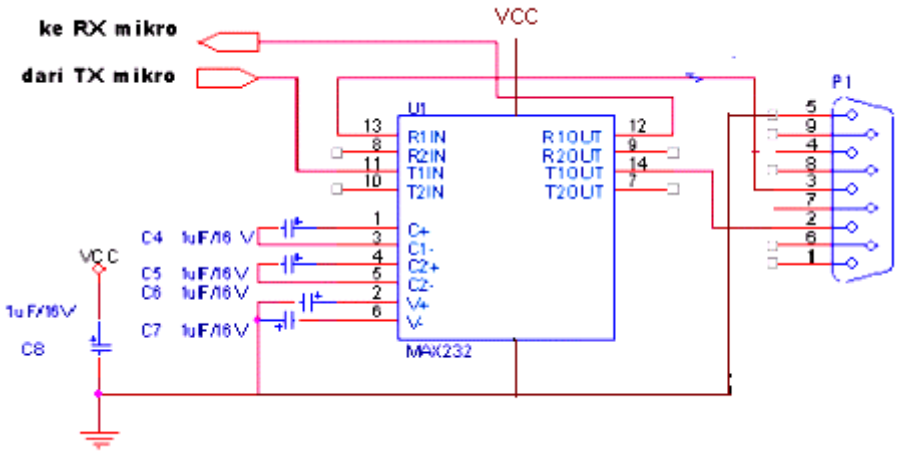
Cara kerja rangkaiaan reset pada saat catu daya diaktifkan, maka Kapasitor (C) sesuai dengan sifat kapasitor akan terhubung singkat pada saat itu Arus mengalir dari VCC langsung ke kaki RST sehingga kaki tersebut berlogika 1 , kemudiaan kapasitor (C) terisi hingga teganggan Vc

mencapai teganggan Vcc sehingga tegangan pada pin RST akan menjadi 0 dan pin RST berlogika 0. jika saklar di tekan, Reset bekerja secara manual aliran arus akan mengalir dari VCC melewati Resistor (R) sehingga Pin RST akan berlogika 1 dan apabila saklar di lepas maka aliran arus dari Vcc melewati Resistor (R) akan terputus sehingga logika pada Pin RST menjadi 0



Gambar 3.5 Rangkaiaan Reset

3.2.3 Rangkaian Driver RS 232



Gambar 3.6 Rangkaian Driver RS 232

Dalam rancangan ini digunakan IC MAX 232 sebagai *driver* komunikasi serial, rangkaiaan ini akan menerima data melalui *port* 3.1 mikrokontroller dan di hubungkan ke Pin 11 IC MAX 232 data yang diterima akan diolah oleh IC Max 232 menjadi data serial yang outputnya pin 14 akan di hubungkan dengan pin 2 konektor DB 9. melalui rangkaiaan inilah yang menjadi jembatan komunikasi serial antara *hardware* dan PC

3.3 Perancangan Software

Setelah merancang bagiaan *hardware* maka langkah selanjutnya adalah meranjang perangkat lunak atau *software* yang akan mendukung kerja dari *hardware*, karena tanpa *software hardware* tidak akan bekerja, dengan demikiaan *software* merupakan bagian penting dari sistem yang dirancang

Dalam sistem ini menggunakan 2 bahasa pemrograman yaitu bahasa *assembly* untuk *mikrokontroller* dan bahasa pemrograman delphi yang di gunakan sebagai *interface* antara Personal Computer dengan *hardware*

3.3.1 Bahasa assembly Mikrokontroller

Program Bahasa *assembly* disimpan dengan ekstensi H51 agar dapat di *download* ke *mikrokontroller* menggunakan produk HB 2000

Secara garis besar Langkah – langkah perancangan perangkat lunak (*software*) bahasa *assembly Mikrokontroller* adalah sebagai berikut:

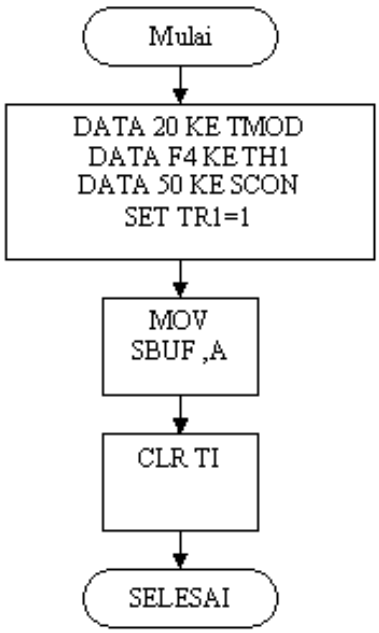
1. Pembuatan *flowchart* urutan pengendalian program
2. Pembuatan listing program dalam bentuk file berekstensi .H51
3. Compile file dengan ekstensi .OBJ
4. Program dimasukan ke EEPROM pada mikrokontroller AT89S51
5. Cek cara kerja rangkaiaan apakah sudah sesuai dengan harapan

3.3.1.1 Program Inisialisasi

Pada bagiaan ini adalah program inisialisasi sebelum program ini bekerja dengan baik. Dengan mengeset beberapa register maka akan mengatifkan com serial

Sebelum membuat program ini perlu diketahui beberapa instruction set sebagai berikut : TMOD, TH1, SCON. Penjelasan

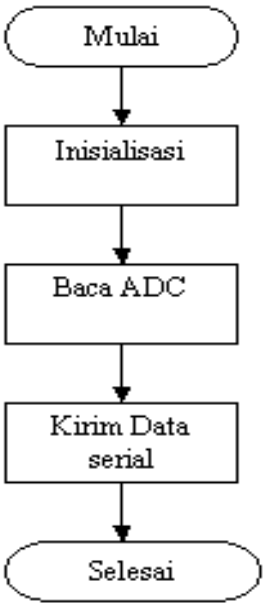
mengenai register tersebut sudah dijelaskan di awal. Dengan mengeset bit ke 5 dari TMOD dengan 1 dan yang lainnya adalah 0 (100000h atau 20h) yang berarti 8 bit *auto reload timer/counter*.



Gambar 3.7. Inisialisasi port serial

3.3.1.2 Program utama

Bagian ini menerangkan cara kerja dari program utama pemrograman bahasa *assembly*, pertama yang dilakukan adalah meginisialisasi *port serial* dan mengaktifkan ADC dengan cara memberikan aktif rendah (CLR) bit 2.7 (ADC_CS), untuk memulai konversi bit 2.5 (ADC_WR) diberi aktif rendah (CLR) sampai mendapatkan sampling data bit 2.5 (ADC_WR) diberi aktif tinggi (SET), setelah data didapatkan data akan di kirim melalui P.0 ke *mikrokontroller*, kemudiaan *mikrokontroller* akan mengirim data ke *port* 3.1



Gambar 3.8 Flowchart Interrupt Service Routine

berikut listing program bahasa assembly

Monitoring Temperature dengan Tampilan di PC

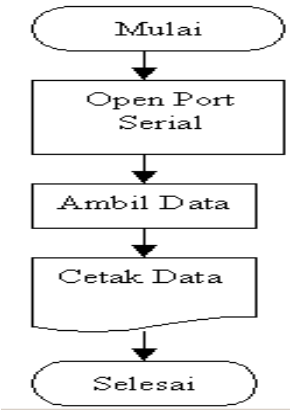
```
;=========================================
ADC_CS      bit     P2.7
ADC_RD      bit     P2.6
ADC_WR      bit     P2.5
ADC_INT     bit     P2.4
            org     0h
            acall init
next_sampling:
            clr     ADC_CS      ; aktifkan ADC0804
            clr     ADC_WR      ; start of conversion
```

```
        setb    ADC_WR
not_EOC:
        jb      ADC_INT,not_EOC
delay:  djnz    R2,$
        djnz    R3,delay
        clr     ADC_RD      ; Baca Data melalui P3
        djnz    R3,$
        mov     A,P0
        setb    ADC_RD
        setb    ADC_CS
        mov     p1,a
        acall   kirim
        sjmp    next_sampling
init:   mov     tmod,#20h       ;timer 1 mode 2
        mov     th1,#0f4h       ;isi th1 dgn data 0f4h untuk baud rate 2400
        mov     scon,#50h       ;mode serial 1 (8 bit uart)
        setb    tr1
        ret
kirim:  clr ti
        mov sbuf,a
        jnb ti,$
        ret
                end
```

3.3.1 Bahasa Pemrograman Borland delphi

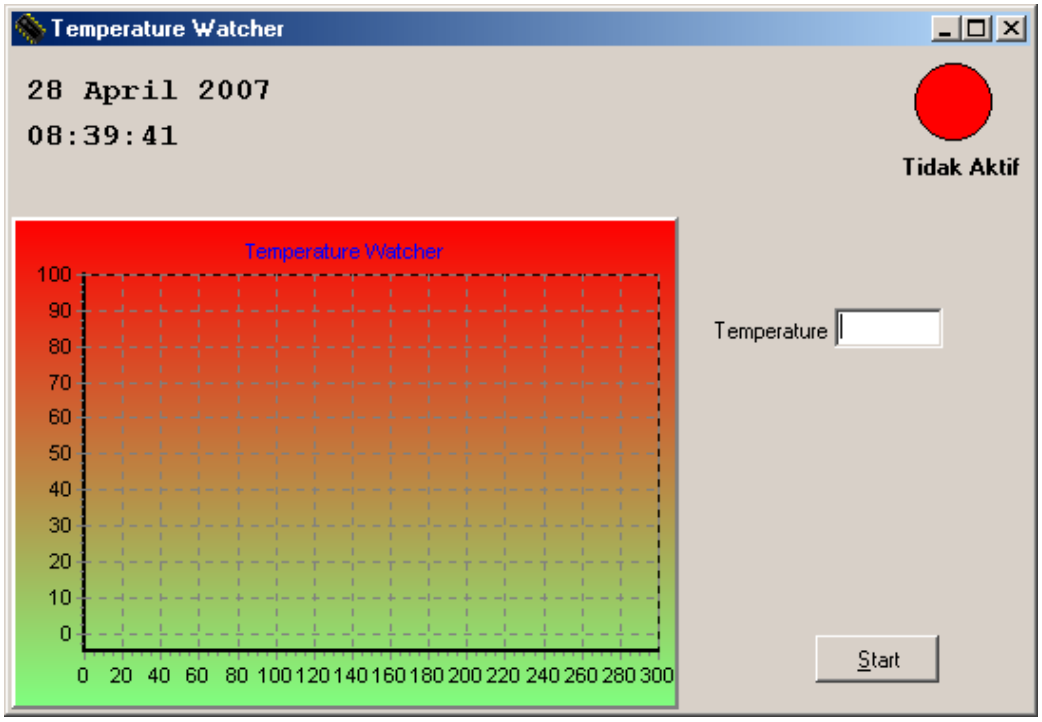
Borland delphi digunakan untuk menampilkan data masukan. Pada program ini data serial yang dikirim oleh *mikrokontroller* akan ditampilkan berupa tampilan grafik. Yang pertama kali di rancang adalah tabel – tabel basis data. Tabel – tabel ini dibuat dengan utilitas *database desktop software* bawaan borland delphi. Tabel - tabel ini akan ditunjuk oleh alias pada BDE *(borland database engine)*.dengan BDE ini program menampilkan basis data dibuat

Form monitoring *temperature* tersusun oleh beberapa komponen edit, label, chart, timer, speedbuton, shape dan c port. Komponen yang utama pada form ini adalah komponen c port. C port adalah komponen data untuk menerima dan menampilkan data serial dari RS 232. komponen ini dilengkapi dengan fitur seting komunikasi serial seperti penggunaan *port, baundrate, databits, stopbits ,farity* dan *flow* kontrol. Penggunaannya sama dengan komponen lain, yaitu dengan mengaktifkan komponen tersebut, kemudiaan menangani kejadiaan pada *events* nya



Gambar 3.9 Flow Chart Pemrograman Borlan Delphi

3.3.2.1 Tampilan Program



Gambar 3.10 Tampilan di Pc

dalam perancangan tampilan interface di komputer harus user friendly, sederhana dan user bisa langsung mengerti penggunaanya

BAB IV

PENGUJIAAN ALAT

Pada pembuatan alat ini telah dilakukan pengujiaan pada alat – alat yang telah dibuat diantaranya adalah pengujiaan rangkaiaan pada sensor suhu LM 35, pengujiaan sensor suhu LM 35 dengan *Analog Digital Converter (ADC)* Mikrokontroler AT89C51, pengujian rangkaiaan *converter* RS 232, pengujiaan rangkaiaan keseluruhan dengan tampilan grafik di PC

4.1 Pengujiaan Sensor Suhu LM35

Pengujian ini dilakukan dengan cara pengambilan data berupa suhu yang dibandingkan dengan sensor suhu LM 35 dan suhu aktual dengan menggunakan *thermometer air raksa dengan resolusi 1 derajat celcius.*

Cara pengambilan data ini dilakukan dengan cara meletakan sensor suhu LM 35 dengan *thermometer* di suatu media yang akan di ukur besar temperaturnya maka di catatlah data suhu aktual dari thermometer dan sensor LM 35 dengan cara mencatat output tegangannya, pengambilan data dilakukan sebanyak 10 kali dengan suhu yang berbeda – beda

Tabel 4.1 Pengujiaan sensor suhu LM 35

N0	Penunjukan Thermometer	Output LM 35 (mVolt)
1	35	354
2	40	406
3	45	457
4	50	506
5	55	554
6	60	603
7	65	657
8	70	704
9	75	756
10	80	804

4.2 Pengujiaan Sensor Suhu Lm 35 Dan ADC

Pengujian ini dilakukan hampir sama dengan pengujian yang telah diuraikan di atas hanya dengan menambahkan output yang keluar dari ADC 0804. Data yang keluar dari ADC adalah berupa 8 bit biner yang akan di konversi dengan bilangan hexa. Tegangan analog yang masuk ke ADC diukur dengan multimeter digital dengan resolusi 1 mV.

Tabel 4.2 Pengujian sensor suhu dengan ADC

No	Penunjukan Thermometer	Output LM 35 (mV)	Output ADC (Hexa)
1	35	354	23
2	40	406	28
3	45	457	2D
4	50	506	32
5	55	554	37
6	60	603	3C
7	65	657	41
8	70	704	46

9	75	756	4B
10	80	804	50

4.3 Pengujian Mikrokontroller

Pengujiaan ini dilakukan dengan cara membuat program mikrokontroller sederhana untuk menyalakan lampu led pada Port 1

Tabel 4.3 Pengujiaan Mikrokontroller

No	Intruksi	P.7	P.6	P.5	P.4	P.3	P.2	P.1	P.0
1	Mov P1,#0FFh	0	0	0	0	0	0	0	0
2	Mov P1,#00h	1	1	1	1	1	1	1	1
3	Mov P1,#0fdh	0	0	0	0	0	0	1	0
4	Mov P1,#0F0h	0	0	0	0	1	1	1	1
5	Mov P1,#0Fh	1	1	1	1	0	0	0	0
6	Mov P1,#0FEh	0	0	0	0	0	0	0	1
7	Mov P1,#23h	1	1	0	1	1	1	0	0
8	Mov P1,#55h	1	0	1	0	1	0	1	0
9	Mov P1,#AAh	0	1	0	1	0	1	0	1
10	Mov P1,#BEh	0	1	0	0	0	0	0	1

4.4 Pengujian Rangkaiaan Converter Rs 232

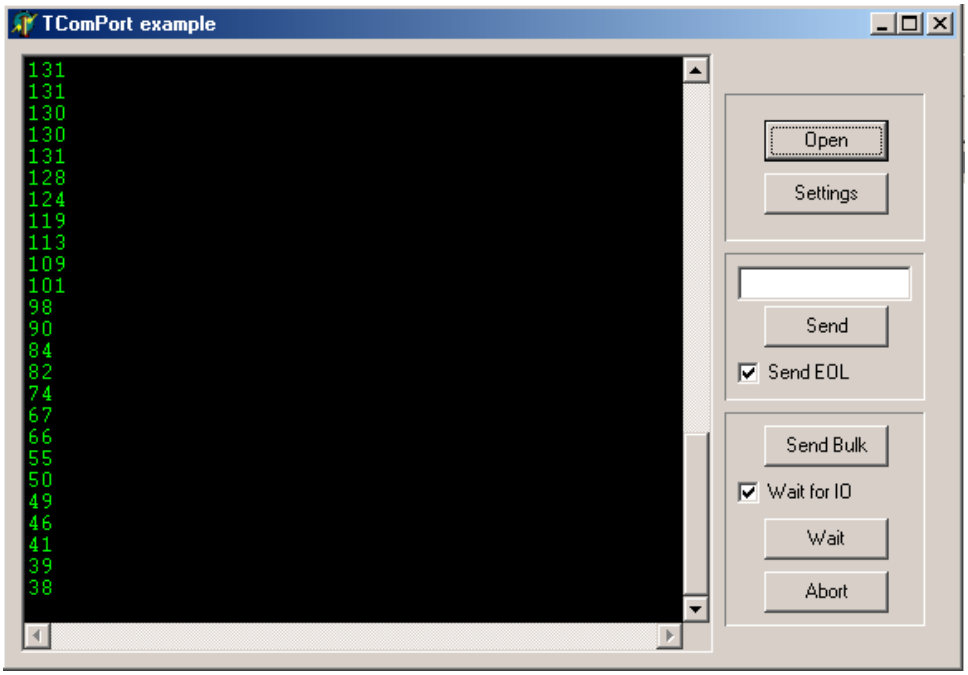
Pengujiaan rangkaiaan RS 232 ini menggunakan software comtest, pada pengujiaan ini sensor suhu diganti dengan potensiometer 5 K ohm ini bertujuaan agar pengujiaan dapat dilakukan dengan maksimal

Tabel 4.4 Pengujian RS 232

NO	Output Potensiometer (mVolt)	Tampilan pada comtest
1	0	0
2	114	10
3	232	20
4	333	30
5	437	40
6	544	50

7	639	60
8	764	70
9	860	80
10	2890	255

Tampilan pada Software Comtest



Gambar 4.1 Tampilan Comtest

4.5 Pengujiaan keseluruhan

Pengujiaan ini meliputi pengujiaan suhu actual, pengukuran output Lm 35 dan tampilan grafik dan display pada PC pada pengujian ini juga disertakan error kesalahan antara suhu actual dan tampilan display dengan rumus

Error = (A – B)/ A x 100%

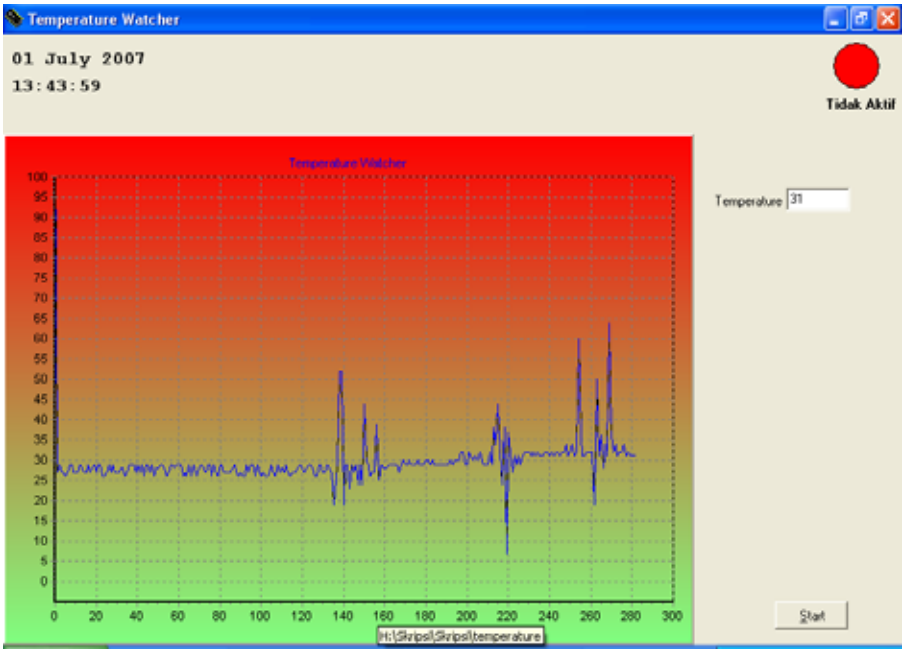
Dimana : A = suhu actual
B = Tampilan display

Table 4.5 Pengujiaan keseluruhan

no	Suhu Actual	Output LM 35 (m Volt)	Tampilan display	Error (%)
1	30	314	29	0.03
2	35	362	34	0.02
3	40	395	39	0.025
4	45	445	46	0.022
5	50	508	49	0.02
6	55	560	54	0.018
7	60	618	58	0.03
8	65	667	64	0.015
9	70	718	69	0.014
10	80	813	79	0.012

Berdasarkan hasil pengujiaan dan spesifikasi alat ini mampu mengukur dengan range antara 20 ° celcius sampai 100 ° celcius

Tampilan pengujiaan grafik pada PC



Gambar 4.2 Tampilan Pengujiaan Grafik

Keterangan Gambar :

Sumbu Y : Besarnya suhu dalam derajat Celcius

Sumbu X : Waktu (perubahan grafik setiap 0.26 detik)

Warna Merah pada grafik : Suhu tinggi (panas)

Warna Hijau pada grafik : Suhu rendah (dingin)

BAB V

PENUTUP

5.1. Kesimpulan

Setelah selesainya tahap pembahasan, pembuatan dan pengujian sistem/alat pada Tugas Akhir ini, maka didapat beberapa kesimpulan :

1. Berdasarkan hasil pengujian keseluruhan, suhu yang dideteksi oleh sensor dapat tampil pada PC dengan resolusi setara termometer air raksa, yaitu 1 derajat celcius.
2. Suhu yang ditampilkan di PC berupa angka dan grafik.

5.2 Saran - saran

Untuk pengembangan lebih lanjut dari tugas akhir ini, maka disampaikan beberapa saran sebagai berikut :

1. Untuk mendapatkan hasil pengukuran yang lebih komprehensif dianjurkan untuk menggunakan sensor suhu yang lebih banyak disesuaikan dengan ukuran ruangan.
2. Dalam implementasi selanjutnya dapat juga di gunakan teknologi nirkabel sehingga monitoring suhu dapat di pantau dari jarak yang sangat jauh.

DAFTAR PUSTAKA

1. Jacob. J.Michael 1985. Industrial Control Electronic. Singapore : Pratice Hall International ,INC
2. Malvino, Hanapi Gunawan 1996. Prinsip – prinsip electronika. Jakarta, Erlangga
3. Paulus Andi Nalwan 2003. Panduaan Praktis Teknik Antarmuka dan Pemrograman Mikrokontroller AT89C51. Jakarta, PT Elex Media Komputindo
4. Sudjadi, 2005, Teori dan Aplikasi Mikrokontroller. Yogyakarta : Graha Ilmu
5. Dwi.Sutadi, 2003. I/O Bus & Motherboard, Yogyakarta, Andi
6. Alam,Magus, 2003. Mengelola Data dengan Borland Delphi, Jakarta, Gramedia
7. Situs Internet, www. Atmel.com
8. Situs Internet, www.Jasakom.org
9. Situs Internet, www.Benkelprogram.com