

**MILIK**  
PERPUSTAKAAN-FTI-UII  
YOGYAKARTA

**MONITOR/LOGGER TEMPERATUR RUANGAN  
BERBASIS AT89C51 DENGAN VISUALISASI VISUAL  
BASIC**

**TUGAS AKHIR**

Diajukan sebagai Salah Satu Syarat untuk  
Memperoleh Gelar Sarjana Teknik Elektro



No. Inv	100/FTI/TE-UII/05.
Tanggal	29 April 05.
Asal	F. TEKNOLOGI INDUSTRI - UII
Harga	<del>RP</del> Rp 500,-
PERPUSTAKAAN FAK. TEKNOLOGI INDUSTRI UNIVERSITAS ISLAM INDONESIA YOGYAKARTA	

Oleh :

Nama : Median Akbar

No. Mahasiswa : 99 524 117

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
JOGJAKARTA  
2005**

# LEMBAR PENGESAHAN PEMBIMBING

## MONITOR/LOGGER TEMPERATUR RUANGAN BERBASIS AT89C51 DENGAN VISUALISASI VISUAL BASIC

### TUGAS AKHIR

Oleh :

Nama : Median Akbar

No.Mhs : 99 524 117

Jogjakarta, Januari 2005

Pembimbing I,



( Ir.Hj. Budi Astuti )

Pembimbing II,



(RM.SisdarmantoAdinandra, ST )

**LEMBAR PENGESAHAN PENGUJI**  
**MONITOR/LOGGER TEMPERATUR RUANGAN BERBASIS AT89C51**  
**DENGAN VISUALISASI VISUAL BASIC**

**TUGAS AKHIR**

Oleh :

Nama : Median Akbar

No.Mhs : 99 524 117

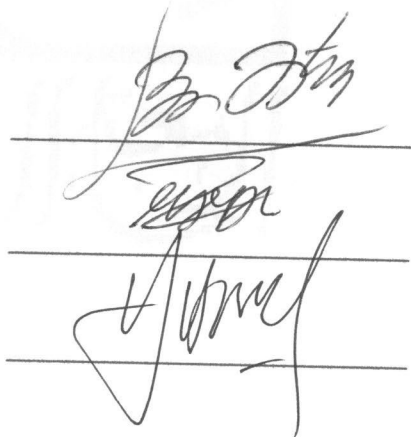
Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat  
untuk Memperoleh Gelar Sarjana Jurusan Teknik Elektro  
Fakultas Teknologi Industri Universitas Islam Indonesia  
Jogjakarta, Februari 2005

Tim Penguji

Ir. Hj. Budi Astuti

RM. Sisdarmanto Adinandra, ST

Yusuf Aziz Amrullah



Mengetahui,

Dekan Fakultas Teknologi Industri  
Universitas Islam Indonesia

( Ir. H. Bachrun Sutrisno, MSc )



## HALAMAN PERSEMBAHAN

### *Aku Persembahkan Untuk*

*Gusti Allah SWT  
Kanjeng Nabi Muhammad saw*

*(Alm/ah) Kedua Kakek dan Nenekku  
Bapak dan Ibu*

*“Terima kasih atas Do’a, Kasih Sayang dan Kepercayaan  
yang Tiada Henti”*

*Kakak-kakak dan Adikku*

*“Terima Kasih untuk Kata Semangat yang Terus Kalian  
Ucapkan”*



## MOTTO

Bacalah dengan nama Tuhan-Mu yang menciptakan. Dia telah menciptakan manusia dari segumpal darah. Bacalah dan TuhanMulah Yang Maha Pemurah. Dia mengajar dengan kalam. Dialah yang mengajar manusia segala yang tidak diketahui.”

**(QS. Al-Alaq : 1-5)**

“Bersabarlah menjalani segala hal yang tak terpecahkan dalam hatimu dan cobalah mencintai pertanyaan-pertanyaan. Hidupkan senantiasa pertanyaan-pertanyaan itu. Perlahan, tanpa perlu mencermatinya, mungkin engkau akan sampai pada jawaban yang engkau cari”

**(Rainer Maria Rilke 1875 – 1926)**

Di saat engkau merasa paling lemah diantara yang lemah. Tidakkah kau merasakan Cinta-Ku

**(anonim)**

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum Wr.Wb.

Puji Syukur kehadiran Allah SWT yang telah memberikan rahmat, taufiq dan hidayah-Nya, karena ridho-Nya lah penulis dapat menyelesaikan tugas akhir dengan judul “Monitor/Logger Temperatur Ruang Berbasis AT89C51 dengan visualisasi Visual Basic”. Sholawat dan salam pun tak lupa penulis haturkan kepada Nabi besar Rasulullah Muhammad saw beserta para keluarga dan sahabatnya.

Adapun maksud dari penyusunan tugas akhir ini adalah untuk memenuhi kurikulum S-1 Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia. Disamping itu untuk menambah pengalaman dan pengetahuan dan juga sebagai perbandingan ilmu-ilmu yang telah diperoleh dibangku kuliah secara teoritis dengan keadaan yang sebenarnya yang ada di dunia industri.

Dalam menyusun tugas akhir, penulis mengucapkan terima kasih berkat dukungan dan bantuan dari berbagai pihak. Untuk itu pada kesempatan ini penulis mengucapkan banyak terima kasih yang sebesar-besarnya kepada :

1. Ibu Ir. Hj. Budi Astuti, selaku Dosen Pembimbing I sekaligus Ketua Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia.
2. Bapak RM. Sisdarmanto Adinandra, ST, selaku Dosen Pembimbing II sekaligus sebagai Kordinator Laboratorium Kontrol dan Instrumentasi pada Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas

Islam Indonesia. Terima Kasih atas kesabarannya dan bimbingannya dan juga saran serta kritiknya yang telah banyak membantu penulis menyelesaikan tugas akhir ini.

3. Bapak dan Ibuku yang telah memberikan dukungan moril, materi, juga doa yang tidak henti-hentinya. Terima kasih atas kepercayaannya selama ini.
4. Kakak-kakakku dan adik-adikku yang banyak memperhatikanku dan tak lupa terus meneriakkan kata semangat. "Terima Kasih".
5. Segenap Dosen di lingkungan Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia. Terima kasih atas bimbingan dan ilmunya selama ini.
6. Segenap rekan-rekan yang tidak dapat penulis sebutkan satu persatu yang banyak memberikan bantuan.

Penulis menyadari bahwa Tugas Akhir ini tidak lepas dari kesalahan dan kekurangan dalam penulisan ataupun bentuk lainnya. Oleh karena itu, kritik dan saran yang membangun akan penulis terima dengan senang hati.

Akhirnya, harapan penulis semoga Tugas Akhir ini dapat bermanfaat bagi kita semua, Sekian dan Terima Kasih.

Wassalamu'alaikum Wr.Wb

Jogjakarta, Januari 2005

Penulis

Median Akbar

## ABSTRAK

Perkembangan teknologi elektronika yang semakin maju mengarah ke perkembangan mikrokontroler. Mikrokontroler sendiri hadir sebagai terobosan teknologi baru dari mikroprosesor dan mikrokomputer. Berbagai aplikasi banyak menggunakan mikrokontroler diantaranya Sistem Akuisisi Data. Sistem Akuisisi Data digunakan untuk aktivitas pemantauan dan pencatatan suatu kondisi pada suatu sistem atau lingkungan secara otomatis tanpa campur tangan manusia. Sistem Akuisisi Data biasanya dibangun dengan PLC dan diperuntukkan industri berskala besar. Pada sistem yang lebih kecil diperlukan teknologi akuisisi data yang lebih sederhana namun dengan kualitas yang baik. Mikrokontroler dapat digunakan untuk aplikasi akuisisi data yang memerlukan titik pengamatan yang lebih sedikit.

Aplikasi sistem pemantauan Monitor/Logger temperatur ruangan yang di buat ini pun merupakan salah satu contoh penggunaan teknologi mikrokontroler yaitu AT89C51 dari Atmel. Sistem pemantauan ini akan membaca data-data berupa temperatur ruangan dari sensor-sensor suhu LM35, data-data tersebut akan dilewatkan pada chip ADC 0809 yang berfungsi mengubah sinyal analog menjadi sinyal digital. Mikrokontroler akan membaca data digital tersebut dari ADC dan mengirimkannya ke *serial* port dari PC (Personal Komputer). Program Monitoring/Logger akan membaca data tersebut, melakukan pemilahan data tiap sensornya dan menampilkannya dan dalam selang waktu 2 detik melakukan perekaman data ke dalam file..

Sistem Monitor/Logger bertujuan untuk melakukan aktivitas pemantauan dan pencatatan temperatur dari berbagai sensor yang ada ke sebuah *file database*. Sistem Monitor/Logger dapat berkerja pada jangkauan suhu antara 2°C - 150°C. Selain itu sensor dapat diletakkan sejauh 30 cm dari modul sensor dan 60 cm untuk kabel serial dari modul mikrokontroler ke PC. Dengan begitu sistem Monitor/Logger ini dapat banyak berguna untuk sistem-sistem kecil yang membutuhkan pengamatan temperatur ruangan semisal peternakan, laboratorium dan sebagainya.

## DAFTAR ISI

HALAMAN JUDUL .....	i
LEMBAR PENGESAHAN PEMBIMBING .....	ii
LEMBAR PENGESAHAN PENGUJI .....	iii
HALAMAN PERSEMBAHAN .....	iv
HALAMAN MOTTO .....	v
KATA PENGANTAR .....	vi
ABSTRAKSI .....	viii
DAFTAR ISI .....	ix
DAFTAR GAMBAR .....	xi
DAFTAR TABEL .....	xii
BAB I PENDAHULUAN	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah Tujuan Penelitian .....	2
1.4 Sistematika Penulisan .....	3
BAB II LANDASAN TEORI	
2.1 Mikrokontroler AT89C51 .....	4
2.1.1. Arsitektur Mikrokontroler AT89C51 .....	4
2.1.2. Deskripsi Pin AT89C51 .....	6
2.1.3. Struktur Memori Mikrokontroler AT89C51 .....	8
2.1.3.1. Ram Internal .....	9
2.1.3.2. Special Function Register .....	9
2.1.3.3. Flash PEROM .....	13
2.1.4. Mode Pengalamatan AT89C51 .....	14
2.1.5. Mengatur Alur Program .....	15
2.1.6. Operasi Serial Port .....	15
2.2. Analog Digital Converter (ADC) 0809 .....	19
2.3. Sensor Suhu LM35 .....	22

BAB III PERANCANGAN SISTEM	
3.1 Rancangan Sistem .....	24
3.1.1. Perancangan Modul Sensor Temperatur .....	24
3.1.2. Perancangan Modul ADC 0809 .....	27
3.1.3. Perancangan Modul AT89C51 .....	29
3.1.4. Rangkaian Catu Daya .....	31
3.1.5. Komunikasi Port Serial AT89C51 dan PC .....	31
3.2. Prototype Sistem Monitor/Logger .....	32
3.3. Rancangan Program .....	33
3.3.1. Program Mikrokontroller .....	33
3.3.2. Program Monitoring/Logger dengan VB .....	41
BAB IV ANALISIS DAN PEMBAHASAN	
4.1 Pengukuran .....	44
4.1.1. Pengukuran Terhadap Output Sensor Suhu .....	44
4.1.2. Pengukuran Terhadap Output Penguat .....	46
4.1.3. Pengukuran Terhadap Output ADC 0809 .....	47
4.1.4. Pengujian Terhadap Output Program .....	49
4.2. Pembahasan .....	52
BAB V PENUTUP	
5.1 Kesimpulan .....	54
5.2 Saran .....	54
DAFTAR PUSTAKA .....	55
LAMPIRAN .....	56

## DAFTAR GAMBAR

Gambar 2.1	Konfigurasi Pin AT89C51	5
Gambar 2.2	Arsitektur AT89C51	5
Gambar 2.3	RAM Internal dan Flash PEROM	8
Gambar 2.4	Diagram Blok Port Serial	16
Gambar 2.5	Komunikasi UART	16
Gambar 2.6	Register SCON	17
Gambar 2.7	Mode 1 Operasi Serial	18
Gambar 2.9	Pin ADC 0809	20
Gambar 2.10	Diagram Blok ADC 0809	20
Gambar 2.11	Diagram Pewaktuan ADC 0809	22
Gambar 2.12	IC LM35DZ	23
Gambar 3.1	Diagram Blok Sistem	24
Gambar 3.2	Rangkaian Sensor LM35 dengan Penguat	26
Gambar 3.3	Rangkaian Modul ADC 0809	28
Gambar 3.4	Skema Single Chip Mikrokontroller AT89C51	29
Gambar 3.5	Modul Development System DST 51	30
Gambar 3.6	Rangkaian Power Supply 5 volt	31
Gambar 3.7	Rangkaian antar muka RS-232	32
Gambar 3.8	<i>Prototype</i> Sistem Monitor/Logger	33
Gambar 3.9	Diagram Alir Program pengambilan data Temperatur	35
Gambar 3.10	Diagram alir program monitoring/logger	41
Gambar 4.1	Rangkaian Sensor Suhu LM35	44
Gambar 4.2	Grafik Suhu Terhadap Tegangan Ouput LM355	45
Gambar 4.3	Grafik Suhu terhadap output Tegangan Penguat	47
Gambar 4.4	Data Suhu dalam Heksa	48
Gambar 4.5	Program Temperature Monitor/Logger	51

## DAFTAR TABEL

Tabel 2.1	Mode Operasi Serial	18
Tabel 2.2	Tabel alamatADC 0809	21
Tabel 4.1	Hasil Pengamatan Pada Sensor Suhu	45
Tabel 4.2	Hasil Pengamatan Pada Sensor Suhu dan OP-AMP	46
Tabel 4.3	Hasil Pengamatan Suhu Terhadap Tegangan Keluaran Sensor, Tegangan OP-AMP dan ADC	49
Tabel 4.4	Hasil Pengamatan pada Suhu dan Program Monitor	51





## BAB I PENDAHULUAN

### 1.1. LATAR BELAKANG

Pesatnya perkembangan piranti elektronik berimbas pula pada teknik pengukuran. Teknologi piranti elektronika yang berkembang diantaranya adalah mikrokontroller. Dengan sebuah komponen elektronika yang dapat bekerja sesuai dengan program yang diisikan ke dalam memorinya, bermacam-macam aplikasi dapat dibuat, diantaranya sistem pemantauan. Saat ini berbagai macam teknologi baru telah diciptakan, sebagian besar digunakan pada proses industri. PLC salah satunya. Harga yang ditawarkan teknologi ini memang mahal, namun sebanding dengan kemampuan sistem dan jangkauan yang dapat ditanganinya. Namun untuk kasus-kasus tertentu dimana sistem yang ditangani sederhana, harga tentu menjadi salah satu pertimbangan.

Seringkali dibutuhkan pemantauan temperatur yang tidak terbatas pada satu ruangan saja. Semisal dalam suatu bangunan terdapat beberapa ruangan yang hendak diamati temperaturnya maka dengan menggabungkan beberapa *tools* seperti mikrokontroler, *personal computer* (PC) dan *software* tambahan, sebuah sistem pemantauan kondisi suatu tempat dalam hal ini temperatur sudah dapat dibuat dengan biaya minimal namun dengan tingkat presisi dan kehandalan yang cukup baik.

## 1.2. RUMUSAN MASALAH

Dari latar belakang diatas maka dapat dirumuskan permasalahan sebagai berikut yaitu bagaimana membuat sistem monitor/logger temperatur ruangan berbasis mikrokontroller AT89C51 dengan visualisasi Visual Basic.

## 1.3. BATASAN MASALAH

Batasan – batasan disini meliputi:

1. Dalam penelitian difokuskan pada perancangan dan pembuatan sensor temperatur ruangan berbasis mikrokontroller AT89C51 dan program *monitoring* temperatur ruangan menggunakan Visual Basic.
2. Obyek pemantauan dibatasi hingga empat ruangan dalam satu bangunan.
3. Rangkaian sensor yang dibuat bekerja pada suhu antara 2°C hingga 150°C.
4. Rangkaian sensor yang dibuat berkerja pada jarak 30 cm dengan tambahan 60 cm untuk komunikasi serialnya.

## 1.4. TUJUAN PENELITIAN

Tujuan penelitian ini adalah untuk membuat sistem monitor temperatur ruangan berbasis AT89C51 dengan Visual Basic dengan kemampuan menyimpan informasi ke dalam *database* berupa *file*.

## **1.5. SISTEMATIKA PENULISAN**

Sistematika penulisan tugas akhir ini terdiri dari V bab, dengan masing-masing bab adalah sebagai berikut :

### **BAB I Pendahuluan**

Bab ini berisi tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan penulisan, dan sistematika penulisan.

### **BAB II Landasan Teori**

Bab ini memuat teori-teori yang berhubungan dengan penulisan dan pembuatan Mesin ini. Dan juga berisikan tentang teori-teori yang berhubungan dengan sistem.

### **BAB III Perancangan Sistem**

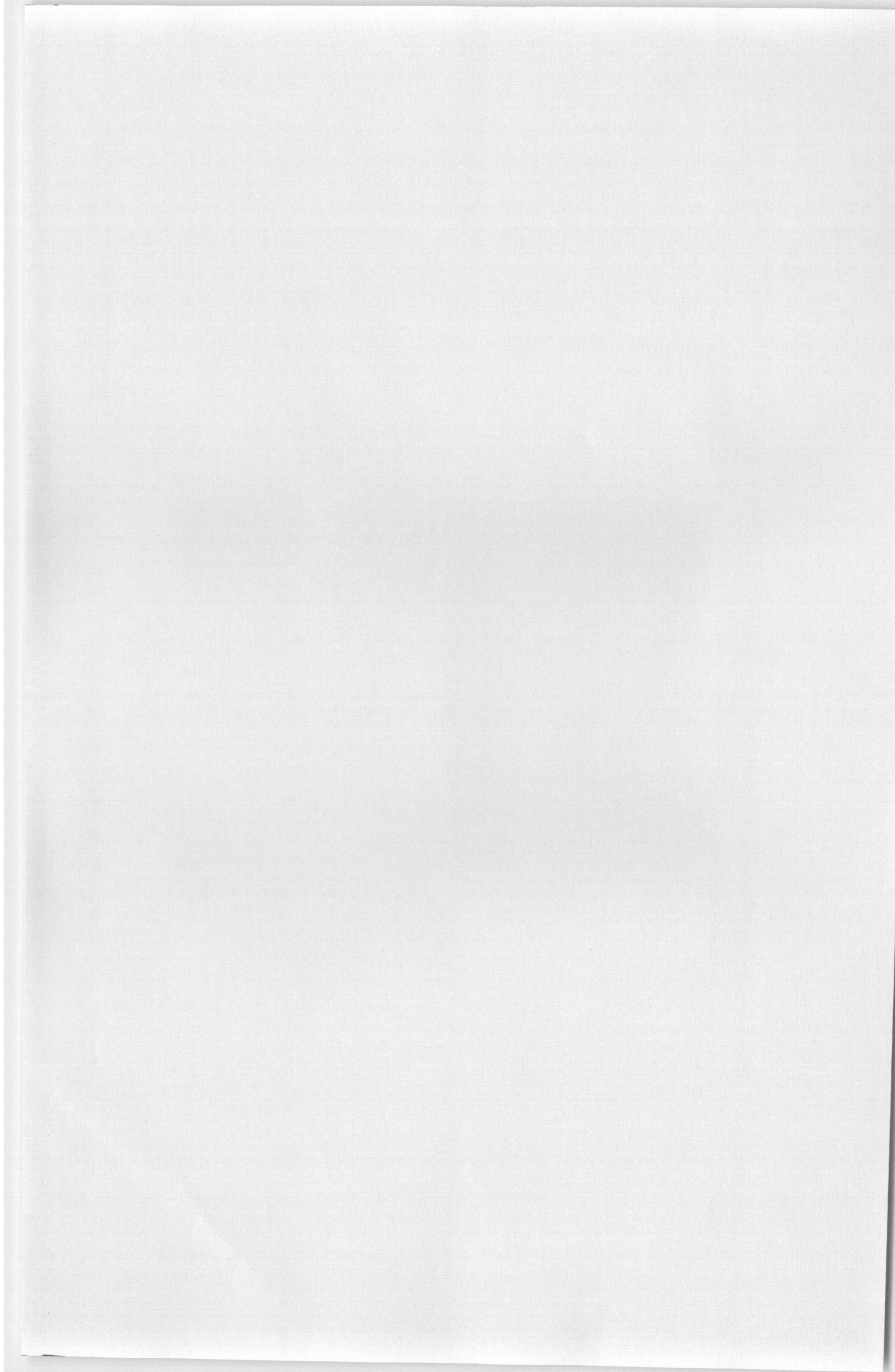
Bab ini menjelaskan metode-metode perancangan yang digunakan, cara mengimplementasikan rancangan dan pengujian sistem yang telah dibuat serta batasan dan hambatan yang ditemui selama proses perancangan dan implementasi sistem.

### **BAB IV Analisis dan Pembahasan**

Bab ini membahas tentang hasil dari sistem yang dibuat dan unjuk kerjanya kemudian dibandingkan dengan dasar teori.

### **BAB V Penutup**

Bab ini memuat kesimpulan dan saran-saran dari proses perancangan, dan juga hasil pengujian dari sistem.





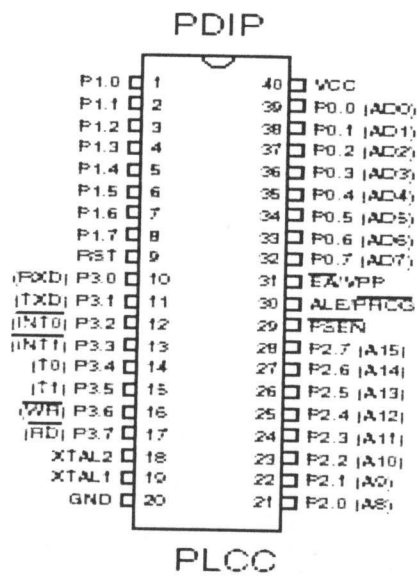
## BAB II

### LANDASAN TEORI

#### 2.1. MIKROKONTROLLER AT89C51

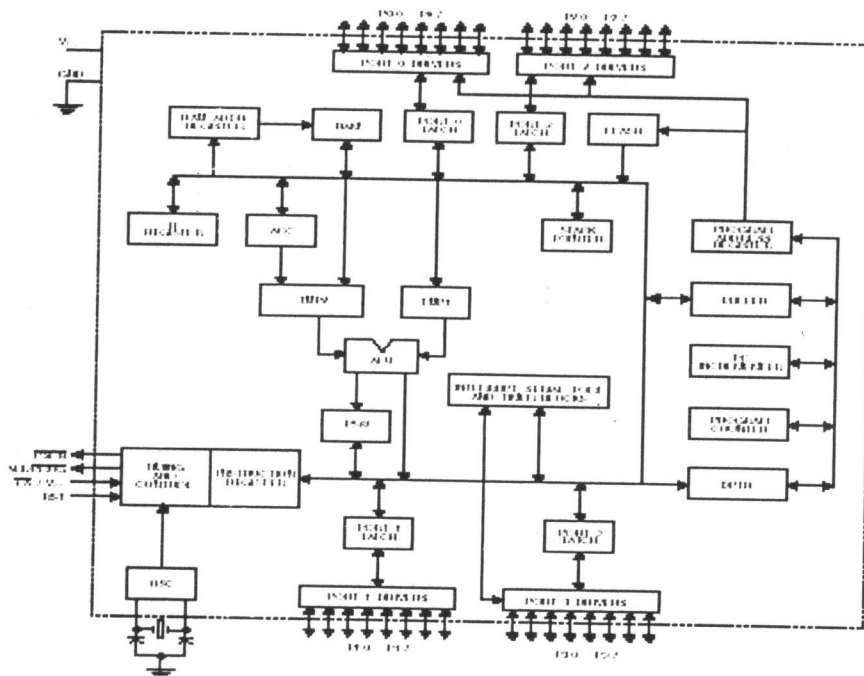
##### 2.1.1. Arsitektur Mikrokontroller AT89C51

Mikrokontroller merupakan komponen elektronika berbentuk chip (IC) yang dapat diprogram sehingga dapat digunakan sebagai chip pengontrol elektronis yang cerdas seperti *processor*. AT89C51 merupakan salah satu produk mikrokontroller yang dikeluarkan oleh Atmel pada tahun 1994. Mikrokontroller AT89C51 sendiri terbentuk dari perpaduan arsitektur perangkat keras keluarga mikrokontroller MCS51 dari Intel dan tambahan teknologi Flash Memori, sehingga AT89C51 terbentuk sebagai mikrokontroller dengan fasilitas *timer*, *port serial*, 32 kaki I/O, RAM dan Flash Memori yang digunakan untuk keperluan penyimpanan program. Dengan demikian, desain elektronika menjadi ringkas, praktis dan ekonomis karena dimungkinkan untuk membuat suatu sistem hanya dalam satu buah *chip* saja.



Gambar 2.1. Konfigurasi Pin AT89C51

Gambar berikut di bawah ini merupakan gambar arsitektur perangkat keras mikrokontroler AT89C51.



Gambar 2.2. Arsitektur AT89C51

### 2.1.2. Deskripsi Pin AT89C51

1. **Vcc (pin 40)** berfungsi sebagai suplai tegangan.
2. **Ground (pin 20)** berfungsi sebagai ground atau pentanahan.
3. **Port 0.0 -0.7 (pin 32-39)**

Port 0 dapat berfungsi sebagai I/O biasa, dan dapat menerima kode *byte* saat *Flash programming*. Sebagai port keluaran, port ini dapat memberikan keluaran *sink* ke 8 buah masukan TTL. Selain itu, port juga dapat difungsikan sebagai masukan dengan memberikan logika 1 pada port tersebut.

4. **Port 1.0 – 1.7 (pin 1 -8)**

Port 1.0 berfungsi sebagai I/O biasa atau menerima *low order address bytes* selama *Flash Programming*. Port ini memiliki *internal pull up* dan dapat berfungsi sebagai masukan dengan memberikan logika 1. Sebagai keluaran port ini dapat memberikan keluaran *sink* ke -4 buah masukan TTL

5. **Port 2.0 – 2.7 (pin 21 – 28)**

Port 2 berfungsi sebagai I/O biasa atau *high order address*, saat mengakses memori secara 16 bit. Sebagai *output*, port ini dapat memberikan keluaran *sink* pada ke empat buah masukan TTL, sedangkan untuk mengfungsikan sebagai port input dilakukan dengan memberikan logika 1.

6. **Port 3.0 – 3.7 (pin 10 -17)**

Memiliki sifat yang sama dengan port 1 dan port 2 yaitu sebagai port I/O, sedangkan untuk fungsi spesifik sebagai berikut:

1. Port 3.0 (pin 10) Rxd berfungsi sebagai port *serial input*.
2. Port 3.1 (pin 11) Txd berfungsi sebagai port *serial output*.



3. Port 3.2 (pin 12) INT0 berfungsi sebagai port external interrupt 0.
4. Port 3.3 (pin 13) INT1 berfungsi sebagai port external interrupt 1.
5. Port 3.4 (pin 14) T0 berfungsi sebagai *port external timer 0 input*.
6. Port 3.5 (pin 15) T1 berfungsi sebagai *port external timer 1 input*.
7. Port 3.6 (pin 16) WR berfungsi sebagai *External Data Memori Write Strobe*.
8. Port 3.7 (pin 17) RD berfungsi sebagai *External Data Memori Read Strobe*.

#### 7. Reset (pin 9)

*Reset Input*. Reset akan aktif dengan memberikan masukan tinggi selama 2 cycle.

#### 8. ALE (pin 30)

Pin ini berfungsi sebagai *Address Latch Enable* (ALE) yang *me-latch low byte address* pada saat mengakses memori *eksternal*. ALE hanya akan aktif saat mengakses memori *eksternal*.

#### 9. PSEN (pin 29)

Pin ini berfungsi pada saat mengakses program yang terletak pada memori eksternal.

#### 10. EA (pin 31)

Pada kondisi rendah, pin ini akan berfungsi sebagai EA yaitu mikrokontroler akan menjalankan program yang ada pada memori eksternal setelah sistem di reset. Sedangkan jika berkondisi tinggi, pin ini akan berfungsi menjalankan program yang ada pada memori internal. Pada saat Flash Programming, pin ini akan mendapat tegangan 12 Volt

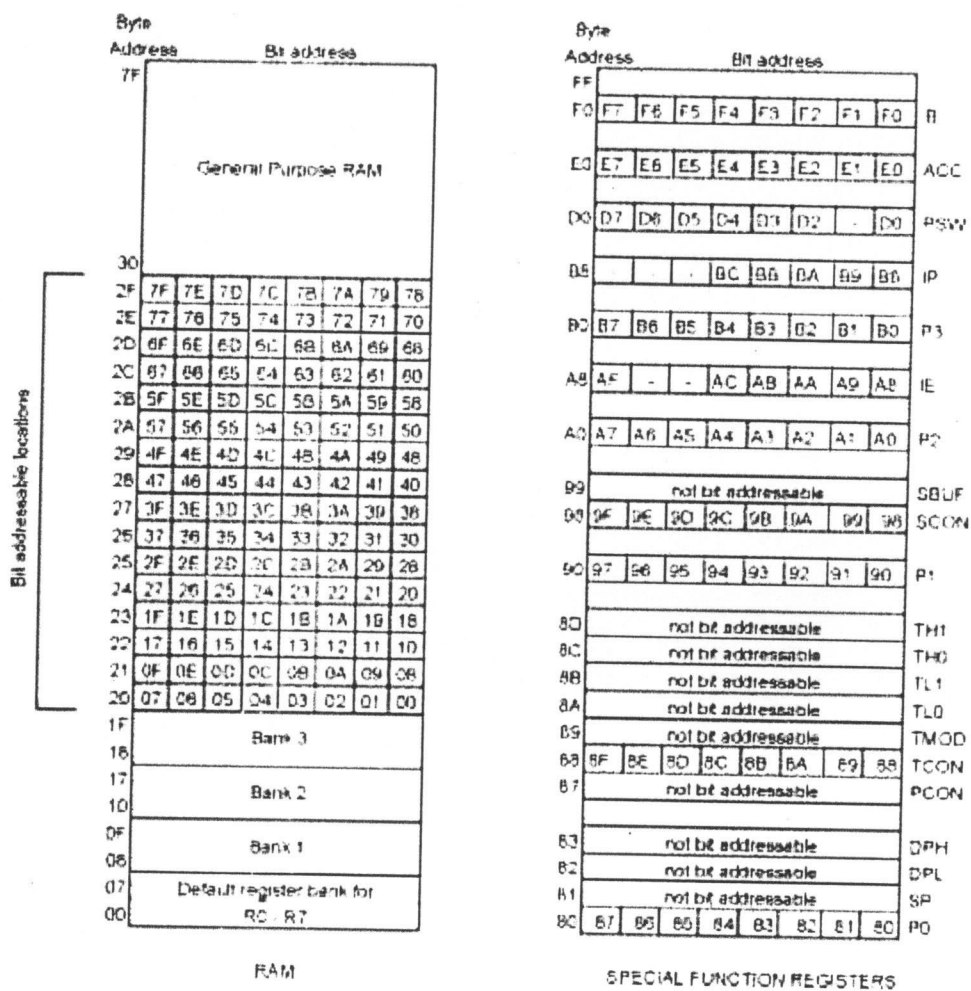
11. XTAL1 (pin 19)

*Input Oscillator*

12. XTAL2 (pin 18)

*Output Oscillator*

2.1.3 Struktur Memori Mikrokontroler AT89C51



Gambar 2.3. RAM Internal dan Flash PEROM

AT89C51 mempunyai struktur memori yang terpisah antara RAM Internal dan Flash PEROM-nya. Seperti tampak pada gambar diatas, RAM Internal dialamat oleh RAM Address Register (Register Alamat RAM) sedangkan Flash PEROM yang menyimpan perintah-perintah MCS-51 dialamat oleh Program Address Register (Register Alamat Program). Dengan adanya struktur memori yang terpisah tersebut, walaupun RAM Internal dan Flash PEROM mempunyai alamat awal yang sama, yaitu alamat 00H, namun secara fisik kedua memori tersebut tidaklah saling berhubungan.

#### **2.1.3.1 Ram Internal**

RAM *Internal*, dengan alamat dari 00H hingga 7FH dengan memori sebesar 128 *byte* yang biasanya digunakan untuk menyimpan data. RAM *Internal* sendiri terdiri atas :

- a. *Register Banks* yang berjumlah delapan buah dari R0 hingga R7, yang terletak pada alamat 00H hingga 07H pada setiap kali sistem direset.
- b. *Bit Addressable* RAM yang terletak pada alamat 20H hingga 2FH yang dapat diakses dengan pengalamatan bit.
- c. RAM keperluan umum yang dimulai dari alamat 30H hingga 7FH dan memiliki sifat dapat diakses dengan pengalamatan langsung maupun tak langsung.

#### **2.1.3.2. Special Function Register**

*Special Function Register* (Register Fungsi Khusus) merupakan memori yang berisi *register-register* yang memiliki fungsi-fungsi khusus yang merupakan

fasilitas yang disediakan oleh mikrokontroler tersebut, mencakup port-port, pewaktu (*timer*), kontrol *periferal* dan lain sebagainya. Sementara *Special Function Register* terdiri dari :

- a. Akumulator. ACC atau akumulator menempati alamat E0H dan digunakan untuk menyimpan data sementara dari hampir semua operasi aritmatik dan operasi logika. Instruksinya sendiri mengacu pada *register A*.
- b. *Register B*. Menempati lokasi F0H digunakan selama operasi perkalian dan pembagian walaupun juga dapat difungsikan sebagai *register* biasa. Pengalamatannya sendiri dapat dilakukan dengan *bit addressable*.
- c. *Program Status Word (PSW)*. *Register PSW (D0H)* mengandung informasi status program yang berkaitan dengan CPU saat itu. Status yang tersimpan pada PSW meliputi :
  1. *Carry Flag* yang berfungsi untuk mendeteksi apabila terjadi aksi lebih pada proses penjumlahan atau terjadi aksi pinjam pada operasi pengurangan. *Carry bit* juga dapat berfungsi sebagai akumulator pada operasi boolean sebesar 1 bit.
  2. *Bit RS0* dan *RS1* berfungsi untuk memilih satu dari empat *bank register*.
  - 3 *Overflow Flag* yang akan diset bila pada operasi aritmatik menghasilkan bilangan yang lebih besar daripada 128 atau lebih kecil dari -128.

4. *Bit* paritas berfungsi untuk menentukan cacah logik “1” dalam akumulator, jika cacah logik “1” ganjil maka bit paritasnya sama dengan 1 ( $P=1$ ) dan sebaliknya. *Bit* paritas ini juga digunakan untuk proses yang berhubungan dengan serial port yaitu sebagai *check sum*.
- d. *Stack Pointer*. Register SP (lokasi 81H) merupakan register dengan panjang 8 bit. Isi dari *Stack Pointer* sendiri merupakan data yang disimpan di *stack*. Proses yang berhubungan dengan *stack* ini biasa dilakukan dengan instruksi-instruksi *Push*, *Pop*, *Acall* dan *Lcall*.
- e. *Data pointer*. *Data pointer* atau DPTR merupakan register 16 bit dan terletak pada alamat 82H untuk DPL (*byte* rendah) dan 83H untuk DPH (*byte* tinggi). DPTR ini biasanya digunakan untuk mengakses source code ataupun data yang disimpan pada memori eksternal. Walaupun merupakan register 16 bit, namun dalam keperluan dapat dimanipulasi sebagai dua buah register 8-bit yang terpisah.
- f. Port. AT89C51 memiliki empat buah port, yaitu Port 0, Port 1, Port 2 dan Port 3 yang menempati lokasi 80h, 90h, A0h dan B0h merupakan pengunci-pengunci (*latches*) yang digunakan untuk menyimpan data yang akan dibaca atau ditulis dari/ke port untuk masing-masing port tersebut. Namun jika digunakan *eksternal* memori ataupun fungsi-fungsi spesial seperti *External Interrupt*, *Serial* maupun *External Timer*. Port 0, Port 2 dan Port 3 tidak dapat digunakan sebagai port dengan fungsi umum. Untuk masalah tersebut dapat digunakan Port 1 yang memang dikhususkan untuk port dengan fungsi umum. Semua

port ini dapat diakses dengan pengalamatan bit sehingga dapat dilakukan perubahan output untuk masing-masing pin tanpa mengganggu pin lainnya.

- g. *Register Port Serial*. AT89C51 memiliki sebuah *on chip serial port* yang dapat digunakan untuk berkomunikasi dengan *device* lain menggunakan serial port. *Buffer* (penyangga) untuk proses pengiriman maupun pengambilan data terletak pada *register* SBUF, yaitu pada alamat 99h. SBUF sebenarnya terdiri dari dua buah *register* yang menempati alamat yang sama yaitu *register* penyangga pengirim (*transmit buffer*) dan *register* penyangga penerima (*receiver buffer*). Pada saat data disalin ke SBUF, maka data sesungguhnya dikirim ke *buffer* pengirim dan sekaligus mengawali transmisi data serial. Sedangkan pada saat data diterima maka itu berasal dari *buffer* penerima.
- h. *Timer Register*. AT89C51 memiliki dua buah *register timer/counter* 16 bit yaitu *Timer 0* dan *Timer 1*. *Timer 0* terletak pada alamat 8Ah untuk TL0 dan 8Ch untuk TH0 dan *Timer 1* terletak pada alamat 8Bh untuk TL1 dan 8Dh untuk TH1.
- i. *Interrupt Register*. AT89C51 memiliki lima buah *register* interupsi dengan dua level prioritas interupsi. Interupsi akan selalu nonaktif setiap kali sebuah sistem direset. Register-register yang berhubungan dengan proses interupsi adalah *Interrupt Enable Register* (IE) atau Register Pengaktif Interupsi pada alamat A8h untuk mengatur

keaktifan tiap-tiap interrupt dan *Interrupt Priority Register* (IP) pada alamat B8h.

- j. Register Kontrol Power. Terdiri atas SMOD yang digunakan untuk melipat gandakan baud rate dari port serial, dua buah bit untuk flag fungsi umum pada bit ketiga dan bit kedua, *Power Down* (PD) bit dan Idle (IDL) bit.

### 2.1.3.3. Flash PEROM

Flash PEROM merupakan memori yang digunakan untuk menyimpan instruksi-instruksi MCS51. Mikrokontroler AT89C51 memiliki 4 Kb *Flash PEROM* (Programmable and Erasable Memory) yaitu ROM yang dapat ditulis ulang atau dihapus menggunakan sebuah *device programmer*. Jenis *Flash* yang digunakan adalah *Atmel's High-Density Non Volatile Technology* yang memiliki kemampuan untuk ditulis ulang hingga 1000 kali dan berisi fungsi standar MCS51.

Mikrokontroler AT89C51 memiliki mode pengalamatan yang sama dengan mikrokontroller MCS51. Mode pengalamatan ini berkaitan erat dengan operan dalam pemrograman mikrokontroller. Operan sendiri merupakan data yang tersimpan di dalam memori, register dan input/output (IO). Operan-operan ini di jalankan melalui instruksi-instruksi. Instruksi secara umum terdiri dari beberapa kelompok yaitu instruksi untuk pemindahan data, aritmatika, operasi logika, pengaturan aliran program.

#### 2.1.4. Mode-mode Pengalamatan AT89C51

Di bawah ini merupakan mode-mode pengalamatan MCS51/AT89C51 :

**a. Mode Pengalamatan Segera (*Immediate Addressing Mode*).**

Ciri mode ini, pengalamatannya menggunakan konstanta, misalnya `MOV A, #21h`. Instruksi ini berarti data konstantanya yaitu 21h di salin ke Akumulator A.

**b. Mode Pengalamatan Langsung (*Direct Addressing Mode*).**

Mode ini digunakan untuk menunjuk data yang berada pada suatu lokasi memori dengan cara menyebut lokasi memori tempat data tersebut berada, misalnya: `MOV A, 30h`. Instruksi ini berarti bahwa data yang berada pada alamat 30h disalin ke Akumulator A.

**c. Mode Pengalamatan Tidak Langsung (*Indirect Addressing Mode*).**

Mode ini digunakan untuk mengakses data yang berada di dalam memori, namun memori tersebut tidak disebutkan secara langsung melainkan dititipkan pada sebuah register lain. Misalnya, `MOV A, @R0`. Yang pertama dilakukan adalah inisialisasi nilai sebuah alamat ke register R0, kemudian untuk operasi selanjutnya R0 akan digunakan terus, sehingga dapat dikatakan bahwa register R0 merupakan register penyimpan alamat. Tanda '@' digunakan untuk menandai lokasi memori yang tersimpan dalam register.

**d. Mode Pengalamatan Register (*Register Addressing Mode*).**

Contoh dari mode ini : `MOV A, R2`. Instruksi berarti bahwa data dari register R2 akan disalin ke Akumulator A.



e. **Mode Pengalamatan Kode Tidak Langsung (*Code Indirect Addressing Mode*).**

Mode ini digunakan sebagai cara penyebutan data dalam memori program yang dilakukan secara tak langsung, misalnya saja : `MOVC A, @A+DPTR`. Tambahan huruf C dimaksudkan untuk membedakan bahwa instruksi ini digunakan untuk memori program (Jika tanpa C, maka berarti memori data).

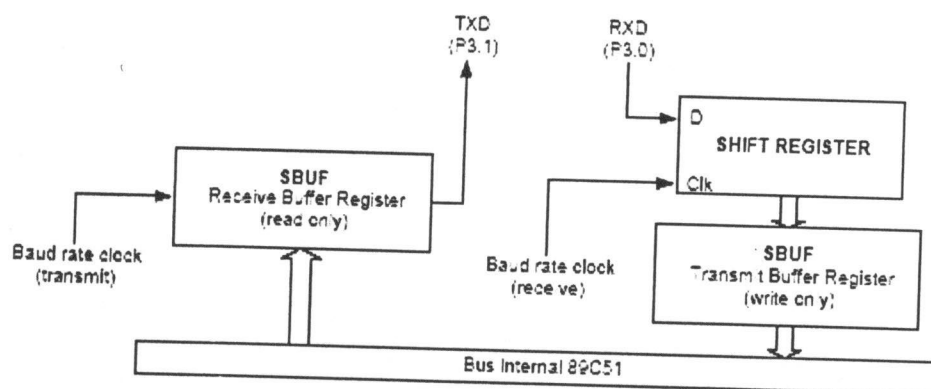
### 2.1.5. Mengatur Alur Program

Instruksi-instruksi yang digunakan untuk mengatur alur program dalam mikrokontroller terdiri atas instruksi-instruksi JUMP (sama dengan instruksi GOTO dalam Pascal), penggunaan *sub-rutin/modul* (sama dengan *procedure/function*) dan instruksi-instruksi JUMP bersyarat (*Conditional Jump* yang setara dengan IF ...THEN di Pascal atau IF ...*statement* di C/C++). Selain itu ada pula instruksi-instruksi PUSH dan POP. Yang harus di ingat adalah karena *Program Counter* merupakan satu-satunya register dalam mikrokontroller yang mengatur alur program, maka mengatur alur program dengan instruksi-instruksi di atas tersebut juga berarti mengubah nilai *Program Counter* juga.

### 2.1.6. Operasi Serial Port

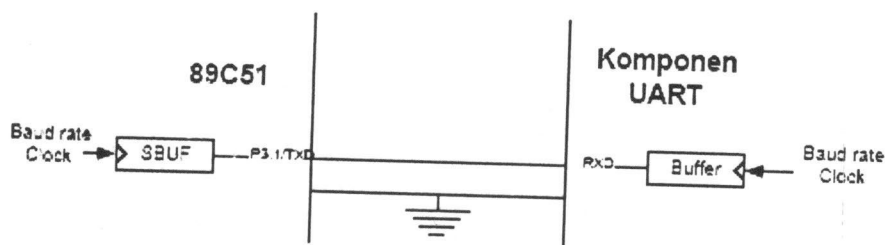
Mikrokontroller juga berkomunikasi dengan Personal Computer (PC) melalui port serial. Port serial AT89C51 dapat digunakan untuk komunikasi *sinkron* dan *asinkron*. Komunikasi *data serial* secara *sinkron* adalah komunikasi *data serial* yang memerlukan sinyal *clock* untuk sinkronisasi setiap kali pengiriman data. Sedangkan komunikasi *asinkron* tidak memerlukan sinyal *clock*

setiap kali pengiriman data. Pengiriman data serial di AT89C51 dilakukan dari bit yang paling rendah (LSB) hingga ke bit yang paling tinggi (MSB).



Gambar 2.4. Diagram Blok Port Serial

Adapun untuk aplikasinya proses komunikasi *asinkron* selalu digunakan untuk mengakses komponen-komponen yang mempunyai fasilitas UART (*Universal Asynchronous Receiver/Transmitter*) seperti port serial PC ataupun port serial mikrokontroler yang lain. Walaupun tidak menggunakan sinyal *clock* sebagai *sinkronisasi*, tetapi pengiriman data harus diawali dengan *start bit* dan *stop bit* seperti pada gambar di bawah ini. Sinyal *clock* sendiri merupakan *baud rate* dari komunikasi data yang dibangkitkan oleh masing-masing pengirim dan penerima data dengan frekuensi yang sama.



Gambar 2.5. Komunikasi UART

AT89C51 memiliki 4 buah mode operasi yang diatur oleh bit ke-7 dan bit ke-5 dari *Register SCON (Serial Control)*.

SCON

	SCON.7	SCON.6	SCON.5	SCON.4	SCON.3	SCON.2	SCON.1	SCON.0
98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI

**Gambar 2.6. Register SCON**

Adapun keterangan dari masing-masing bit register SCON adalah sebagai berikut :

1. SM0: *Serial Port Mode bit 0*, bit Pengatur Mode Serial
2. SM1: *Serial Port Mode bit 1*, bit Pengatur Mode Serial
3. SM2: *Serial Port Mode bit 2*, bit untuk mengaktifkan komunikasi multiprosesor pada kondisi set.
4. REN: *Receive Enable*, bit untuk mengaktifkan penerimaan data dari port serial pada kondisi *set*. Bit ini di *set* dan *clear* oleh program.
5. TB8: *Transmit bit 8*, bit ke 9 yang akan dikirimkan pada mode 2 atau 3. Bit ini di *set* dan *clear* oleh program.
6. RB8: *Receive bit 8*, bit ke 9 yang diterima pada mode 2 atau 3. Pada Mode 1 bit ini berfungsi sebagai *stop bit*.
7. TI: *Transmit Interrupt Flag*, bit yang akan set pada akhir pengiriman karakter. Bit ini di *set* oleh perangkat keras dan di *clear* oleh program.
8. RI: *Receive Interrupt Flag*, bit yang akan *set* pada akhir penerimaan karakter. Bit ini di *set* oleh perangkat keras dan di *clear* oleh program.

Tabel 2.1. Mode Operasi Serial

SM0	SM1	Mode	Deskripsi
0	0	0	Shift Register 8 bit
0	1	1	UART 8 bit dengan baud rate yang dapat diatur.
1	0	2	UART 9 bit dengan baud rate permanen.
1	1	3	UART 9 bit dengan baud rate yang dapat diatur.

Pada mode 1, UART 8 bit dengan *baud rate* yang dapat diatur akan tampak seperti gambar di bawah ini :

SCON		SCON.7	SCON.6	SCON.5	SCON.4	SCON.3	SCON.2	SCON.1	SCON.0
98H		0	1	X	X	X	X	X	X

Gambar 2.7 Mode 1 Operasi Serial

Pada mode ini komunikasi data dilakukan secara 8 bit data *asinkron* yang terdiri 10 bit yaitu 1 *bit start*, 8 *bit data* dan 1 *bit stop*. *Baud Rate* pada mode ini dapat diatur dengan menggunakan Timer 1. Tidak seperti pada mode 0, pada mode ini yang merupakan mode UART, fungsi-fungsi alternatif dari P3.0/RXD dan P3.1/TXD digunakan. P3.0 berfungsi sebagai RXD yaitu kaki untuk penerimaan data serial dan P3.1 berfungsi sebagai TXD yaitu kaki untuk pengiriman data serial. Hal ini juga berlaku pada mode-mode UART yang lain seperti mode 2 dan mode 3. Pengiriman data dilakukan dengan menuliskan data yang akan dikirim ke *Register SBUF*. Data serial akan digeser keluar diawali dengan *bit start* dan diakhiri dengan *bit stop* dimulai dari bit yang berbobot terendah (LSB) hingga bit berbobot tertinggi (MSB). Bit TI akan set setelah *bit stop* keluar melalui kaki TXD yang menandakan bahwa proses pengiriman data telah selesai. Bit ini harus

di-clear oleh perangkat lunak setelah pengiriman data selesai. Penerimaan data dilakukan oleh mikrokontroler dengan mendeteksi adanya perubahan kondisi dari logika *high* ke logika *low* pada kaki RXD di mana perubahan kondisi tersebut adalah merupakan *bit start*. Selanjutnya data serial akan digeser masuk ke dalam SBUF dan *bit stop* ke dalam bit RB8. Bit RI akan *set* setelah 1 *byte* data diterima ke dalam SBUF kecuali bila *bit stop* = 0 pada komunikasi multiprosesor (SM2 = 1).

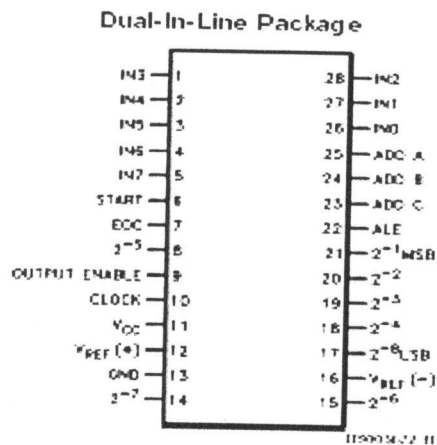
## 2.2. ANALOG DIGITAL CONVERTER (ADC 0809)

ADC digunakan untuk mengkonversi besaran-besaran analog menjadi besaran-besaran digital yang dimengerti oleh komputer. ADC yang digunakan adalah ADC 0809 dari National Semiconductor. ADC tipe ini merupakan komponen akuisisi data dengan 8 bit A/D konverter, 8 buah masukan yang dimultipleks dan kompatibel dengan kontrol logika mikroprosessor. 8 buah kanal atau masukan memungkinkan untuk mengakses secara langsung 8 buah sinyal atau masukan analog.

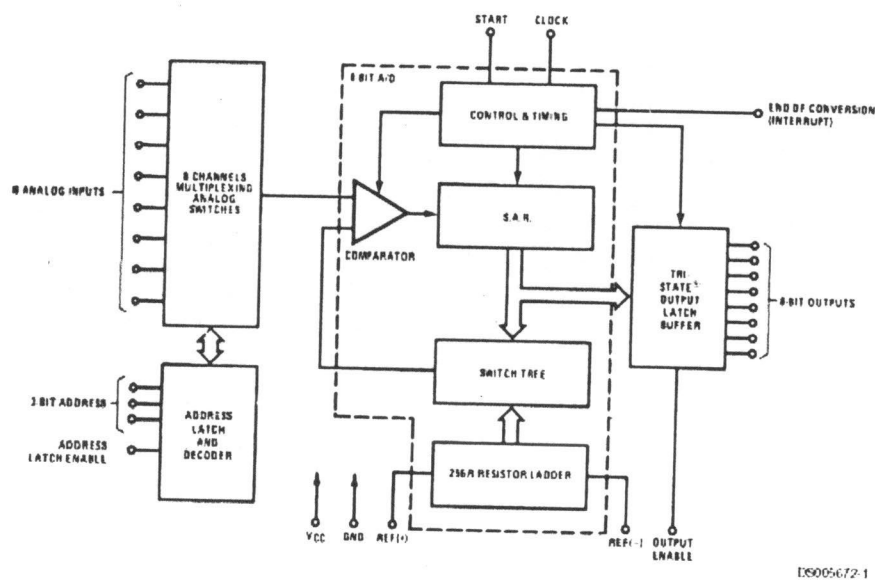
Adapun kemampuan yang terdapat pada ADC 0809 adalah sebagai berikut:

1. Mudah untuk di gunakan bersama rangkaian mikroprosessor.
2. Tidak diperlukan penyesuaian yang rumit.
3. 8 *channel multiplexer* dengan 8 buah alamat logic.
4. Jangkauan input berkisar 0 -5 volt dengan hanya satu buah catu daya 5 volt.
5. Memiliki resolusi 8 bit.

6. Total kesalahan sekitar  $\pm \frac{1}{2}$  LSB dan  $\pm 1$  LSB.
7. Catu daya yang dikonsumsi sebesar 5 volt.
8. Waktu konversi sekitar 100  $\mu$ s



Gambar 2.9 Pin ADC 0809



Gambar 2.10. Diagram Blok ADC 0809

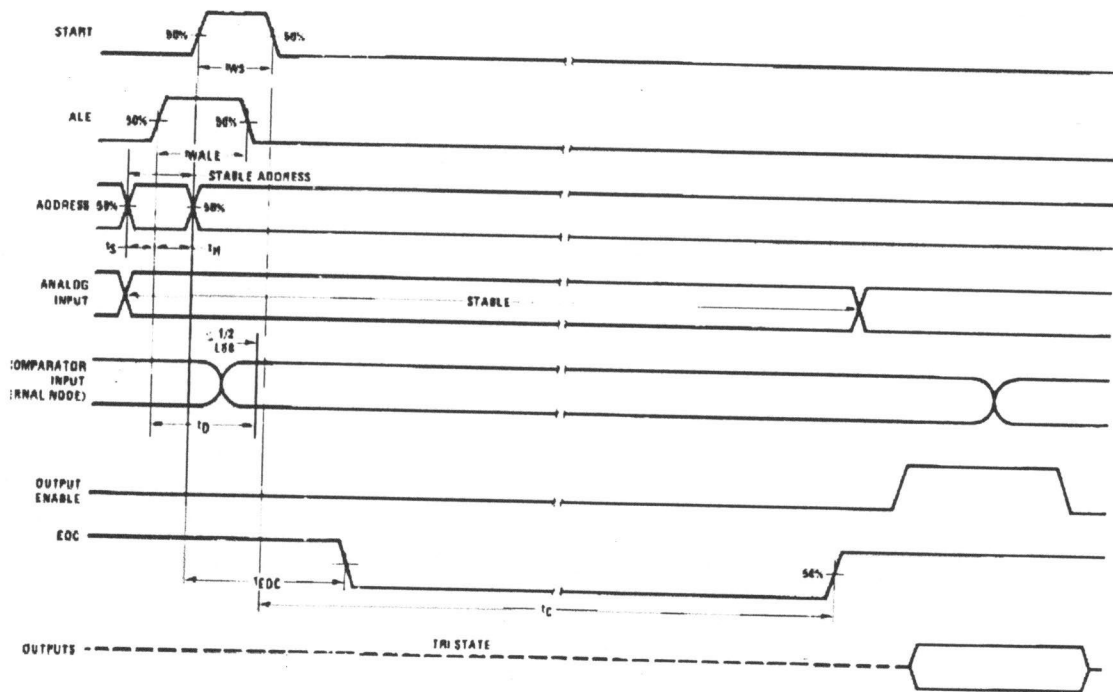
ADC ini mempunyai ketelitian sebesar 1 bit LSB, untuk ketelitian yang lebih akurat  $\frac{1}{2}$  bit LSB, IC ini dapat digantikan dengan ADC0808 yang mempunyai konfigurasi pin sama persis dengan ADC0809. ADC0809 melakukan konversi tegangan analog ke digital dengan menggunakan metode SAR (*successive approximation register*) dengan resolusi 8 bit dan waktu konversi 100  $\mu$ s.

Pada dasarnya *Analog to Digital Converter* (ADC) memiliki 2 bagian yaitu, bagian multiplexer dan bagian konverter. Bagian multiplexer ini memiliki 8 buah masukan. Setiap masukan memiliki alamat sendiri sehingga dapat dipilih secara terpisah melalui *Address* A0, A1 dan A2. Tabel dibawah ini menunjukkan alamat dari masing – masing masukan.

**Tabel 2.2. Tabel alamatADC**

SELECTED ANALOG CHANNEL	ADDRESS LINE		
	A2	A1	A0
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

Konverter sendiri merupakan jantung dari chip ADC. Konverter di desain untuk kecepatan konversi, keakuratan dan jangkauan yang luas dalam suatu aplikasi. Proses konversi ADC 0809 sendiri dapat dijelaskan melalui diagram pewaktuan dibawah ini.



Gambar 2.11. Diagram Pewaktuan ADC 0809

Pada diagram pewaktuan di atas, tampak bahwa proses konversi mulai terjadi saat sinyal ALE dan START muncul. Sinyal analog yang muncul pada kanal-kanal akan sesuai dengan konversi sinyal-sinyal analog pada kaki-kaki A0,A1,A2. Akhir proses konversi ditandai dengan berubahnya nilai logika dari 0 ke 1 pada kaki EOC. Data hasil konversi akan muncul di *Data Bus* (D0..D7) saat sinyal logika OE bernilai 1.

### 2.3. SENSOR SUHU LM35

Sensor suhu berfungsi untuk mengkonversi besaran panas yang di tangkap oleh sensor menjadi besaran tegangan. Sensor temperatur yang digunakan dalam rangkaian adalah IC LM35 yang merupakan sensor suhu semipenghantar dengan presisi tinggi yang beroperasi pada tegangan +5 volt DC. Sensor ini merupakan



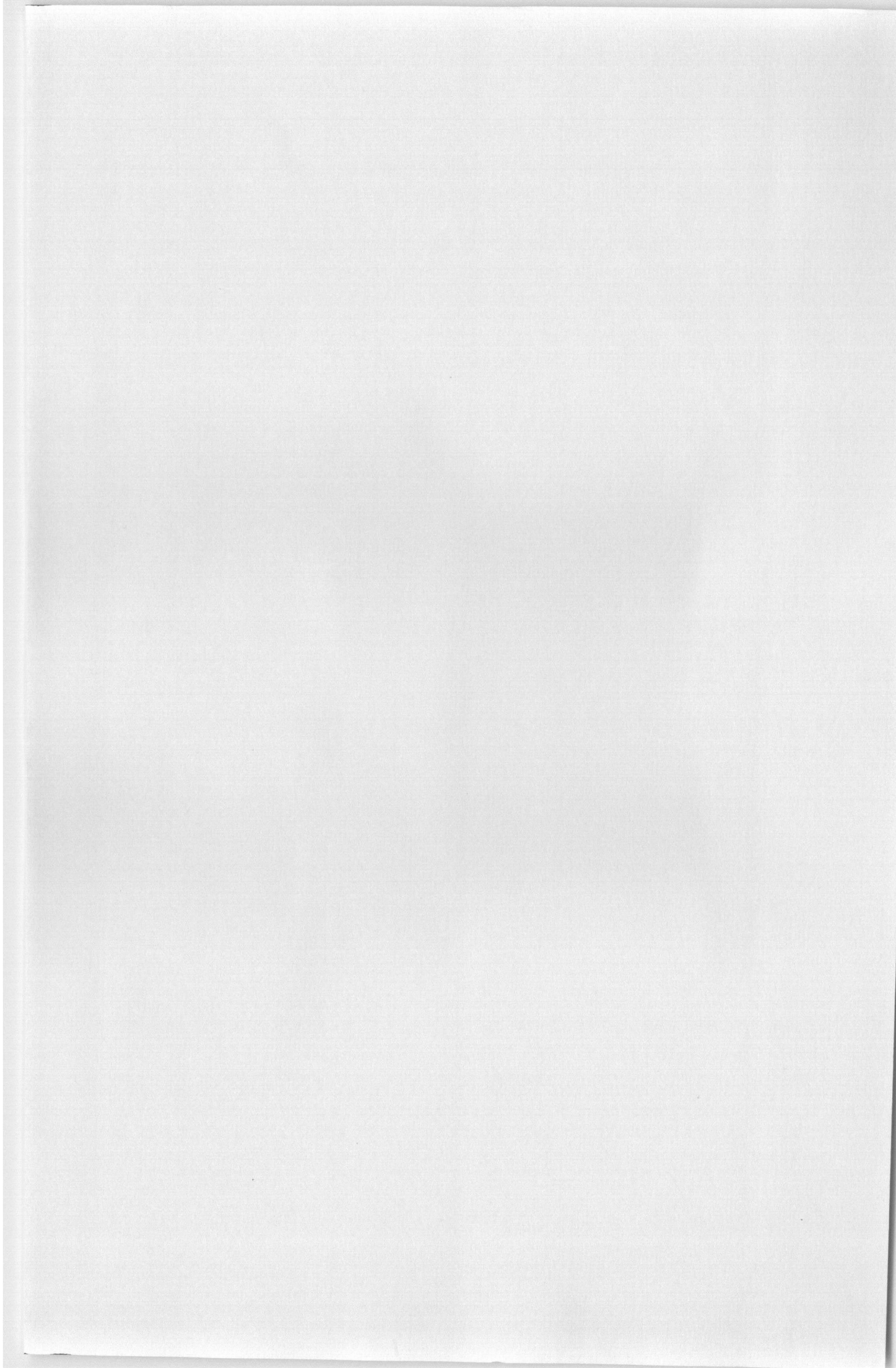
komponen sederhana, dengan hanya 3 kaki membuat antarmuka komponen ini begitu ringkas jika dibandingkan menggunakan IC DS1620. Kaki pertama IC LM35 dihubungkan ke sumber daya, kaki yang kedua sebagai output dan kaki yang ketiga dihubungkan ke *ground*. IC LM35DT mengubah besaran panas menjadi besaran listrik, dengan menghasilkan nilai konversi suatu voltase keluaran analog, sebanding dengan perubahan yang melingkupi temperatur pada skala celcius (10 mV/C). Berarti nilai konversi yang diperoleh pada perubahan temperatur 1°C di wakili tegangan 10 mV.

Adapun karakteristik dari IC LM35 adalah sebagai berikut:

1. Terkalibrasi ke dalam besaran *Celcius*.
2. Ketepatan presisi berkisar 0.5 °C.
3. Memiliki jangkauan pengukuran dari – 55 hingga 150°C.
4. Cocok untuk aplikasi *remote*.
5. Beroperasi dari 4 hingga 30 volt.
6. Arus *drain* yang kurang dari 60  $\mu$ A.
7. Memiliki Impedansi keluaran yang rendah, sekitar 0.1  $\Omega$  untuk beban 1 mA.



**Gambar 2.12. IC LM335**



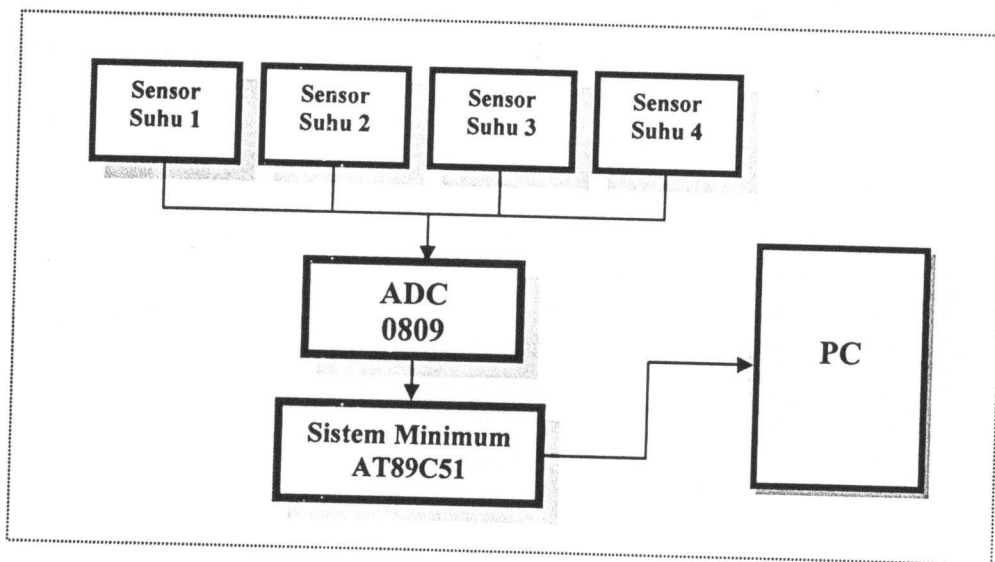
## BAB III

### PERANCANGAN SISTEM

Dalam bab 3 akan dibahas mengenai perancangan sistem termasuk rancangan sensor, rancangan penguat, rancangan *Analog to Digital Converter*, Sistem Minimum AT89C51, dan Komunikasi serial port dengan RS232 dan perancangan perangkat lunak.

#### 3.1. Rancangan Sistem

Perancangan untuk Sistem Monitor/Logger Temperature berbasis AT89C51 dapat dilihat pada diagram dibawah ini :



**Gambar 3.1 Diagram Blok Sistem**

##### 3.1.1. Rangkaian Sensor Temperatur

Aplikasi monitor/logger temperature ini dilakukan dengan melakukan konversi besaran suhu menjadi data digital yang dapat diolah mikrokontroller. Terdapat dua kali proses konversi, yang pertama kali konversi dari besaran suhu

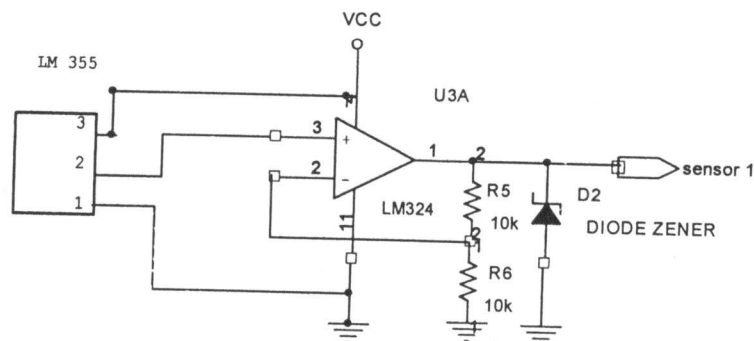
menjadi besaran tegangan (analog) dan proses konversi kedua yaitu perubahan besaran besaran analog menjadi besaran digital. Konversi suhu menjadi tegangan (analog) dilakukan dengan menggunakan sensor suhu tipe LM35 di mana sensor ini dapat beroperasi dengan menggunakan tegangan sumber di antara 4 – 30 volt DC. Keluaran sensor ini akan naik sebesar 10 mV setiap derajat *Celcius* sehingga diperoleh persamaan berikut:

$$V_{LM35} = \text{Temperatur} * 10 \text{ mV} \quad (3.1)$$

Yang menjadi permasalahan adalah mengenai penguatan tegangan dari sensor temperature, seperti diketahui ADC 0809 memiliki ketelitian dalam satuan tegangan adalah:

$$\begin{aligned} \text{Ketelitian}(V) &= \frac{\text{Resolusi} * \text{skalamaksimum}}{2^8 - 1} \quad (3.2) \\ &= 0,0196 \text{ volt atau } 19,6 \text{ mV} \end{aligned}$$

dimana Resolusi =1 dan skala maksimum = skala maksimum tegangan= 5 volt. Dengan toleransi ketelitian 19,6 mV maka keluaran LM35 yang mempunyai kenaikan 10 mV untuk setiap derajat *Celcius* tidak dapat langsung dihubungkan ke ADC-0809. Toleransi ketelitian yang lebih besar dari tingkat kenaikan tegangan yang diukur akan menyebabkan kesalahan dalam pengukuran. Untuk menghindari kesalahan tersebut diperlukan rangkaian penguat sehingga tingkat kenaikan tegangan berada di atas toleransi ketelitian seperti pada gambar dibawah ini.



**Gambar 3.2 Rangkaian Sensor LM35 dengan Penguat**

Rangkaian penguat ditunjukkan pada gambar 3.2 diatas yang merupakan penguat non-inversi (*non-inverting amplifier*), penguatan yang didapat berdasarkan persamaan sebagai berikut :

$$V_o = \left(1 + \frac{R_b}{R_a}\right) \cdot V_i \quad (3.3)$$

sehingga penguatan tegangannya adalah :

$$\frac{V_o}{V_i} = 1 + \frac{R_b}{R_a} \quad 3.4)$$

$$\text{Gain} = 2,0 \text{ kali}$$

Setiap sensor suhu LM35 membutuhkan penguatan sehingga dibutuhkan 4 buah rangkaian penguat untuk masing-masing sensor. IC LM 324 ini memiliki 4 buah rangkaian penguat dalam satu IC, sehingga dapat digunakan untuk keperluan penguatan 4 buah sensor. *Diode zener* 5,1 volt berfungsi untuk mencegah tegangan dari penguat lebih dari 5 volt agar tidak merusak modul ADC dan modul Development System yang menggunakan tegangan 5 volt. IC LM 324 dan sensor LM35 mendapat catu daya sebesar 12 volt DC. Modul sensor sendiri

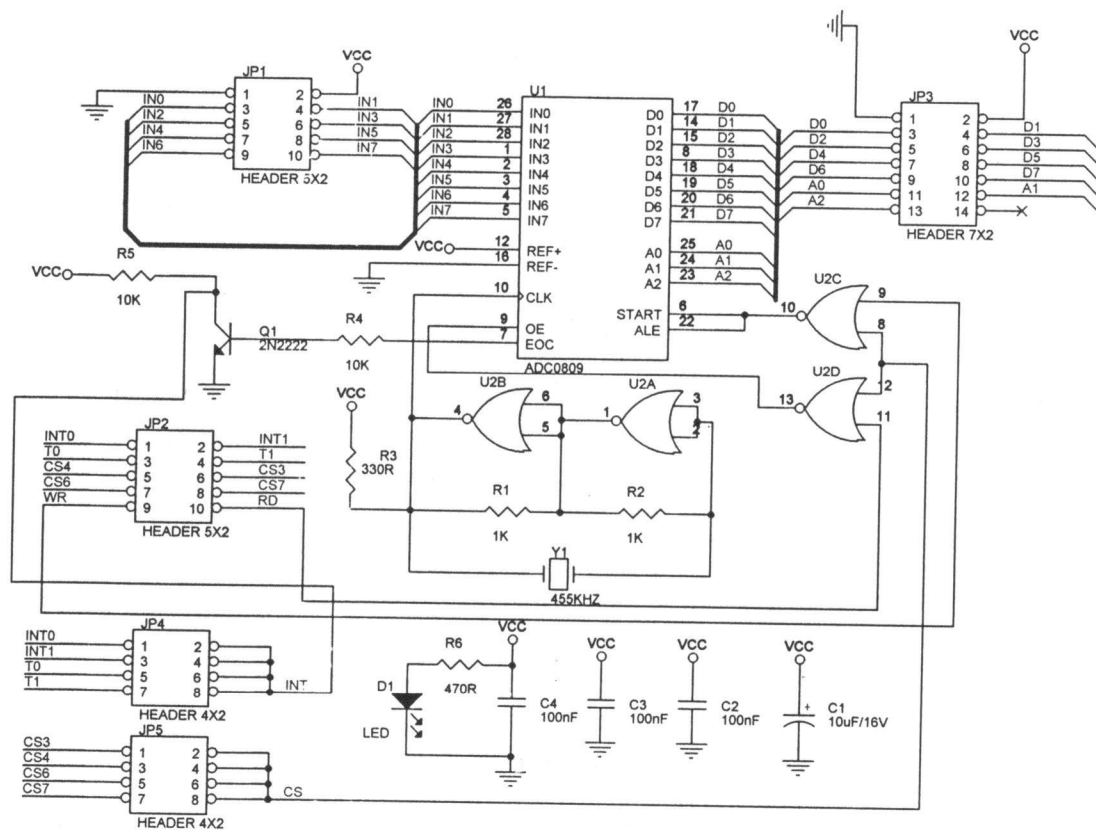
mendapatkan input 12 volt dari Konektor Serba guna Development System yang juga berfungsi sebagai keluaran tegangan dengan nilai sebesar tegangan masukan pada konektor *Input*.

### 3.1.2. Perancangan Modul Analog to Digital Converter (ADC 0809)

ADC-0809 adalah merupakan modul konversi tegangan analog ke digital dengan spesifikasi:

1. Konversi digital 8 bit
2. Resolusi 1 LSB
3. Tegangan input maksimum 5Volt

ADC 0809 ini tidak seperti ADC 0804 yang memiliki Chip Select (CS) yang digunakan untuk mengaktifkan dan menonaktifkan IC tersebut, sehingga tidak dapat dihubungkan secara langsung ke sistem pengalamatan development system, oleh karena itu dibuat rangkaian tambahan untuk ADC 0809 yang berfungsi sebagai CS dan juga rangkaian osilator. Rangkaian ini menggunakan IC 74LS02. Rangkaian *Chip Select* tetap digunakan karena rangkaian ini berfungsi sebagai osilator yang diperlukan clock ADC sekaligus *inverter* kondisi sinyal WR dan RD dari AT89C51.



**Gambar 3.3. Rangkaian Modul ADC 0809**

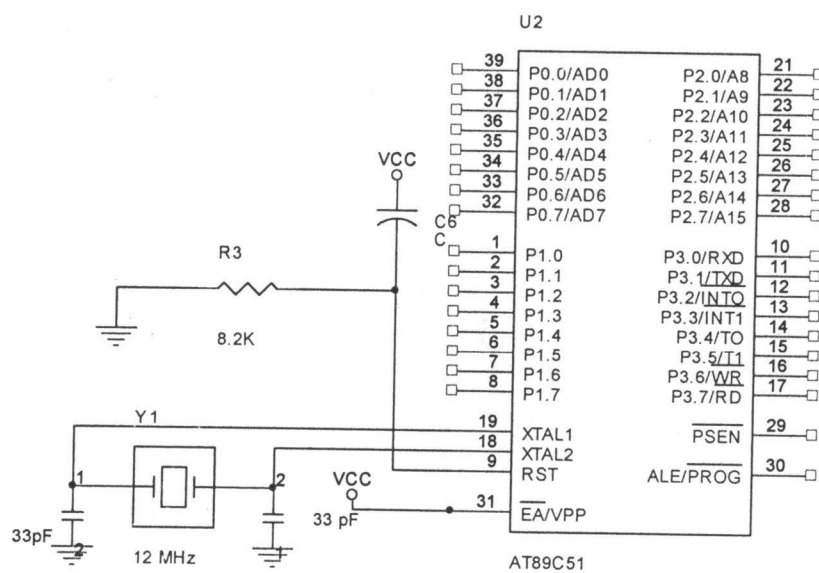
Rangkaian transistor berfungsi sebagai saklar untuk menandai jika proses konversi selesai EOC (*End Of Conversion*). Kolektor dari transistor terhubung ke pin INT0 dari sistem minimum. Untuk input digunakan *header 5x2* yang dapat menampung 8 masukan, 2 pin sisa berfungsi sebagai *Vcc* dan *Ground* bagi modul masukan ADC. *Header 7x2* berfungsi sebagai *bus* data dan *bus* alamat D0-D7 dan A0-A2, 2 pin sisa juga merupakan pin *Vcc* dan *Ground*, sehingga modul ADC mendapat input catu daya pula dari *Development System* sehingga tidak perlu membuat lagi catu daya sendiri untuk modul ADC. 2 header tersisa yaitu masing-masing header 4x2 berfungsi sebagai Interrupt dan satunya berfungsi sebagai



alamat CS (Chip Select) yang dapat dipilih dari alamat 4000H,6000H dan 8000H, dengan cara memindahkan jumper pada pin alamat bersangkutan.

### 3.1.3. Perancangan Mikrokontroller AT89C51

Bagian sistem sebuah mikrokontroller sangatlah sederhana, hanya memerlukan sebuah mikrokontroller AT89C51, rangkaian osilator yang dibentuk dari rangkaian kristal 11,0592 MHz (untuk sistem ini dipakai kristal 12 Mhz, karena tidak tersedia di pasar) dan 2 buah kapasitor 33pF dan rangkaian reset yang terdiri dari gabungan sebuah resistor dan kapasitor, seperti ditunjukkan pada gambar di bawah ini.



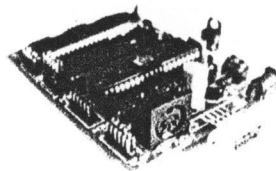
Gambar 3.4. Skema Single Chip Mikrokontroller

Ada satu hal yang harus diperhatikan dalam pin mikrokontroller keluarga Atmel, yaitu untuk kaki EA. Kaki EA ini berfungsi untuk menentukan akses awal dilakukan dari Flash Perom atau dari Memori eksternal. Jika kaki EA terhubung VCC atau berlogika 1 maka jalannya program akan dimulai dari alamat 0000H



dari Flash Perom, namun sebaliknya jika EA terhubung ground atau berlogika 0 maka jalannya program akan dimulai dari memori eksternal.

Namun pada sistem ini digunakan modul *development system* dari Delta Electronic. *Development System* sendiri sebenarnya merupakan modul *singel chip* yang dilengkapi memori tambahan dan berbagai antar muka dengan hardware yang lain. *Development System* dipilih karena kemudahannya dalam memprogram mikrokontroler, simulasi *online* dengan sistem dan pertimbangan jika sistem hendak di *upgrade* atau di perluas maka tidak banyak merubah sistem yang telah ada.



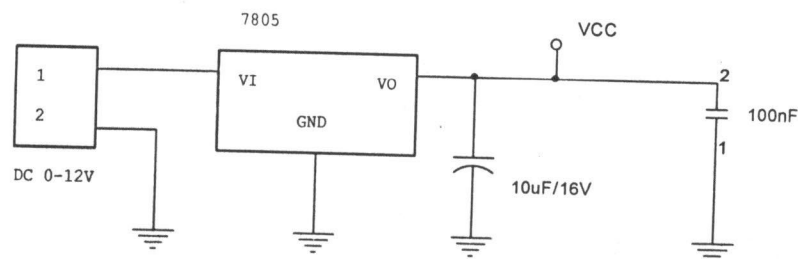
**Gambar 3.5 Modul Development System DST 52**

Adapun Fitur dari Development System DST-51/52 sebagai berikut :

- a) 3 in 1 *Development, Application & Prototype System*.
- b) 5 *Operation Mode*.
- c) 8 Kb EEPROM.
- d) *Parallel I/O (PPI 8255)*.
- e) *On Cable RS 232 Communication*.
- f) *PC Keyboard Connector*.
- g) *Universal Power Connector*.
- h) *DDT-52 Debugger Tools Software*.
- i) *DST-52 Monitor Tools Software*.

### 3.1.4. Rangkaian Catu Daya

Catu daya yang digunakan untuk keseluruhan rangkaian ini hanyalah 5 volt DC, hal ini dikarenakan komponen-komponen yang digunakan kebanyakan seperti IC TTL yang bekerja pada daya 5 volt.



Gambar 3.6. Rangkaian Power Supply

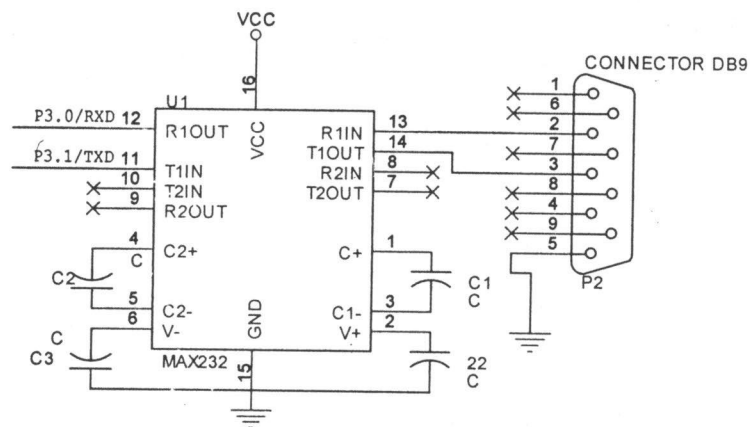
Untuk sumber tegangan digunakan adaptor DC 9-12 V yang tersedia di pasaran dan dengan harga yang murah, IC LM 7805 digunakan sebagai regulator yang berfungsi untuk mengubah tegangan dari 9-12 volt menjadi 5 volt saja.

### 3.1.5. Komunikasi Port Serial AT89C51 dan PC

Besaran listrik yang merupakan sinyal-sinyal analog akan diubah menjadi sinyal digital 8-bit oleh ADC0809. Keluaran sinyal analog dari sensor ini sebanding dengan voltase masukan sinyal analog. Keluaran sinyal digital dari ADC0809 akan diumpankan ke mikrokontroller AT89C51. AT89C51 memiliki *On Chip Serial Port* yang dapat digunakan untuk komunikasi data serial secara *Full Duplex* sehingga Port Serial ini masih dapat menerima data pada saat proses pengiriman data terjadi. Untuk menampung data yang diterima, AT89C51 mempunyai sebuah register, yaitu SBUF yang terletak pada alamat 99H. Register ini berfungsi sebagai *buffer* sehingga pada saat mikrokontroller ini membaca data yang pertama dan data kedua belum diterima secara penuh, data ini tidak hilang.

Register SBUF sendiri terdiri atas dua buah register, yaitu *Transmit Register* yang bersifat *write only* dan *Receive Register* yang bersifat *read only*.

Program pada Microcontroller AT89C51 kemudian akan membaca data-data temperatur ruangan dari data yang dikirimkan melalui ADC0809, kemudian mengirimkannya ke *Program Monitoring* Komputer melalui Port Serial. Mikrokontroler AT89C51 akan mengirimkan data serial tersebut melalui port serial dalam level TTL dan akan diubah oleh antar muka RS232 menjadi bentuk RS232 ke port serial PC.



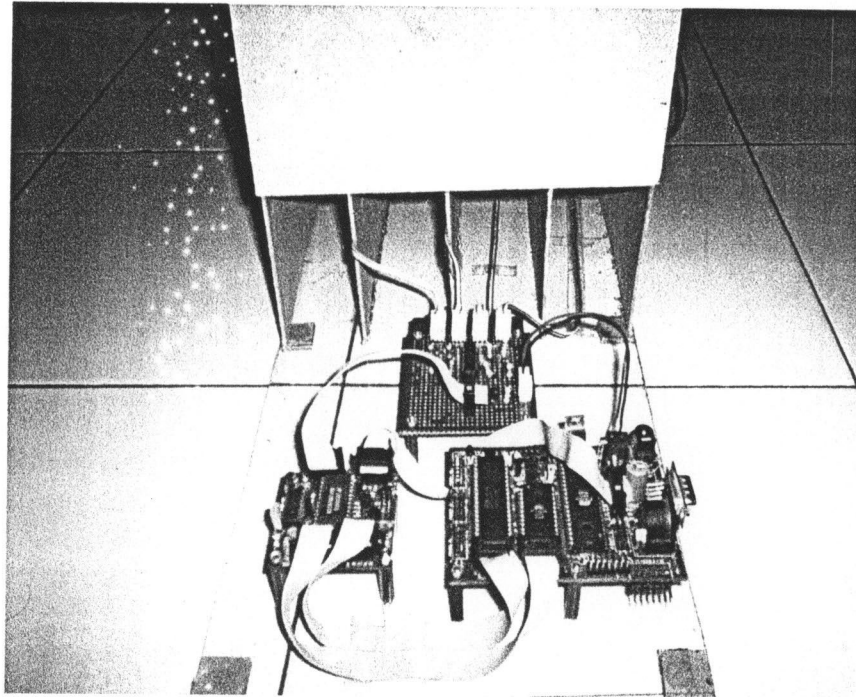
**Gambar 3.7. Rangkaian Antar Muka RS 232**

Untuk antar muka RS 232 ini digunakan IC Max 232 yang akan mengubah level tegangan TTL ke RS232 dan sebaliknya dari RS232 ke level TTL. IC max 232 terhubung ke *conector* DB9, melalui kaki 2 DB9 ke kaki 13 Max232 (R1 IN), kaki 3 DB9 ke kaki 14 Max 232 (T1Out).

### 3.2 *Prototype* Sistem Monitor/Logger

*Protoype* sistem monitor/logger di desain seperti halnya sebuah bangunan yang memiliki empat buah ruangan yang terpisah oleh sekat. Pada atap ruang satu dan dua diletakkan sebuah lampu pijar 40 watt yang berfungsi untuk memberi

panas saat simulasi dilakukan. Bangunan gedung dibuat dengan akrilik dengan panjang 20 cm dan tinggi 22 cm. *Protoype* tampak seperti gambar berikut ini.



**Gambar 3.8. Protoype Sistem Monitor/Logger**

### **3.3 Rancangan Program**

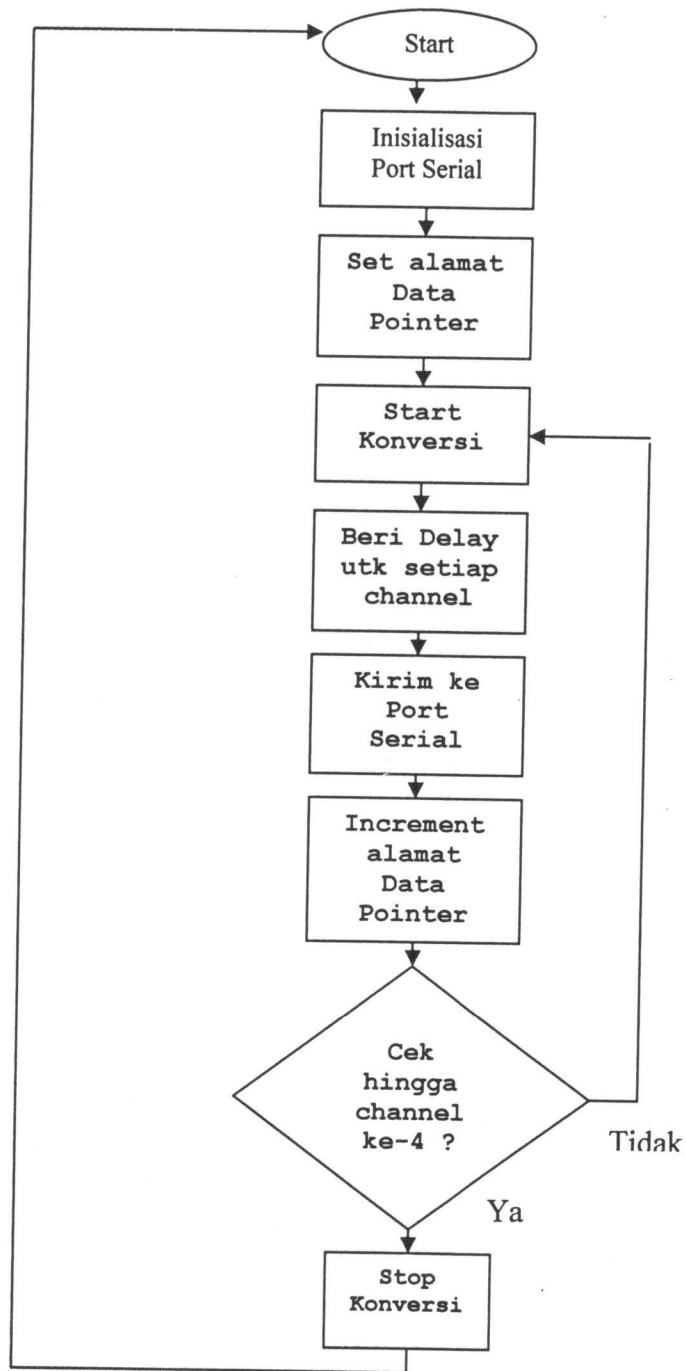
Perancangan perangkat lunak terbagi atas 2 buah macam, yang pertama perancangan program untuk Mikrokontroller dan yang kedua untuk program pada PC.

#### **3.3.1. Program pada Mikrokontroller.**

Pengembangan aplikasi mikrokontroller selalu menggunakan bahasa Assembly, walaupun terdapat bahasa lain semisal Basic ataupun C, namun untuk kasus-kasus tertentu seperti pada pemakaian yang menuntut kecepatan, bahasa Assembly dipilih untuk menyelesaikan. Program pada mikrokontroller ini berfungsi untuk mengambil data dari ADC ( 4 *channel* ) setiap satu kali masa

konversi. Data dari ADC tersebut kemudian akan dikirimkan ke port serial pada PC.

Berikut ini merupakan diagram alir diagram Program pengambilan data dari sensor oleh Mikrokontroler AT89C51 ke serial port.



Gambar 3.9. Diagram Alir Program pengambilan data Temperatur

Dari algoritma di atas, telah dibuat program untuk konversi ADC menggunakan bahasa Assembly. Listing Program dapat dilihat di bawah ini:

```

$mod52
$title(byte signed multiply)
$pagewidth(132)
$debug
$object
$nopaging
    ORG          2000H          ; Set Alamat ke alamat 2000H
    AJMP        Start          ; Loncat ke Rutin Start
    ORG          2003H          ; Set Alamat External INT0
    Reti
    ORG          200BH          ; Set Alamat Timer 0 Interrupt
    Reti
    ORG          2013H          ; Set Alamat External INT1
    Reti
    ORG          201BH          ; Set Alamat Timer 1 Interrupt
    Reti
    ORG          2023H          ; Serial I/O Interrupt
    Reti

Start:
    MOV         Dptr,#8000H ; ADC pd alamat 8000H
    ACALL      Init_Serial

Loop:
    ACALL      ADC           ; akses ADC
    LCALL      delay_5ms    ; beri delay
    INC        DPTR         ; ke channel berikut
    MOV        A,DPL        ; pindahkan nibel bawah ke a
    CJNE      A,#04H,Loop ; sudah sampai ke channel 4?
    LCALL      delay_1detik
    LCALL      delay_1detik
    AJMP      Start        ; Ulangi proses dari awal lagi

```

*; Rutin untuk mengakses ADC*

ADC:

```
MOV      A,#00H      ; Mulai untuk start konversi
MOVX    @DPTR,A      ; Salin Acc A ke Data Pointer
```

TungguEOC:

```
JB      INT0,TungguEOC ; Tunggu EOC
MOV     A,#00H        ; Baca hasil data konversi
MOVC   A,@A+DPTR
ACALL  ASCII_OUT     ; Kirim ke serial output
Ret
```

*; Rutin untuk delay*

Delay:

```
PUSH   B
MOV    B,#0FFH
DJNZ  B,$
POP   B
Ret
```

*; Rutin untuk Inisialisasi Mode Serial*

Init\_Serial:

```
MOV     SCON,#52H      ; Mode 1 Ren
MOV     TMOD,#20H     ; T0 Mode 2, T1 Mode 2
MOV     TH1,#0FDH     ; Set Baudrate 9600
MOV     TCON,#040H    ; T1 On, T0 Off
Ret
```

*; Rutin untuk mengirim data ke register SBUF ke port serial*

Serial\_Out:

```
Clr     TI
Mov     SBUF,A
```

TungguKirim:

```
Jnb    TI,TungguKirim
Ret
```



*; Rutin untuk konvert ADC dlm heksa ke ASCII*

ASCII\_Out:

```

ACALL    Hex_ASCII2 ;panggil rutin hex_ascii2
ACALL    Serial_Out ;panggil  rutin  serial_out
XCH      A,B        ;tukar data A dan B
RET

```

Hex\_ASCII2:

```

PUSH     ACC        ; Simpan Acc A ke dalam SP
ACALL    Hex_ASCII  ; panggil rutin hex_ascii
MOV      B,A        ; salin isi reg A ke B
POP      ACC        ; Ambil isi dari Stack Pointer
SWAP     A          ; tukar
ACALL    Hex_ASCII
Ret

```

Hex\_ASCII:

```

ANL      A,#0FH
CJNE     A,#10,LihatCarry

```

LihatCarry:

```

JNC      Bukan_Angka ;
Add      A,#30
Ret

```

Bukan\_Angka:

```

ADD      A,#37H
Ret
END

```

Urutan kerja diagram alir (*flowchart*) diatas adalah sebagai berikut :

1. Setelah inisialisasi alamat awal untuk program, yaitu alamat 2000H yang digunakan pada development system, maka program akan loncat ke rutin Start.

```
MOV    Dptr,#8000H
ACALL  Init_Serial
```

Alamat DPTR pada ADC di set pada alamat 8000H, kemudian program menuju rutin Init\_Serial. Adapun rutin Init\_Serial sebagai berikut :

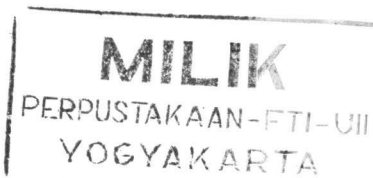
```
Init_Serial:
MOV    SCON,#52H           ; Mode 1 Ren
MOV    TMOD,#20H          ; T0 Mode 2
MOV    TH1,#0FDH         ; Set Baudrate 9600
MOV    TCON,#040H        ; T1 On, T0 Off
Ret
```

- Setelah inisialisasi untuk komunikasi dengan port selesai, maka program akan mengerjakan rutin utama yaitu proses konversi nilai-nilai masukan tiap-tiap sensor menjadi besaran digital. Setelah proses konversi dilakukan maka akan dilakukan pengecekan apakah proses konversi ke empat masukan (*channel*) telah selesai jika belum maka proses konversi berlanjut pada *channel* berikutnya. Pengecekan dilakukan dengan membandingkan alamat DPL pada DPTR dengan angka 4.

```
Loop:
Acall  ADC                ;Akses ADC
Lcall  delay_5ms          ;
Inc    Dptr               ;Tunjuk ke channel berikut
Mov    A,DPL              ;
Cjne   A,#04H,Loop       ;Channel 4? Belum Loop
Lcall  delay_1detik
Lcall  Delay_1detik
Ajmp   Start              ;Ulang dari awal
```

Proses Konversi ADC adalah sebagai berikut :

```
Mov    A,#00H            ;Start Conversion
Movx   @Dptr,A           ;Salin ke DPTR
TungguEOC:                ;Tunggu hingga INTO=1
Jb     INTO,TungguEOC
Acall  Delay
Mov    A,#00H            ;Read Convert Data
Movx   A,@DPTR
Lcall  ASCII_Out         ;Send to Serial Out
Ret
```



3. Setelah proses konversi ADC berakhir yang ditandai dengan EOC yang bernilai 1, maka hasil konversi tersebut yang berupa bilangan heksa dikonversikan menjadi ASCII. Hal ini disebabkan Program Komputer menerima kiriman data dari port serial berupa data ASCII.

```
ASCII_Out:
  ACALL    Hex_ASCII2    ;panggil rutin hex_ascii2
  ACALL    Serial_Out    ;panggil  rutin  serial_out
  XCH      A,B           ;tukar data A dan B
  RET
```

```
Hex_ASCII2:
  PUSH     ACC           ; Simpan Acc A ke dalam SP
  ACALL    Hex_ASCII    ; panggil rutin hex_ascii
  MOV      B,A          ; salin isi reg A ke B
  POP      ACC          ; Ambil isi dari Stack Pointer
  SWAP     A            ; tukar
  ACALL    Hex_ASCII
  Ret
```

```
Hex_ASCII:
  ANL      A,#0FH
  CJNE     A,#10,LihatCarry
```

```
LihatCarry:
  JNC      Bukan_Angka ;
  Add      A,#30
  Ret
```

4. Setelah dikonversi ke dalam ASCII maka data siap untuk dikirimkan.

Rutin untuk mengirim data adalah sebagai berikut:

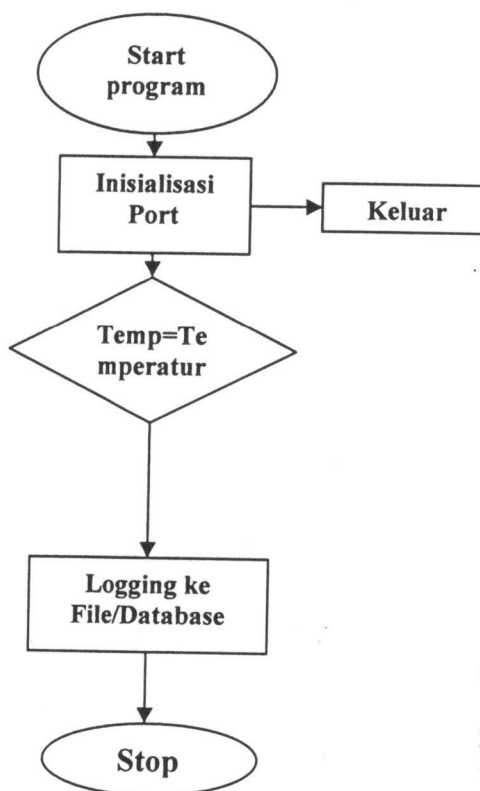
```
Serial_Out:
  Clr      TI
  Mov      SBUF,A
TungguKirim:
  Jnb      TI,TungguKirim
  Ret
```

Yang pertama dilakukan adalah mengosongkan isi register TI, kemudian memindahkan isi register SBUF ke register A, proses pengiriman data ditandai dengan isi register TI yang bernilai 1.

### 3.3.2. Perancangan Program Monitoring dengan Visual Basic (VB 6.0)

Pada program kedua yang dibuat menggunakan software Visual Basic 6.0, bertugas untuk menerima data-data yang dikirim dari mikrokontroller dari port serial setiap detiknya, kemudian menampilkan dalam sebuah main window, menyimpan data temperatur harian tersebut dalam sebuah file dan membuat grafik atau plot temperatur dalam periode 24 jam. Program Monitoring/Logger yang kita rancang akan memiliki fungsi-fungsi sebagai berikut:

- a. Menampilkan informasi temperatur ruangan (4 buah ruangan).
- b. Fasilitas *logging* berupa data : temperatur dan waktu ke dalam file.



**Gambar 3.10. Diagram alir program monitoring/logger**

Bahasa Visual Basic di pilih karena kemudahannya, dan karena tersedianya dukungan *library*. Untuk dapat berkomunikasi dengan port serial, Visual Basic telah menyediakan komponen MsComm. Untuk mengaktifkannya hanya diperlukan pemilihan port serial dan pendefinisian beberapa variabel seperti pada potongan kode di bawah ini :

```

Private Sub CmdOpen_Click()
If MSComm1.PortOpen = True Then
Exit Sub
End If
MSComm1.CommPort = 1
MSComm1.Settings = "9600,n,8,1"
MSComm1.RThreshold = 2
'MSComm1.InputLen = 8
MSComm1.InBufferSize = 8
MSComm1.PortOpen = True
  
```

End Sub

Sedangkan fungsi untuk *logging* dilakukan oleh kode di bawah ini, terlebih dahulu operator membuat folder tempat penyimpanan *file-file logging*, untuk sistem digunakan `c:\logger\templog`.

```
Private Sub Form_Load()  
file = "c:\logger\templog\" & Date$ & ".txt"  
Open file For Append As #1
```

Untuk menjalankan Program Monitoring, yang pertama kali dilakukan adalah menekan tombol Connect, yang mengaktifkan fungsi untuk memulai koneksi dengan port serial, selama program dijalankan, semua program yang mengakses port serial seperti HyperTerminal dan sebagainya tidak boleh digunakan karena hanya boleh satu program yang mengakses port serial. Jika proses koneksi berhasil maka akan tercetak "*Connected*" pada label program dan Gambar Status akan menyala kuning. Jika proses gagal atau program monitoring di stop maka akan tampil keterangan "*Disconnected*" dan gambar status akan kembali berwarna abu-abu. Saat program sudah terkoneksi, maka data-data dari mikrokontroller akan di baca dan di konversi dan di tampilkan di program.

## BAB IV

### ANALISIS DAN PEMBAHASAN

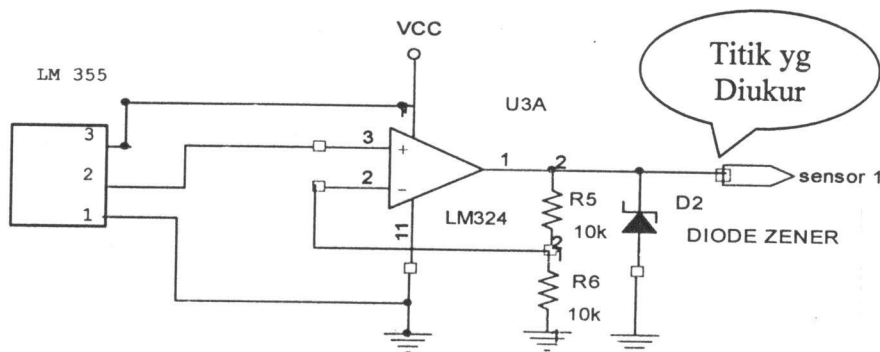
#### 4.1. PENGUKURAN

Pada bab ini akan dijelaskan mengenai pengujian terhadap keseluruhan sistem. Sistem yang diuji adalah temperatur ruangan yang data-datanya akan dikirim ke Program Monitor/Logger melalui port serial.

Pengujian sistem dilakukan dengan menghidupkan lampu pijar 40 watt sebagai sumber energi panas. Setelah itu dilakukan pengukuran beberapa tahap dengan mengukur beberapa titik *output*, antara lain *output sensor* LM35DT, *output* dari penguat IC LM324, *output* dari ADC 0809 dan data dari Program *Logger*. Hasil pengukuran akan dibandingkan dengan termometer ruangan untuk mengetahui sejauh mana sistem dapat berkerja.

##### 4.1.1 Pengukuran Terhadap Output Sensor Suhu

Pada Sensor LM35DZ akan mengalami perubahan tegangan tiap satu derajat celsius sebesar 10 mV.

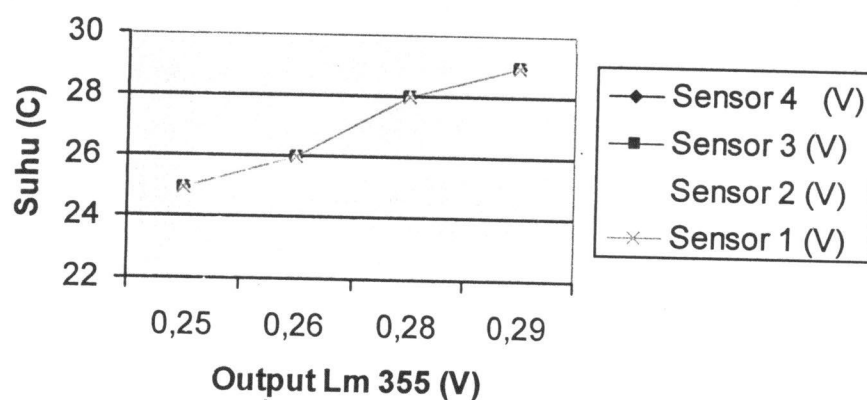


Gambar 4.1 Rangkaian Sensor Suhu LM35

Pada gambar 4.1 titik yang akan diamati pada rangkaian sensor adalah pada titik input IC LM324. Dalam mengamati perubahan suhu, digunakan termometer ruangan untuk mengkalibrasi suhu ruang dengan suhu yang terukur pada sensor suhu LM35. Pengukuran pertama dilakukan saat termometer menunjukkan 25°C pada suhu kamar, kemudian dilakukan pengukuran tegangan pada tiap-tiap sensor. Hasil pengamatan untuk tiap-tiap sensor terlihat pada tabel berikut.

**Tabel 4.1 Hasil pengamatan pada kaki sensor LM35**

Suhu Termometer (°C)	Sensor 1 (V)	Sensor 2 (V)	Sensor 3 (V)	Sensor 4 (V)
25	0,25	0,25	0,26	0,25
26	0,26	0,26	0,25	0,25
28	0,28	0,28	0,28	0,28
29	0,29	0,29	0,29	0,29



**Gambar 4.2 Grafik Suhu Terhadap Tegangan Ouput LM355**



Dengan demikian output LM35 dengan konfigurasi rangkaian seperti gambar 4.1 yang diukur sesuai dengan yang diharapkan dalam perancangan.

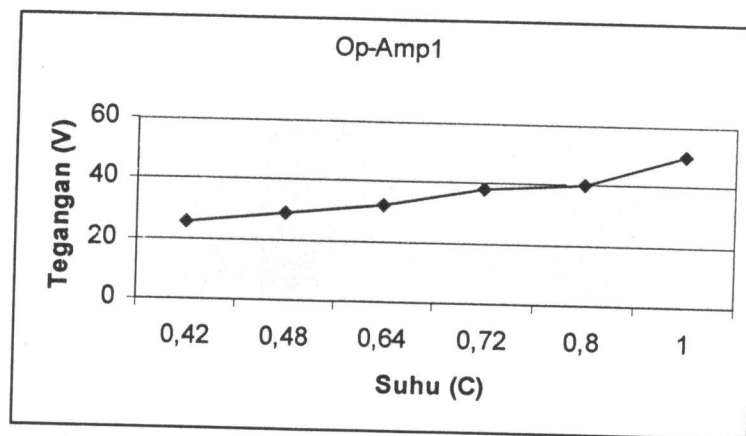
#### 4.1.2 Pengukuran Terhadap Pengukuran Output Penguat LM324

Penguatan yang diperoleh dari Operasional Amplifier (Op Amp) seperti pada rangkaian gambar 4.1 adalah 2 kali. Pengukuran dilakukan pada kaki-kaki atau pin output LM324. Dengan memberi gangguan pada sensor berupa lampu 40 watt, maka akan terjadi perubahan tegangan yang dapat diamati untuk tiap-tiap sensornya.

**Tabel 4.2 Hasil Pengamatan Pada Sensor Suhu dan OP-AMP**

Suhu Termometer (C)	Keluaran Sensor (V)				Keluaran OP-AMP (V)			
	1	2	3	4	1	2	3	4
26	0,26	0,26	0,26	0,26	0,42	0,42	0,42	0,42
29	0,29	0,29	0,29	0,29	0,48	0,48	0,48	0,48
32	0,32	0,32	0,32	0,32	0,64	0,64	0,64	0,64
38	0,38	0,38	0,38	0,38	0,76	0,76	0,76	0,76
40	0,40	0,41	0,41	0,40	0,80	0,80	0,82	0,80
50	0,50	0,50	0,50	0,50	1,00	1,00	1,00	1,00

Dari data hasil pengukuran dapat diketahui bahwa rangkaian sensor dan penguat telah berjalan sesuai dengan rancangan.



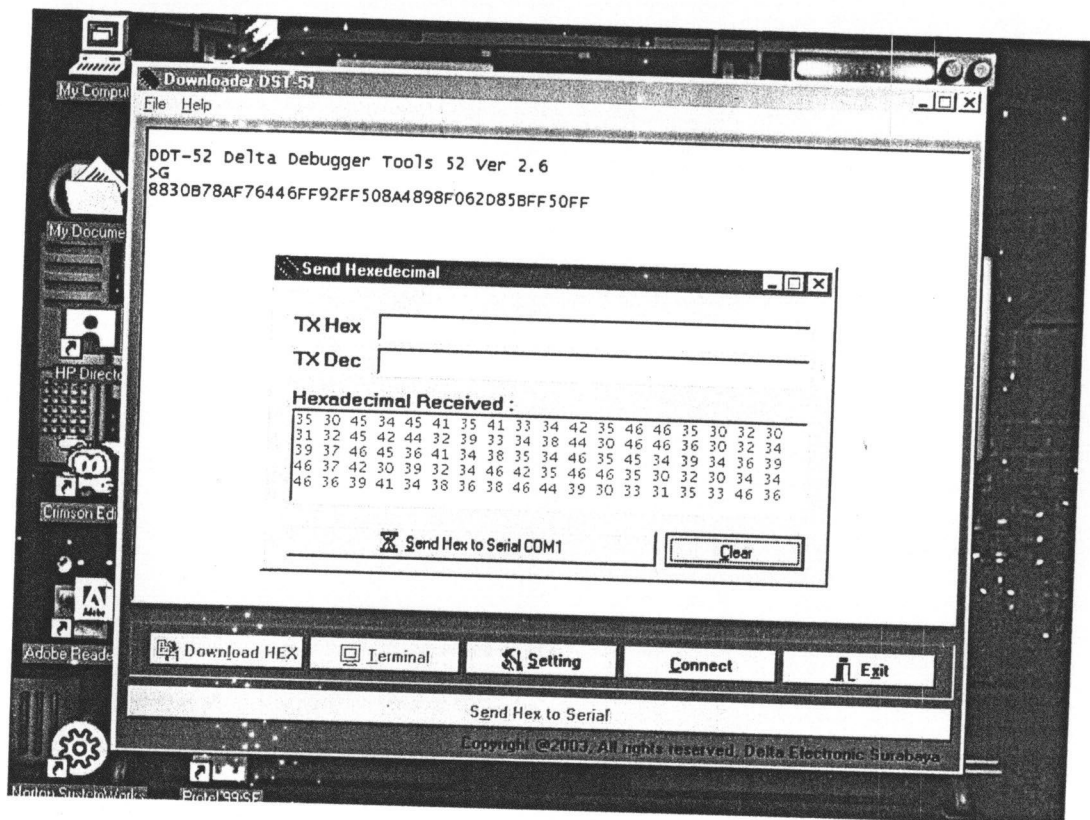
**Gambar 4.3 Grafik Suhu terhadap output Tegangan Penguat**

#### 4.1.3 Output ADC 0809

Tegangan yang terukur pada rangkaian sensor suhu dan penguat ketika dikonversikan menjadi sinyal digital haruslah tepat. Untuk mengetahui apakah data-data hasil konversi tepat, maka mikrokontroler diprogram untuk mengirimkan data-data hasil konversi ke port serial. Program Downloader dapat digunakan untuk menerima kiriman data dari mikrokontroler melalui port serial. Yang harus diingat adalah dalam konversi sinyal analog ke digital, mikrokontroler akan menyimpan data dalam bilangan heksa sedangkan untuk komunikasi serial, data dari mikrokontroler harus diubah terlebih dahulu ke dalam bentuk ASCII. Namun dari program downloader tersedia pula program untuk mengecek kiriman data dalam bentuk heksa walaupun program tersebut hanya berfungsi untuk melihat kiriman data dalam bentuk heksa tanpa bisa dipilah tiap masukannya, sehingga kita tidak dapat mengetahui dari sensor seberapa data yang muncul.

Dari masalah tersebut, maka cara menguji apakah kiriman data sudah sesuai maka kita hanya perlu mencocokkan data temperatur sensor dengan data

yang dikirim ke program terminal, karena telah keluaran sensor telah mengalami penguatan 2 kali maka data yang tampil di program terminal pun harus sesuai dengan data yang terukur di keluaran sensor. Dari hasil pengukuran terhadap tegangan sensor, tegangan penguat dan pembacaan data dari ADC melalui program Downloader sudah terdapat kesesuaian, hanya saja data yang tampil tidak dapat konsisten muncul namun bergerak dalam range  $\pm 4$  °C. Hal ini kemungkinan besar dipengaruhi oleh ketidakakuratan modul ADC dalam melakukan proses konversi.



Gambar 4.4 Data Suhu dalam Heksa

**Tabel 4.3 Hasil Pengamatan Suhu Terhadap Tegangan Keluaran  
Sensor, Tegangan OP-AMP dan ADC**

Suhu (C)	Keluaran Sensor (V)				Keluaran OP-AMP (V)				ADC (C) (ASCII)			
	1	2	3	4	1	2	3	4	1	2	3	4
26	0,26	0,26	0,26	0,26	0,52	0,52	0,52	0,52	52	53	52	51
29	0,29	0,29	0,28	0,28	0,58	0,58	0,58	0,58	58	59	56	57
32	0,32	0,32	0,32	0,32	0,64	0,64	0,64	0,64	64	64	64	61
38	0,38	0,38	0,38	0,38	0,72	0,76	0,76	0,76	76	76	75	76
40	0,40	0,41	0,41	0,40	0,80	0,80	0,80	0,80	80	81	80	80

#### 4.1.4 Pengujian Terhadap Output Program Logger

Program mikrokontroller akan mengirimkan data dalam bentuk ASCII. Data yang tampak dari program HyperTerminal terlihat sebagai rangkaian karakter 8 bit seperti : 8830B78A. Program pada Visual Basic akan membaca data dari buffer dan kemudian melakukan *parsing* untuk memecah data menjadi 2 digit yang mewakili tiap channel. Potongan kode berikut merupakan inisialisasi serial port.

```

MSComm1.CommPort = 1
MSComm1.Settings = "9600,n,8,1"
MSComm1.RThreshold = 2
MSComm1.InputLen = 8
MSComm1.InBufferSize = 8
MSComm1.PortOpen = True

```

Untuk mengambil data dari buffer dan proses pemsihannya dapat dilihat pada program berikut :

```

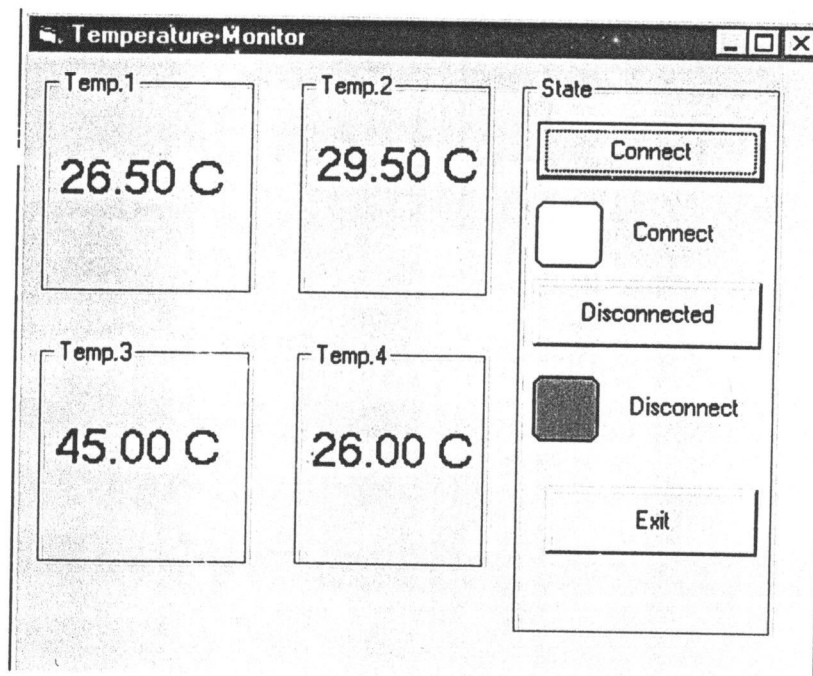
Do
    dummy = DoEvents()
    Loop Until MSComm1.InBufferCount = 8
Select Case MSComm1.CommEvent
Case comEvReceive
MSComm1.InputLen = 8
Buffer = MSComm1.Input
If (Buffer) = "" Then Buffer = Chr(0)
Data = (Buffer)

temp1 = (Mid(Data, 1, 2))
temp1 = ((Val(temp1)) / 2 + 3)
temp2 = (Mid(Data, 3, 2))
temp2 = ((Val(temp2)) / 2 + 3)
temp3 = (Mid(Data, 5, 2))
temp3 = ((Val(temp3)) / 2 + 3)
temp4 = (Mid(Data, 7, 2))
temp4 = ((Val(temp4)) / 2 + 3)

```

Data-data yang diambil oleh ADC adalah berupa data tegangan dalam bentuk hexa sedangkan data yang akan ditampilkan ke komputer adalah berupa suhu dalam bentuk desimal. Pertama-tama perangkat lunak pada Modul DST-51 mengubah data tegangan menjadi data suhu. Proses konversi dilakukan dengan menyesuaikan Vref (tegangan referensi) pada ADC dengan tegangan dari sensor. Karena Modul ADC memiliki tegangan referensi sebesar 5 volt untuk 8 bit data jadi setiap kenaikan satu bit adalah 0.019607 atau sama dengan 0.02 volt (pembulatan), sehingga program akan langsung membagi nilai data masukan dengan angka 2, penambahan dengan nilai 3 sebagai nilai koreksi agar pembacaan dari keluaran sensor sebanding dengan pembacaan data dari program maka proses perhitungan pada program monitor/logger sebagai berikut:

$$\text{Suhu} = \text{Data Masukan} / 2 + 3 \quad (4.1)$$



Gambar 4.5 Program Temperature Monitor/Logger

Tabel 4.4 Hasil Pengamatan pada Suhu dan Program Monitor dalam Celcius

Suhu	Sensor 1		Sensor 2		Sensor 3		Sensor 4	
	Suhu	Error	Suhu	Error	Suhu	Error	Suhu	Error
12	12	0%	12.50	3.75%	12	0%	12	0%
13	13	0%	13	0%	13	0%	13	0%
14	14.50	3.21%	15	6.42%	14	0%	14	0%
25	25	0%	25.50	1.8%	26.50	5.4%	23.50	5.4%
26	26.50	1.7%	26	0%	27	3.4%	26.50	1.7%
28	28	0%	27	3.2%	28	0%	26.50	4.8%
29	29	0%	30.50	4.6%	27	6.2%	28	3.1%
32	33	2.8%	31.50	1.4%	30	5.6%	33	2.8%

35	35	0%	35	0%	34	2.5%	36	2.5%
45	46	1.9%	45	0%	47	3.9%	44	1.9%

Error kesalahan didapat dari persamaan :

$$\text{Error} = \frac{(\text{SuhuTerukur} - \text{SuhuAcuan})}{\text{SuhuAcuan}} * c \quad (4.2)$$

Dimana c = koefisien kesalahan dengan nilai 0 hingga 1  
c = 0.9

Dari tabel diatas dapat kita amati nilai rata-rata *error* tiap-tiap sensor adalah

- a. Sensor 1 sebesar : 0.961%
- b. Sensor 2 sebesar : 2.117 %
- c. Sensor 3 sebesar: 2.7 %
- d. Sensor 4 sebesar: 2.22 %

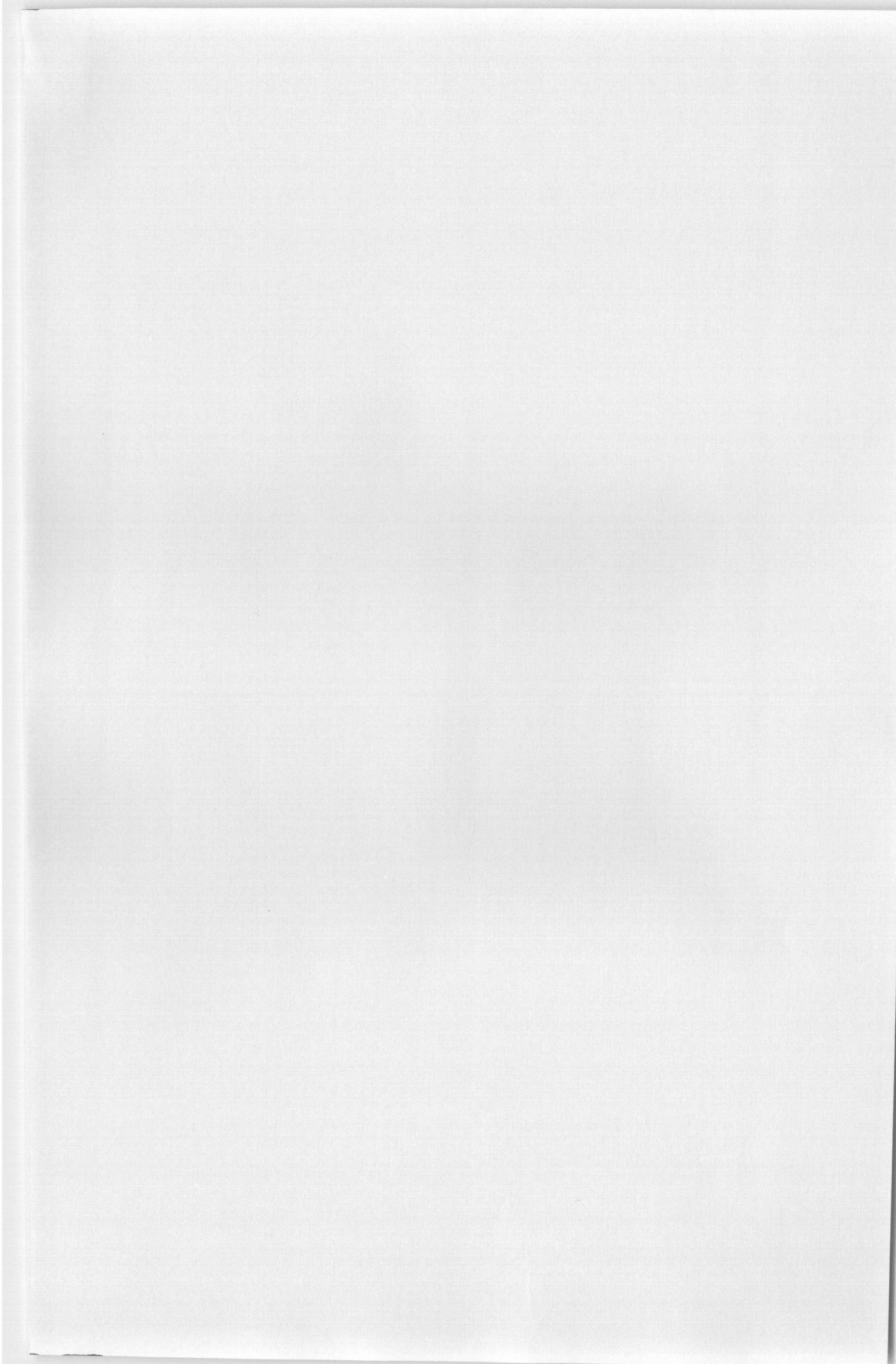
#### 4.2 PEMBAHASAN

Program monitor/logger sebenarnya sudah berjalan baik dengan mampu menampilkan data suhu yang terukur dan mampu menampilkan perubahan suhu yang terjadi, walaupun masih memiliki kekurangan berupa data yang tampil tidak bisa tetap melainkan selalu berubah dalam jangkauan  $\pm 4^{\circ}\text{C}$  dan juga terdapat kiriman data ASCII yang tidak berupa angka desimal saja namun juga kombinasi angka dan huruf yang tidak dapat dikonversi oleh program dengan baik. Namun dikarenakan konversi data dari modul ADC tidak sepenuhnya akurat pula maka program hanya mampu menampilkan data dengan tingkan kesalahan sebesar  $\pm 4^{\circ}\text{C}$  .Tentu saja hal ini tidak diharapkan, karena untuk sistem pemantauan

haruslah mampu menampilkan data-data yang diamati secara akurat dengan tingkat kesalahan yang rendah. Namun program monitor sendiri sudah dapat membaca data-data yang dikirimkan oleh mikrokontroller dan melakukan pemisahan data-data tersebut untuk setiap kanal-kanalnya. Perlu perbaikan pada modul ADC agar dapat melakukan proses konversi secara tepat. Untuk modul ADC sendiri diperlukan rangkaian *oscillator* yang lebih baik

Proses konversi dari modul ADC sebenarnya berjalan secara terus menerus dalam waktu yang cepat dalam orde mikro detik, namun program monitor tentunya tidak dapat menampilkan dalam waktu yang sama karena akan terlihat terlalu cepat dan tidak dapat diamati. Karenanya dalam program mikrokontroller telah dibuat dua buah delay, masing-masing adalah delay satu detik dan delay 500 milidetik. Delay 500 milidetik digunakan untuk delay perpindahan setiap kanal sedangkan delay 1 detik digunakan untuk delay tiap satu kali proses konversi. Dengan demikian program monitor akan menerima kiriman data setiap dua detiknya tanpa harus membuat fungsi lagi di dalam programnya karena telah dikerjakan di dalam program mikrokontroller.





## BAB V

### KESIMPULAN DAN SARAN

#### 5.1. Kesimpulan

Berdasarkan hasil pengujian dari sistem logger temperatur ruangan berbasis mikrokontroller maka dapat diambil kesimpulan sebagai berikut:

1. Program monitor/logger sudah dapat menampilkan temperatur 4 ruangan yang terpisah yang memiliki temperatur berbeda dan melakukan aktivitas pencatatan setiap 2 detik.
2. Sistem pemantau temperatur ruangan yang dibuat mampu berkerja pada *range* suhu 2°C hingga 150°C, namun begitu suhu maksimal yang telah dicoba adalah 45 – 50 °C dan suhu terendah adalah 12 °C.
3. Sistem pemantauan yang dibuat telah di uji coba pada jarak  $\pm$  90 cm dengan rincian 30 cm untuk kabel sensor dan 60 cm untuk kabel serial. Perlu dilakukan pengujian lebih lanjut untuk jarak lebih dari 90 cm.
4. Diperlukan teknik pemisahan data (algoritma) yang lebih baik jika ingin menambah jumlah kanal lebih dari 4 buah.
5. Tingkat kesalahan yang terjadi untuk tiap sensor ( Sensor 1:0.961%; Sensor 2: 2.117%; Sensor 3:2.7%; Sensor 4: 2.2%)

#### 5.2 Saran

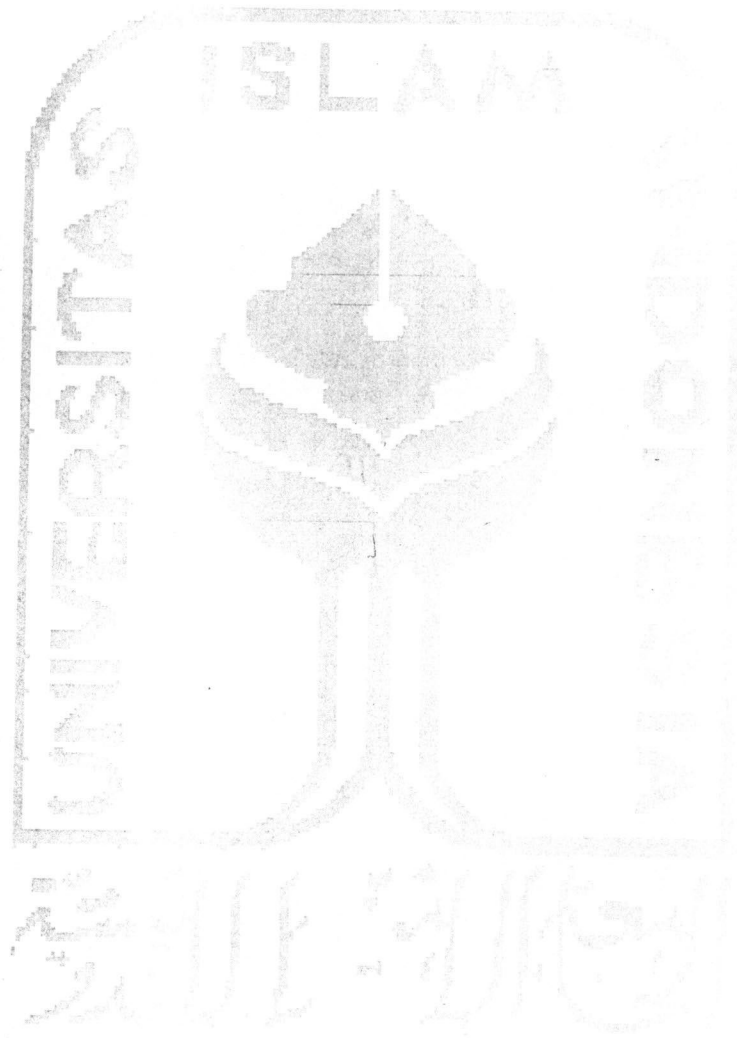
1. Diperlukan perbaikan pada modul ADC terutama pada rangkaian *oscilator* agar proses konversi dapat berjalan baik.
2. Untuk penelitian selanjutnya dapat digunakan mikrokontroller dari ATMEL yang memiliki ADC semisal mikrokontroller AT83C511.
3. Sistem monitor/logger dapat dikembangkan menjadi sistem telemetri dengan menggunakan gelombang radio.



## DAFTAR PUSTAKA

- Andi Nalwan,Paulus, *Teknik Antarmuka dan Pemograman Mikrokontroler AT89C51*, Elex Media Komputindo, Jakarta, 2003.
- Budiharto,Widodo, S.Si, M.Kom, *Interfacing Komputer dan Mikrokontroler*, PT. Elex Media Komputindo, Jakarta, 2004.
- Eko Putra,Agfianto, *Belajar Mikrokontroler AT89C51/52/55 Teori dan Aplikasi*, Gava Media, Jogjakarta, 2002.
- Eko Putra,Agfianto, *Penapis Aktif Elektronika*, Gava Media, Jogjakarta, 2002.
- F. Coughlin,Robert; F. Driscoll,Frederick; Widodo Soemitro,Herman, *Penguat Operasional dan Rangkaian Terpadu Linear*, Erlangga, Jakarta, 1992
- M. Barmawi,Millman, Halkias. M.O Tija. *Elektronika Terpadu “ Rangkaian dan Sistem Analog dan Digital” (Jilid 2)*, Erlangga, Jakarta, 1985.
- Paul Malvino,Albert, Ph.D; Barmawi,M; O. Tija,M, *Prinsip-Prinsip Elektronika (Edisi ke Tiga)*, Erlangga, Jakarta, 1987.
- Atmel 89C51 *technical data*, Atmel [www.atmel.com](http://www.atmel.com)

LAMPIRAN





(BYTE SIGNED MULTIPLY)  
DTH(132)  
BOLS  
?  
ING

\*\*\*\*\*  
4 CHANNEL ADC DATA (ADC0808/0809) OLEH DST-52 DAN KIRIM KE SERIAL\*  
\*  
\*

isters: \*  
DPTR : DPH.5 ... DPH.7 --> ADC Address \*  
: DPL.0 ... DPL.2 --> ADC Channel \*  
A : Data \*  
B : Delay \*  
\*  
\*\*\*\*\*

ut EQU 00CBH  
rial EQU 00ECH  
Out EQU 00FCH  
detik EQU 0090h

2000H  
p Start  
2003H ;External INTO  
i  
200BH ;Timer 0 Interrupt  
i  
2013H ;External INT1  
i  
201BH ;Timer 1 Interrupt  
i  
2023H ;Serial I/O Interrupt

Dptr,#8000H ;ADC pada alamat 800XH, CS4 pada DST-51  
1 Init\_Serial ;Inisial Serial Port

1 ADC ;Akses ADC  
1 Delay ;  
A,#' ' ;  
1 Serial\_Out ;Kirim Spasi  
Dptr ;Tunjuk ke channel berikut  
A,DPL ;  
A,#04H,Loop ;Channel 4? Belum Loop  
1 Delay\_1detik  
Start ;Kembali ke Start

A,#00H ;Start Conversion  
@Dptr,A ;  
C: INT0,TungguEOC ;Tunggu EOC  
1 Delay  
A,#00H ;Read Convert Data  
A,@A+Dptr ;  
1 ASCII\_Out ;Send to Serial Out

B  
B,#0FFH  
B,Delay  
3

Org \*

I\_Out:

```
ACALL hex_ascii2
ACALL Serial_Out
XCH  A,B
ACALL Serial_Out
RET
```

DPTR

```
ACALL Enter_Code
MOV  A,DPH
ACALL ASCII_Out
MOV  A,DPL
ACALL ASCII_Out
RET
```

c\_Code:

```
PUSH A
MOV  A,#00Dh
ACALL Serial_Out
MOV  A,#00Ah
ACALL Serial_Out
POP  A
RET
```

Serial:

```
OV  SCON,#52H      ; Mode 1 Ren
OV  TMOD,#20H     ; T0 Mode 2, T1 Mode 2
OV  TH1,#0FDH    ; 9600 Baudrate
OV  TCON,#040H   ; T1 On, T0 Off
OV  PCON,#00H    ;
ET
```

l\_Out:

```
CLR  TI
MOV  SBUF,A
JNB  TI,*
RET
```

l\_In:

```
CLR  RI
JNB  RI,*
MOV  A,SBUF
RET
```

ing\_2Ser:

```
MOV  A,#000h
MOVC A,@A+DPTR
CJNE A,#00Fh,Send_String
RET
```

tring:

```
ACALL Serial_Out
INC  DPTR
AJMP OutString_2Ser
```

```

*****
ROUTINE KONVERSI HEXA KE ASCII
*****

```

```

-----
: A = Angka -> tambah 30H
: A <> Angka -> tambah 37H

```

```

      Org *
ASCII2:
Push  A          ;Simpan Acc A ke SP
Acall Hex_ASCII1 ;Konversi 1 nibble
Mov  B,A        ;Simpan nibble bawah di Reg B
Pop  A          ;Ambil Acc A dari SP
Swap A          ;Tukar
Acall Hex_ASCII1 ;Konversi 1 nibble
Ret

```

```

ASCII1:
Anl  A,#0FH      ;Hapus Nibble Atas
Cjne A,#10,*+3   ;Acc A = 10 dan carry 0 -> bukan angka
                    ;Acc A <>10 dan carry 0 -> bukan angka
Jnc  Bukan_Angka ;Acc A <>10 dan carry 1 -> tambah 30H
Add  A,#30H
Ret

```

```

; Angka:
Add  A,#37H
Ret

```



```

DSEG
Org      $
r_5ms:   Ds      1          ;Jumlah delay 5mS yg terjadi

```

```

CSEG
Org      $
*****
ay 5 mS sebanyak 200 x

```

```

1detik:
v Counter_5mS,#0200

```

```

_1detik:
all Delay_5mS
iz Counter_5mS,Tunggu_1detik
:

```

```

****
y 5 mS sebanyak 100 x

```

```

00mS:
Counter_5mS,#0100

```

```

500mS:
ll Delay_5mS
z Counter_5mS,Tunggu_500mS

```

```

*****
/ 5 mS sebanyak 20 x

```

```

00mS:
Counter_5mS,#020

```

```

.00mS:
l Delay_5mS
Counter_5mS,Tunggu_100mS

```

```

****
5 mS sebanyak 15 x

```

```

mS:
Counter_5mS,#015

```

```

5mS:
l Delay_5mS
Counter_5mS,Tunggu_75mS

```

```

*****
ini bekerja hanya pada crystal 11.0592 MHz

```

```

:
TMOD
MOD,#21H ;Timer Mode 16 bit counter
HO,#0EDH
LO,#0FFH
TRO

```

```

S:
FO,Sudah_5mS
Tunggu_5mS

```

```

:
RO
IOD

```

Untuk Proses Koneksi ke Port Serial

Sub CmdOpen\_Click()

mscmm1.PortOpen = True Then

End Sub

Agar untuk komunikasi serial

.CommPort = 1

.Settings = "9600,n,8,1"

.RThreshold = 2

.InBufferSize = 8

.PortOpen = True

Untuk proses disconnect

Sub CmdDisc\_Click()

mscmm1.PortOpen = False Then

End Sub

PortOpen = False

Untuk Keluar

Sub CmdExit\_Click()

Sub Form\_Load()

FileOpen("c:\templog\" & Date\$ & ".txt",

OpenMode:=OpenMode.Append)

Sub Timer1\_Timer()

mscmm1.PortOpen = False Then

Label1.FillColor = &H808080

End Sub

Label1.FillColor = &HFFFFFF

Sub MSCComm1\_OnComm()

Dim Data As String

Dim temp1 As String

Dim temp2 As String

Dim temp3 As String

Dim temp4 As String

Dim temp5 As String

Dim temp6 As String

Dim temp7 As String

Dim temp8 As String

DoEvents

Until MSCComm1.InBufferCount = 8

MSCComm1.CommEvent

MSCComm1.Receive

MSCComm1.Input

MSCComm1.Buffer = Chr(0)

MSCComm1.Buffer

MSCComm1.Buffer di buffer dan pecah menjadi 2 byte

Mid(Data, 1, 2)

(Val(temp1) / 2 + 3)

Mid(Data, 3, 2)

(Val(temp2) / 2 + 3)

Mid(Data, 5, 2)

(Val(temp3) / 2 + 3)

Mid(Data, 7, 2)

(Val(temp4) / 2 + 3)

Label1.Caption = Data

Label1.Caption = Format(temp1, "##0.00" & " C")

Label1.Caption = Format(temp2, "##0.00" & " C")

Label1.Caption = Format(temp3, "##0.00" & " C")

Label1.Caption = Format(temp4, "##0.00" & " C")

Label1.Caption

Label1.Caption

Label1.Caption = Format(temp1, "##0.00"), Format(temp2, "##0.00"), Format(temp3, "##0.00"), Format(temp4, "##0.00"), Time

Label1.Caption