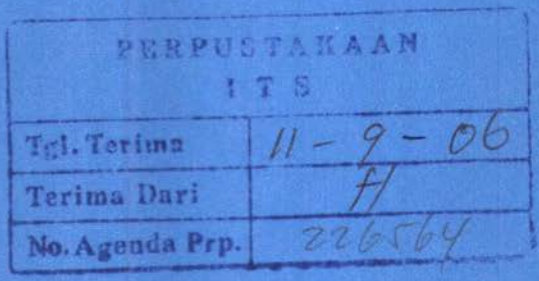
27650/H/2006

ITS
Institut
Teknologi
Sepuluh Nopember

RSF
681.2
Kus
r-1
2006



FINAL PROJECT - RF 0469

DESIGN and BUILD THE PORTABLE POCKET OF MEASURING INSTRUMENT of PRESSURE BASED ON MICROCONTROLLER

NATRIA IKE KUSDIANINGSIH
NRP 2403 030 025

Counsellor
Ir. Ali Musyafa', MSc.

DIPLOMA III OF INSTRUMENTATION ENGINEERING
ENGINEERING PHYSICS DEPARTMENT
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2006

RANCANG BANGUN ALAT UKUR TEKANAN "PORTABLE" BERBASIS MIKROKONTROLER

TUGAS AKHIR

Oleh :
Natria Ike Kusdianingsih
NRP. 2403 030 025

Surabaya, Juli 2006
Mengetahui / Menyetujui

Pembimbing I	Pembimbing II
Ir. Ali Musyafa',MSc	**Fitri Adi Iskandarianto,ST**
NIP. 131 652 210	**NIP. 132 318 296**
Ketua Jurusan	**Ketua Program Studi**
Teknik Fisika FTI – ITS	**DIII Instrumentasi**
DR, Ir. Totok Suhartanto, DEA	**Ir. Matradji, MSc**
NIP. 131 879 399	**NIP. 131 478 882**

RANCANG BANGUN ALAT UKUR TEKANAN "PORTABLE" BERBASIS MIKROKONTROLER

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Ahli Madya Diploma
pada
Bidang Studi Instrumentasi
Program Studi DIII Teknik Instrumentasi
Jurusan Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh:
NATRIA IKE KUSDIANINGSIH
NRP. 2403.030.025

Disetujui oleh Tim Penguji Tugas Akhir:

1. Ir. Ali Musyafa', MSc (Dosen Pembimbing)
2. Ir. SArwono.MM (Dosen Penguji I)
3. Dr. Ir. Sekartedjo,MSc (Dosen Penguji II)
4. Ridho Hantoro, ST, MT (Dosen Penguji III)

SURABAYA
JULI, 2006

RANCANG BANGUN ALAT UKUR TEKANAN "PORTABLE" BERBASIS MIKROKONTROLER

Nama mahasiswa	**: Natria Ike Kusdianingsih**
NRP	**: 2403.030.025**
Program Studi	**: DIII Teknik Instrumentasi**
Jurusan	**: Teknik Fisika FTI-ITS**
Dosen Pembimbing	**: Ir. Ali Musyafa', MSc**

Abstrak

Dari sekian banyak alat ukur sebagian besar menggunakan sensor yang universal dan mahal serta tampilan masih dalam analog. Bahkan alat ukur kebanyakan berukuran besar dan berat. Oleh karena itu pada penelitian ini akan dibuat sebuah alat ukur tekanan yang dikemas secara portable pocket dengan tampilan digital untuk kemudahan dalam pembacaan satuan tekanan yang terukur. Dengan menggunakan pressure gauge yang dimidifikasi dengan LDR dan led serta rangkaian pembagi tegangan yang dipasang plat hitam diantara LDR dn led. Dengan memanfaatkan prinsip kerja LDR maka dapat digunakan sebagai sensor tekanan. Untuk pengukuran dilakukan pada range pengukuran 0 – 15 Psi. Keunggulan dari alat ini adalah sensor yang dibuat sendiri dan tampilannya dua digit dibelakang koma. Untuk pengukuran didapatkan nilai uncertaint, dengan nilai pengukuran pada tekanan 0 Psi adalah 0 ± 0, nilai pengukuran pada tekanan 5 Psi adalah = 4,40 ± 0,37, nilai pengukuran pada tekanan 10 Psi adalah 9,44 ± 0,11, dan nilai pengukuran untuk tekanan 15 Psi adalah 15,02 ± 0,05.

Kata kunci: *portable pocket, sensor pressure gauge*

DESIGN and BUILD THE PORTABLE POCKET OF MEASURING INSTRUMENT of PRESSURE BASED ON MICROCONTROLLER

Name	**: Natria Ike Kusdianingsih**
Nrp	**: 2403.030.025**
Major	**: Engineering DIII Instrumentasi**
Counsellor	**: Ir. Ali Musyafa', MSc**

Abstracs

From so much measuring instrument most of using costly and universal censor and also appearance still in analogue. Even measuring instrument] most heavy and king sized. Therefore this research will be made a measuring instrument tidy pressure by portable pocket with the digital appearance for the amenity of in read set of pressure measured. By using pressure gauge which modification by LDR and led and also network of voltage divider attached a black plate among LDR and led. By exploiting principle work the LDR hence serve the purpose of sensor the pressure. For the measurement of conducted at range measurement 0 - 15 Psi. Excellence from this appliance is censor made byself and its appearance two digit behind comma. For the measurement of got value uncertainty, with the measurement value of pressure 0 Psi is 0 ± 0, measurement value at pressure 5 Psi is = 4,40 ± 0,37, measurement value at pressure 10 Psi is 9,44 ± 0,11, and assess the measurement for the pressure of 15 Psi is 15,02 ± 0,05.

***Keyword**: portable pocket, sensor the pressure gauge.*

KATA PENGANTAR

Syukur Alhamdulillah atas limpahan rahmat yang diberikan Allah SWT, yang maha agung dan maha bijaksana. Atas rahmat, petunjuk dan karunia-Nya lah sehingga penulis mampu melaksanakan dan menyelesaikan Tugas Akhir dengan judul

RANCANG BANGUN ALAT UKUR TEKANAN "PORTABLE" BERBASIS MIKROKONTROLER

Dalam penulisan laporan ini, penulis menyadari masih terdapat begitu banyak kekurangan yang tentunya membutuhkan kritik dan saran sehingga kekurangan yang ada dapat diperbaiki.

Penulis mengucapkan banyak terimakasih atas segala dukungan yang diberikan, sehingga penulis tetap semangat dalam melaksanakan Tugas Akhir. Terimakasih penulis ucapkan kepada:

1. Bapak Dr. Ir. Totok Suhartanto, DEA, selaku Ketua jurusan Teknik Fisika ITS
2. Bapak Matradji, MSc selaku Ketua Program Studi DIII Teknik Instrumentasi jurusan Teknik Fisika ITS.
3. Bapak Ir. Ali Musyafa', MSc selaku dosen pembimbing, terima kasih atas saran dan bimbingannya.
4. Bapak Suyanto ST,MT, selaku Kepala Laboratorium Workshop Instrumentasi, serta bapak Fitri Adi Iskandarianto,ST, yang memberi ide dalam penyelesaian Tugas Akhir ini
5. Keluarga di Nganjuk, Bapak Ibu atas kasih sayang dan kepercayaan kepada adhek. Serta doa dan dukungan yang selalu mengalir dengan tulus. Mas Yupi, Mas Niko terima kasih atas dukungan moril maupun materiil.

Semoga laporan Tugas Akhir ini dapat berguna bagi penulis dan semua pihak yang mempelajarinya.

Surabaya, Juli 2006
Penulis

DAFTAR ISI

BAB		HALAMAN
HALAMAN JUDUL		i
HALAMAN PENGESAHAN		ii
ABSTRAK		iii
ABSTRACT		v
KATA PENGANTAR		vii
DAFTAR ISI		ix
DAFTAR GAMBAR		xi
DAFTAR TABEL		xiii
Bab I	PENDAHULUAN	
1.1	Latar Belakang	1
1.2	Permasalahan	2
1.3	Batasan Masalah	2
1.4	Tujuan dan Manfaat Penelitian	2
1.5	Sistematika Laporan	3
Bab II	DASAR TEORI	
2.1	Pressure Gauge	5
2.2	LDR	9
2.3	Rangkaian Pembagi	11
2.4	Rangkaian ADC	11
2.5	Mikrokontroler	13
2.6	Regulator	32
2.7	LCD 16 x 2	33
Bab III	PERANCANGAN ALAT	
3.1	Mekanisme Sistem Kerja	36
3.2	Perancangan Perangkat Keras (Hardsware)	38
3.3	Perancangan Perangkat Lunak (Software)	44
Bab IV	PENGUJIAN DAN ANALISA	
4.1	Pengukuran data	55
4.2	Pengambilan data sensor tekanan	57
4.3	Pengambilan data display	63

4.4	Pengambilan data tegangan sensor	69
4.5	Analisa Data	70
Bab V	PENUTUP	
5.1	Kesimpulan	73
5.2	Saran	73
DAFTAR PUSTAKA		75
LAMPIRAN		

DAFTAR GAMBAR

	HALAMAN
Gambar 2.1 Salah satu Bourdon Tube "C"	7
Gambar 2.2 Salah satu pressure gauge type helical	8
Gambar 2.3 Salah satu pressure gauge type spiral	8
Gambar 2.4 Pressure Gauge	9
Gambar 2.5 LDR	10
Gambar 2.6 Rangkaian Pembagi	11
Gambar 2.7 Pin ADC 0804	12
Gambar 2.8 Pin AT89S51	14
Gambar 2.9 Timing Diagram Mode 0 Serial	18
Gambar 2.10 Membaca ROM Eksternal	20
Gambar 2.11 Menulis ke RAM	20
Gambar 2.12 Membaca dari RAM	21
Gambar 2.13 Rangkaian Power ON Reset	21
Gambar 2.14 Rangkaian Kristal AT89S51	23
Gambar 2.15 IC Regulator 7805	32
Gambar 2.16 LCD 16x2	33
Gambar 3.1 Diagram Blok Sistem	35
Gambar 3.2 Flowchart mekanisme kerja sistem	37
Gambar 3.3 Rangkaian Power Supply	38
Gambar 3.4 Rangkaian Sensor	39
Gambar 3.5 Rangkaian pembagi	39
Gambar 3.6 Rangkaian ADC 0804	40
Gambar 3.7 Diagram blok LCD	42
Gambar 3.8 Timing diagram data write	43
Gambar 3.9 Timing diagram data read	44
Gambar 3.10 Flowchart Perancangan Software	46
Gambar 3.11 Kompile Progarm ASM	53
Gambar 3.12 Program Pengisian Mikro ATS51	54
Gambar 4.1 Grafik Histerisis	69
Gambar 4.2 Grafik hubungan tegangan - tekanan	70

DAFTAR TABEL

	HALAMAN
Tabel 2.1 Fungsi –Fungsi Khusus Pada AT 89S51	16
Tabel 2.2 Tabel Special Function Registers	24
Tabel 3.1 Jenis Kapasitas ROM Mikrokontroler Atmel	41
Tabel 3.2 Tabel fungsi pin LCD	43
Tabel 4.1 Data Pengukuran naik pressure gauge tekanan 0 Psi	57
Tabel 4.2 Data Pengukuran naik pressure gauge tekanan 5 Psi	58
Tabel 4.3 Data Pengukuran naik pressure gauge tekanan 10 Psi	59
Tabel 4.4 Data Pengukuran naik pressure gauge tekanan 15 Psi	59
Tabel 4.5 Data Pengukuran turun pressure gauge tekanan 15 Psi pada display	59
Tabel 4.6 Data Pengukuran turun pressure gauge tekanan 10Psi	61
Tabel 4.7 Data Pengukuran turun pressure gauge tekanan 5 Psi	61
Tabel 4.8 Data Pengukuran turun pressure gauge tekanan 0 Psi	61
Tabel 4.9 Data Pengukuran naik tekanan 0 Psi pada display	63
Tabel 4.10 Data Pengukuran naik tekanan 5 Psi pada display	64
Tabel 4.11 Data Pengukuran naik tekanan 10 Psi pada display	64
Tabel 4.12 Data Pengukuran naik tekanan 15 Psi pada display	65
Tabel 4.13 Data Pengukuran turun tekanan 15 Psi pada display	66
Tabel 4.14 Data Pengukuran turun tekanan 10 Psi pada display	66

Tabel 4.15 Data Pengukuran turun tekanan 5 Psi pada display 67
Tabel 4.13 Data Pengukuran turun tekanan 0 Psi pada display 68
Tabel 4.17 Pengukuran Sensor 69

BAB I
PENDAHULUAN

1.1 Latar Belakang

Untuk mengetahui sebuah nilai dari suatu alat maka diperlukan alat ukur yang mudah dimengerti. Hasil suatu pengukuran merupakan karakteristik dan resultan dari beberapa faktor yang mempengaruhi termasuk sarana pendukung tersebut dan mempunyai pengaruh terhadap bobot metode pengukuran yang baik dan benar akan membantu mengurangi adanya kesalahan dalam proses pengukuran untuk mendapatkan nilai yang benar. Dengan metode yang berbeda akan menghasilkan nilai yang berbeda pula meskipun kecil.

Alat ukur tidak dituntut sekecil mungkin atau dipandang dari segi fisiknya, tetapi apakah alat ukur tersebut memenuhi standardisasi pengukuran. Kebanyakan para pengguna alat ukur tidak hanya memerlukan keakuratan dari alat ukur, tetapi juga kenyamanan dalam pemakaian. Salah satunya adalah kenyamanan dalam membawa alat ukur kemana pengguna akan mengukur dan tampilan. Kebanyakan alat ukur berukuran besar dan tampilannya masih analog.

Dengan alasan seperti yang telah disebutkan diatas maka pada tugas akhir ini mekalukan penelitian pembuatan alat ukur tekanan yang dikemas secara *portable pocket* dengan tampilan digital berbasis mikrokontroler. Dikemas secara *portable pocket* ini agar dapat dibawa kemana-mana, sedangkan dengan tampilan digital akan mempermudah dalam pembacaan satuan terukur dengan menggunakan satuan standart. Dalam penelitian ini diharapkan adanya perkembangan dalam alat ukur tekanan dan bermanfaat bagi semua pengguna alat ukur.

1.2 Permasalahan

Permasalah dalam penelitian ini adalah :

- Bagaimana sensor pressure gauge yang telah dimodifikasi denga LDR dan led dapat dipakai untuk mengukur tenakan dan menampilkan dalam tampilan digital.
- Bagaimana merancang alat ukur tekanan dalam bentuk *portable pocket.*
- Bagaimana caranya menampilkan tekanan terukur ke LCD agar sesuai dengan tekanan yang terukur oleh *pressure gauge* sensor.

1.3 Batasan Masalah

Batasan permasalah yang digunakan dalam penelitian ini adalah sebagai berikut :

- Alat ukur ini adalah sebuah penelitian untuk menampilkan display dalam bentuk digital berbasis mikrokontroler dalam pengukuran tekanan.
- Pengambilan data dilakukan pada kompresor yang ada pada *Laboratorium Workshop* .
- Range pengambilan data 0 – 15 Psi
- Tampilan tekanan dalam satuan Psi
- Kalibrasi dilakukan antara *display*, *pressure gauge* sensor, dan *regulator pressure* yang *untraceable*

1.4 Tujuan

Tujuan dari tugas akhir ini adalah melakukan penelitian dengan membuat alat ukur tekanan yang dikemas secara *portable pocket* dan didisplaykan dalam bentuk digital pada LCD berbasis mikrokontroler AT89S51 dengan menggunakan sensor yang dimodifikasi sendiri dari sensor yang murah yaitu dari *pressure gauge*..

1.5 Sistematika Laporan

Laporan tugas akhir ini kami susun terdiri dari

- Bab I Pendahuluan

  Bab ini meliputi penjelasan secara singkat mengenai latar belakang permasalahan, permasalahan yang ada saat ini, batasan masalah, tujuan serta sistematika laporan tugas akhir.

- Bab II Teori Penunjang

  Bab ini menguraikan dan membahas teori yang dipakai untuk menunjang tugas akhir ini, khusus teori tentang peralatan perangkat keras dan perangkat lunak tugas akhir ini.

- Bab III Perencanaan Dan Pembuatan Alat

  Bab ini menjelaskan mengenai langkah - langkah pembuatan perangkat keras (*Hardware*).

- Bab IV Pengujian Perangkat Keras

  Bab ini membahas hasil – hasil pengujian perangkat keras dan pengujian alat tugas akhir secara keseluruhan.

- Bab V Penutup

  Pada bab ini berisi kesimpulan beserta saran – saran untuk tugas akhir ini.

- Daftar Pustaka

  Bagian ini berisikan literatur-literatur yang mendukung pembuatan tugas akhir ini.

- Lampiran-lampiran

BAB II
DASAR TEORI

Pada bab sebelumnya telah dijelaskan tentang latar belakang dan permasalahan dalam tugas akhir ini. Sebagai acuan dalam melakukan penyelesaian masalah. Beberapa teori atau referensi secara singkat yang menjadi dasar dalam pembuatan dan perancangan alat ukur tekanan dengan berbasiskan mikrokontroler AT89S51.

2.1 Pressure Gauge

Banyak bentuk dan cara kerja alat pengukur tekanan yang dipakai dalam suatu proses. Secara garis besar metode pengukuran tekanan dapat dibagi dalam beberapa metode, yaitu *gravitation instrument, alastic devormation instrument, special type* (*strain gauge* dan *piezo electric crystal*).

Pada dasarnya pengukuran tekanan dibedakan menjadi dua metode yaitu pengukuran tekanan absolute dan tekanan gauge, pengukuran tekanan absolute dilakukan dengan menganggap tekanan udara sekitar juga berpengaruh yang berarti hasil pengukuran absolute sedangkan pengukuran tekanan gauge dilakukan dengan mengabaikan tekanan udara luar atau tekanan udar luar dianggap sama dengan nol pada pengukuran ini bourdon *tube, bellow, spiral* dapat digunakan untuk menera besar perubahan tekanan.

Gauge pressure adalah tekanan yang terbaca pada alat ukur. Sedangkan *absolute pressure* adalah tekanan yang diukur mulai dari titik nol absolute, jika tekanan ini melebihi tekanan atmosfir local maka tekanan ini dapat dinyatakan sebagai jumlah dari tekanan atmosfir local dengan tekanan gauge. Perbedaan antara *gauge pressure scala* dengan *absolute pressure* adalah terletak pada lokasi nolnya. Pada *gauge pressure scala* titik nol adalah pada tekanan atmosfir, sedang pada *absolute pressure*

MILIK PERPUSTAKAAN ITS

scala titik nol terletak pada *absolute zero pressuren point. Pressure gage* terdiri dari dua macam, yaitu mekanik pressure gages dan elektrik pressure gage. Perbedaan dari kedua alat pengukur tekanan tersebut pada dasarnya terletak pada sensor yang mengubungkan antara tekanan dengan jarum penunjuk sekala yaitu menggunakan mekanik dan elektrik.

Metode pengukuran tekanan dapat dibagi sebagai berikut :

1. *Gravitation instrument*

   Adalah metode pengukuran tekanan dengan jalan mengimbangi tekanan yang diketahui dengaan yang tidak diketahui, misalkan

   - *Liquid coloumn gauge*
   - *U tube manometer*
   - *WIU tube manometer*
   - *Inclined manometer*
   - *Piston gauge*

2. *Elastik deformation instrument*

   Adalah pengukuran tekanan dengan membuat keseimbangan antara tekanan yang tidak diketahui dengan *kalibrated spring*, misalkan :

   - *Bourdon tube element*
   - *Metalik diapraghma element*
   - *Capsule element*
   - *Bellows element*

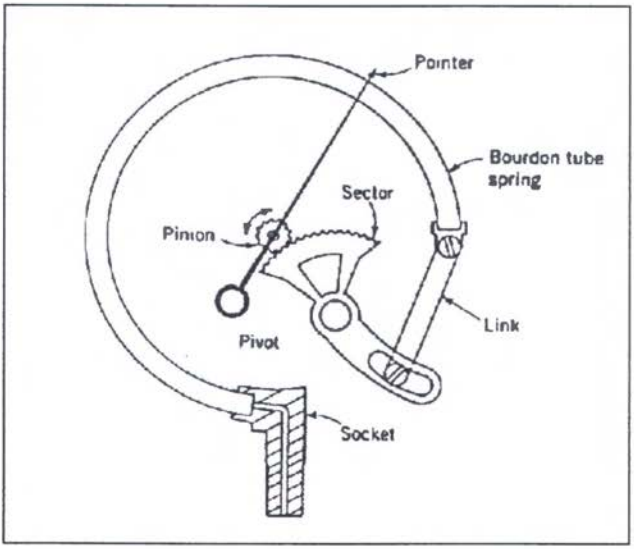
3. Metode elektrik

   Adalah pengukuran tekanan dengan metode yang berbeda dengan menggunakan :

   - *Strain gage*
   - *Piezoelektric crystal*

Tipe bourdon dan diaphagma gauge umumnya digunakan karena mempunyai gerakan ke luarnya secara mekanik, yang dapat dihubungkan secara langsung kepada tipe gerakan dari

pengontrolan *pneumatic control*, atau bias pula digabung dalam *pneumatic* atau *electronic force balance transmitter*

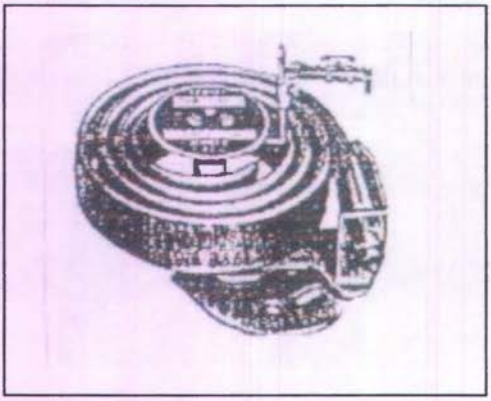


Gambar 2.1 Salah satu *Bourdon Tube "C"* [1]

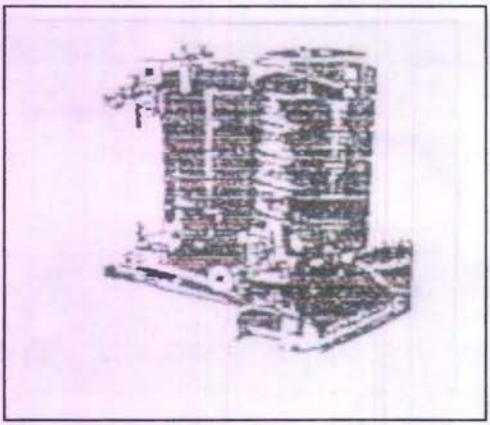
Bahan yang digunakan untuk tabung bourdon adalah *bronze, monel, alloy steel, beryllium copper, Ni-span C* dan *stainless steel*. Disamping *bourdon tube* instrument ini juga ada yang dilengkapi spiral dan atau *helix element* untuk membantu pergerakannya

Elastic pressure gage adalah tipe yang paling sering digunakan. Dalam *elastic pressure gage*, sebuah bodi elastis digunakan sebagai elemen sensor, dan tekanan tergantung dari jumlah perubahan bentuk dari elemen sensor. Range dari tekanan yang terukur tergantung dari *bourdon tube*, diafragma atau *bellow* yang digunakan sebagai elemen sensor.

Dibawah ini adalah gambar dari jenis-jenis dari *pressure gauge* yang bisa dipakai dalam industri.



Gambar 2.2 Salah satu *pressure gauge type helical* [1]



Gambar 2.3 Salah satu *pressure gauge type spiral* [1]

Tipe *pressure gauge* seperti gambar di bawah ini menggunakan *bellows* yang terbuat dari kuningan, *phosphor bronze* atau dari *stainless steel*. *Bellows pressure gage* biasanya

digabungkan dengan sebuah pegas. Bentuk alarm contact adalah suatu bentuk yang dihubungkan ke suatu alat peringatan bahaya, dengan menstel atau menset pada tekanan tertentu, sehingga apabila melebihi dari yang dipasang maka akan memberikan tanda bahaya. Alat ini dipakai untuk mengontrol mesin-mesin dan peralatan pada suatu pabrik atau kilang.

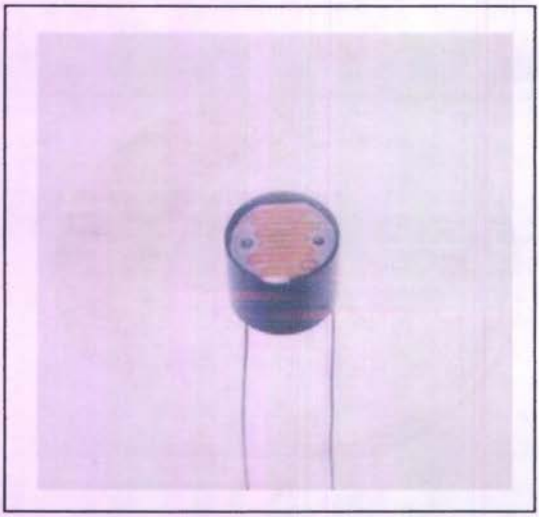
Gambar 2.4 *Pressure gauge* [2]

Tipe ini dapat dibagi dua sistem kerjanya, yang satu berdasarkan elektrik atau pneumatik dan pemasangnannya juga dapat melalui kontak langsung dengan jalur atau peralatan yang diukur atau tidak kontak secara langsung.

2.2 LDR

Sensor adalah salah satu bagian dari suatui sistem yang berfungsi sebagai penerima sinyal dan sinyal inilah yang akan digunakan untuk monitoring atau bisa juga untuk poengendalian. Dan sensor mampu merubah dari besaran fisis menjadi besaran elektronik agar dapat diolah oleh rangkaian transduser yang

nantinya data tersebut cukup kuat untuk ditransmisikan, karena dalam pensttransmisian data akan mengalami jika data tersebut dalam bentuk sinyal yang lemah.



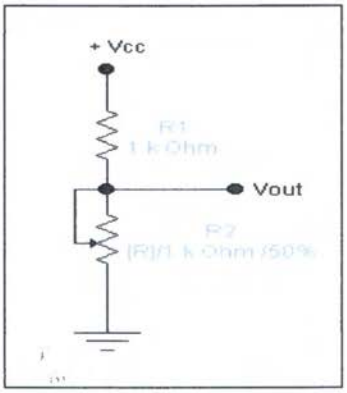
Gambar 2.5 LDR (*Light Dependent Resistor*) [2]

LDR (*Light Dependent Resistor*) merupakan sensor cahaya. LDR yang terbuat dari *Cadmium Sulfida*, yaitu bahan semikonduktor yang resistansinya berubah menurut banyaknya cahaya yang jatuh padanya. Daerah peka suatu sel foto konduktif terjadi dari lapisan *Cadmium Sulfida* yang diendapkan pada lapisan bawah isolasi rangkaian dengan kabel-kabel penyambungan diplotkan dalam epoksin respon jernih.

Cahaya yang jatuh di atas warna coklat berliku-liku bentuk pada sensor, menyebabkan perlawanan yang biasa dikenal dengan koefisien negatif. Prinsip kerja dari LDR adalah semakin banyak cahaya yang jatuh di sensor maka semakin kecil nilai hambatannya dan sebaliknya.

2.3 Rangkaian Pembagi

Rangkaian pembagi tegangan dapat dilihat pada gambar 2.6 yang digunakan untuk pengkonversian yang diperoleh dari variasi hambatan dan variasi tegangan.



Gambar 2.6 Rangkaian Pembagi [3]

Tegangan keluaran dari rangkaian pembagi tegangan adalah:

$$V_{out} = \frac{R1}{R1 + R2} Vin \quad \text{.....(2.1)}$$

2.4 Rangkaian ADC

Rangkaian ADC (*Analog to Digital Current)* adalah rangkaian yang digunakan untuk mengkonversikan dari tegangan analog ke digital. Sedangkan rangkaian oscilometrik adalah rangkaian akan membaca sinyal yang dikeluarkan oleh sensor.

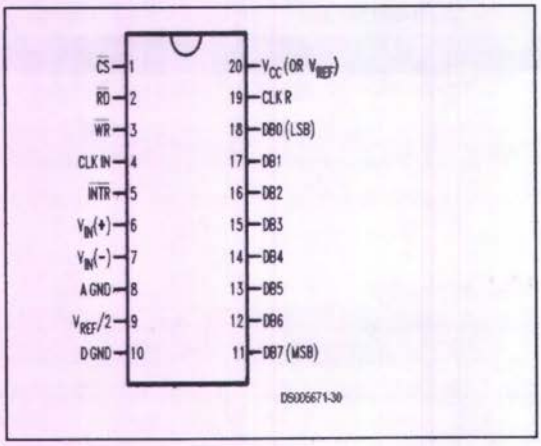
Analog to Digital Converter (ADC) digunakan untuk mengubah tegangan analog menjadi tegangan digital, yang bertujuan agar tegangan analog dapat diolah oleh suatu pengolah data yang berbasis rangkaian TTL (*Transistor – transistor Logic*), misalnya komputer. Jenis ADC yang umum digunakan

dalam ilmu elektronika terbagi menjadi dua golongan, yakni golongan *open loop* dan golongan *close loop*.

Yang termasuk ke dalam golongan *open loop* adalah tipe *Flash* ADC, *Slope Converter, Dual Slope Converter*. Sedangkan yang termasuk ke dalam golongan *close loop* adalah *single counter* ADC, *Tracking* ADC dan *Successive Approximation* ADC.

Mengingat banyaknya tipe ADC, maka dalam pembahasan ini hanya akan diterangkan mengenai ADC tipe *Successive Aproximation* ADC, karena dalam perencanaan alat ini akan digunakan ADC jenis ini.

ADC jenis *Successive Aproximation* merupakan golongan ADC *closed loop*. ADC jenis ini dapat dikatakan merupakan perpaduan yang sangat baik antara faktor kecepatan dan pada tingkat ketelitian atau kesusahan rangkaian. Waktu melakukan konversi pada ADC ini selalu tetap, tidak bergantung pada besarnya sinyal analog sebagai masukannya.



Gambar 2.7 Pin ADC 0804 [2]

Untuk nilai n bit ADC jenis *Successive Aproximation* memerlukan waktu konversi n + 1 siklus. Tetapnya waktu

konversi ini juga merupakan salah satu keuntungan penggunaan ADC jenis *Succesive Aproximation*. ADC disini digunakan untuk mengolah sinyal masukan dari sensor ke dalam mikrokontroler agar terbaca.

2.5 Mikokontroler

Suatu sistem dapat dikatakan sebagai mikrokomputer jika didalamnya telah tersedia tiga bagian utama yang mendukung sebuah komputer, yaitu RAM (*Random Access Memory*), EEPROM (*Electrical Erasable Programmable Read Only Memory*), atau EPROM (*Erasable Programmable Read Only Memory*), atau PROM (*Programmable Read Only Memory*), unit *input/output*, antarmuka serial dan parallel, *timer* dan *counter*, serta interup kontroler.

Mikrokontroler merupakan suatu elektronik yang termasuk *general purpose device* yang cocok untuk banyak aplikasi. Mikrokontroler adalah suatu komputer yang dibangun pada satu chip. Input dan outputnya serta sistem memori dibangun di dalam mikrokontroler, maka memungkinkan alat ini untuk di-*interface*-kan dengan *hardware* dan fungsi kontrol.

2.5.1 Mikrokontroler AT89S51

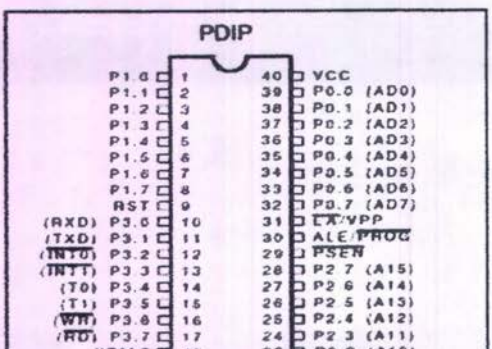
Mikrokontroler AT89S51 adalah keluarga dari MCS-51 dari Intel Corp, yang paling populer untuk mikroprosesor kelas 8 bit. Alat ini dikemas dalam bentuk QUAD dan DIP. Mempunyai 40 pin yang terdiri dari 4 buah *port* 8-bit yang sangat fleksibel penggunaannya. Pada Gambar 2.9 diperlihatkan konfigurasi pin-pin dari mikrokontroler AT89S51.

Fungsi pin pada mikrokontroler AT89S51 dapat dikelompokkan menjadi pin sumber tegangan, pin kristal, pin kontrol, pin input/output, dan pin interupsi.

Pada mikrokontroller AT89S51 ini dilengkapi dengan port untuk data LCD sebelum diolah dengan menggunakan *software assembler.*

Spesifikasi teknik dari mikrokontoller AT89S51 adalah sebagai berikut :

- 8-bit *Central Prosessing Unit*
- 4096 (4K) *bytes of In-System Reprogrammable Flash Program Memory*
- 128 bytes memori data internal
- 32 jalur masukan dan data keluaran yang bersifat dua arah yang dikelompokkan menjadi 4 bit port.
- Mode yang lebih bervariasi (*multiple mode*), dan kemampuan pemrograman dengan kecepatan tinggi untuk port serial.
- *Full Duplex serial port.*
  Dua buah 16 bit pewaktu atau *counter*



Gambar 2.8 Pin AT89S51 [2]

Kemampuan pengalamatan ke memori program dan memori data eksternal masing-masing mencapai 64 *Kbytes.*

Pada Tegangan suply, dihubungkan dengan tegangan 5 Volt.*Ground*, dihubungkan ke Vss atau *ground*.

➢ **Port I/O**

Port I/O pada mikrokontroler AT89S51 ada empat buah, dimana setiap port terdiri dari delapan bit. Masing-masing bisa diakses sendiri-sendiri maupun diakses secara bersama sama. Pada *port* 3 ada penambahan fungsi khusus yaitu untuk interup, timer, serial dan /RD serta /WR untuk akses memori eksternal.

Bila digunakan ROM (*Read Only Memory*) dan atau RAM (*Random Access Memory*) eksternal maka *port* 0 dan *port* 2 menjadi *address bus*. Selain itu *port* 0 juga bergantian menjadi data *bus*. Yang bertugas mengaturnya adalah ALE dan /PSEN yang bekerja sama dengan pin /RD dan /RW.

• **Port 0**

Port 0 merupakan *port* paralel 8 bit *open drain* dua arah. Bila digunakan untuk mengakses memori luar, *port* ini akan memultipleks alamat memori dengan data. Port 0 juga memerlukan kode *byte* pada saat pemrograman *flash*, dan kode *byte output* selama pembuktian program. *Eksternal pull up* dibutuhkan selama pembuktian program.

• **Port 1**

Port 1 adalah sebuah *port input output* 8 bit *bidirectional* dengan *pull-up internal*. *Output buffer port* 1 dapat menjadi sumber dari 4 TTL *input*. Ketika '1' ditulis ke pin-pin *port* 1, mereka di *pull high* oleh *pull-up internal* dan dapat digunakan sebagai *input*. Sebagai *input*, pin *port* 1 dipull rendah secara eksternal. *Port* 1 juga menerima perintah alamat *byte* rendah selama pemograman *flash* dan pembuktian.

• **Port 2**

Port 2 adalah sebuah *port input output* 8 bit *directional* dengan internal *pull up*. *Buffer output port* 2 dapat berfungsi sebagai sumber 4 input TTL. Ketika '1' dituliskan ke pin *port* 2 mereka di *pull high* oleh internal *pull up* dan dapat

digunakan sebagai input. *Port* 2 mengeluarkan perintah alamat *byte* selama pengambilan dari program memori eksternal dan selama pemasukan ke data memori eksternal yang menggunakan pengalamatan 16 bit (MOVX @DPTR). Dalam aplikasi ini menggunakan internal *pull up* yang kuat ketika menggeluarkan '1'. *Port* 2 juga menerima perintah alamat bit tinggi dan beberapa sinyal kontrol selama pemrograman *Flash* dan pembuktian. Pada *port* 1 dan juga *port* 0 dan *port* 2 (jika tidak dipakai memori eksternal) pengaksesan bisa tiap pin sendiri-sendiri atau dalam satu kumpulan *port* 8 bit.

- **Port 3**

Port 3 adalah sebuah port input output 8 bit *directional* dengan internal *pull up*. *Buffer output port* 3 dapat berfungsi sebagai sumber 4 input TTL. Ketika '1' dituliskan ke pin *port* 3 mereka di *pull high* oleh internal *pull up* dan dapat digunakan sebagai input. Pin-pin Port 3 yang secara eksternal di *pull low* seperti sumber arus disebabkan oleh *pull up*. *Port* 3 juga menerima beberapa sinyal kontrol untuk pemrograman *Flash* dan pembuktian. *Port* 3 juga melayani berbagai macam ciri fungsi-fungsi khusus dari AT89C51 sebagai berikut :

Tabel 2.1 Fungsi –Fungsi Khusus Pada AT 89S51

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

- **Interrupt**

  Terdiri dari sepasang *interrupt* eksternal yaitu INT0 dan INT1. Untuk mengaktifkan kedua *interrupt* ini maka pada register IE (alamat 0A8H) diberi angka aktif 1 pada bit pertama dan ketiga. Prioritas dari *interrupt* ini diatur oleh register IP (alamat 0B8H), juga pada bit pertama dan ketiga. Untuk mematikan kerja interrupt sementara maka register EA diberi nilai 0 atau dengan program clr EA.

- **Timer**

  Selain *interrupt eksternal port control* ini terdapat sepasang *interrupt timer* T0 dan T1. Untuk mengaktifkan kedua *timer* harus melalui beberapa prosedur dahulu. Pertama adalah penentuan jenis *timer* dengan register TMOD. Register ini tidak bisa diakses bit per bit. biasanya *setting* dilakukan pada awal inisialisasi program. Setelah *setting* TMOD maka *timer* siap untuk dijalankan *(start)*, dihentikan atau lain-lainnya dengan mengakses register timer khusus yaitu SFR (*Special Function Registers*).

  Timer ada 3 mode yaitu mode 0 (13 bit *timer*), dimana 5 bit dihandle oleh register TLx dan 8 bit dihandle oleh register THx serta *overflow* dihandle oleh TFx. Yang kedua adalah mode 1 yaitu 16 bit *Timer*, dimana TLx dan THx sama-sama 8 bit. Pada mode 2 adalah 8-bit *Auto-Reload Timer*, artinya TLx akan menghitung dari 00H sampai FFH kemudian saat kembali ke 00H maka nilai THx akan diletakkan ke TLx. Hal itu dilakukan berulang-ulang (*Auto Reload*) sampai TFx di set *overflow* yang mana tergantung inisialisasi TMOD dan THx.

  Pada *Timer* yang keempat yaitu *Split Timer*, yaitu jenis *timer* yang mempunyai dua buah 8 bit *timer* tetapi keduanya tidak saling mempengaruhi baik waktu aktif maupun kondisi *start* maupun *stop timer* masing-masing.

  Sumber pewaktu dari *timer* bisa dari dalam (internal) maupun dari luar (eksternal). Untuk kasus sumber *timer* eksternal (disebut juga kondisi *counter*) maka yang mengaktifkan adalah bit C/D pada TMOD jika bit 1 maka

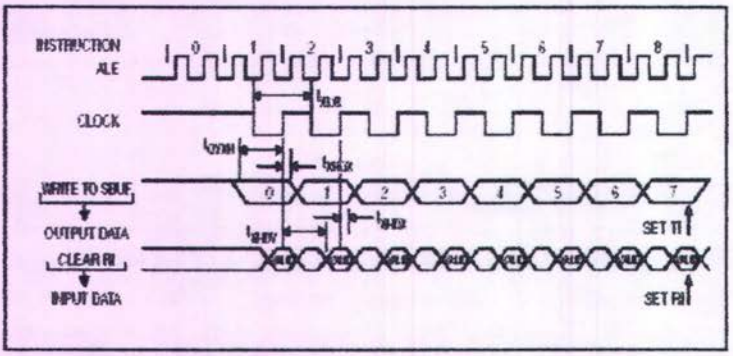
sumber internal sebagai *interval timing* dan eksternal sebagai *event counting* atau menghitung sinyal yang datang dari luar. Jika bit = 0, maka ibersifat kontinyu. Frekuensi *timer* eksternal maksimal adalah 500 kHz dengan *clock osilator* 12 MHz. Secara *software timer* diaktifkan dengan perintah SETB TR0 dan dihentikan dengan perintah CLR TR0.

➢ **Serial**

Pada *Port Control* (3) ada sepasang pin serial, dimana digunakan untuk komunikasi data serial. Mode komunikasi serial ini ada empat macam yang kesemuanya diatur pada register SCON (*Serial Control Register*) pada alamat 099H.

Mode pertama yaitu Mode 0, dimana aktif bila bit SM0 dan SM1 bernilai 0. Hal ini mengakibatkan serial menjadi shift register mode dimana pin TxD sebagai *clock* dan RxD sebagai data input maupun output 8 bit yang dilewatkan satu persatu dengan LSB yang pertama dikirim.

Mode kedua yaitu mode 1, 8-bit UART dengan *Baud Rate* bisa diatur, artinya kecepatan pengiriman data per detik bisa diubah. Hal itu disebabkan pengiriman data dilewatkan TxD dan penerimaan data lewat RxD, sedangkan clock diambil dari *timer* internal, yaitu melihat TFx dari *timer*



Gambar 2.9 *Timing Diagram Mode 0 Serial*[4]

Semuanya itu mudah dilakukan sebab lebar data tetap yaitu 10 bit, bit pertama yang dikirim disebut *start* bit dengan kondisi harus selalu 0, kemudian bit kedua sampai dengan bit ke sembilan adalah data yang sebenarnya dimulai dari LSB ke MSB (D0 dulu dikirim sampai dengan D7) dan terakhir adalah bit *stop* yang harus berkondisi *high*. Begitu *stop* bit ini sudah muncul di TxD untuk dikirim maka *flag* register T1 akan diset secepat mungkin sehingga bisa mengirim 8 bit data serial selanjutnya.

Pada sisi penerima (RxD), setelah menerima perubahan dari 1 ke 0, maka *counter* pembagi 16 mulai menghitung dari awal, RxD mengambil data 0 dulu sebanyak delapan kali, bila tidak sampai delapan kali pengambilan terjadi perubahan dari 0 ke 1 (data tidak 0) maka dianggap bahwa penerima telah ditrigger oleh *noise* dan bukan dimasukkan sebagai input yang *valid*.

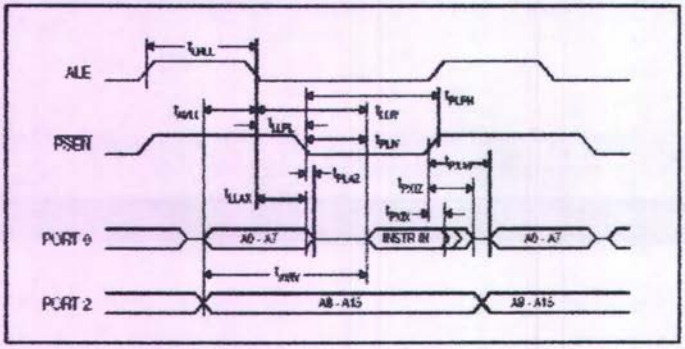
Mode ketiga yaitu mode 2, 9-bit UART dengan *Baud Rate* tetap, yaitu pengiriman dengan baud rate 1/32 atau 1/64 dari osilator. Jumlah bit yang dikirim adalah 11 bit, yaitu bit *start* yang pertama, kemudian delapan bit data yang sebenarnya, setelah itu ada bit ke sembilan yang dapat diprogram apakah 1 atau 0 tergantung kita, dan yang terakhir adalah bit *stop*. Adanya bit yang kesembilan dimana dapat kita atur apakah 1 atau 0 hampir sama dengan cara kerja *parity* bit. Mode keempat yaitu mode 3, 9-bit UART dengan *Baud Rate* bisa diatur. Mode ini sama dengan mode 2 hanya bedanya pada *Baud Rate* yang bisa diatur.

- **RD dan /WR**

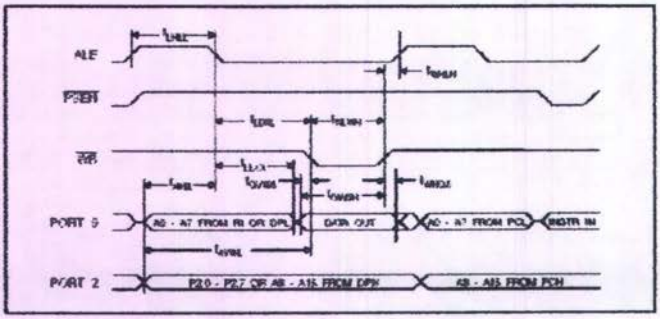
Sepasang pin yang terakhir adalah untuk pembacaaan dan penulisan memori eksternal yaitu /RD dan /WR. Kedua pin ini aktif low. Kerja pin-pin ini bersama-sama dengan pin control ALE dan /PSEN. Sistem pengalamatan memori dari AT89S51 menggunakan *multiplex addressing* (pengalamatan bergantian). Pin yang digunakan untuk mengakses memori program external ialah pin PSEN'. Pin PSEN' dihubungkan

dengan pin pada memori eksternal yang berfungsi sebagai *Output Enable* (OE'), sedangkan Pin RD' dihubungkan pada pin memori eksternal yang berfungsi sebagai *Output enable* (OE').

Jika pengambilan data dilakukan menggunakan sinyal *Read* maka, pengambilan kode akan dilakukan dengan pengambilan sinyal PSEN' (*Program store enable*). Sinyal PSEN' aktif pada saat mikrokontroler mengeksekusi program di memori eksternal, yaitu pada saat pin no 31 ($\overline{EA}$) berlogika 0 atau terdapat lompatan proram kememori eksternal.

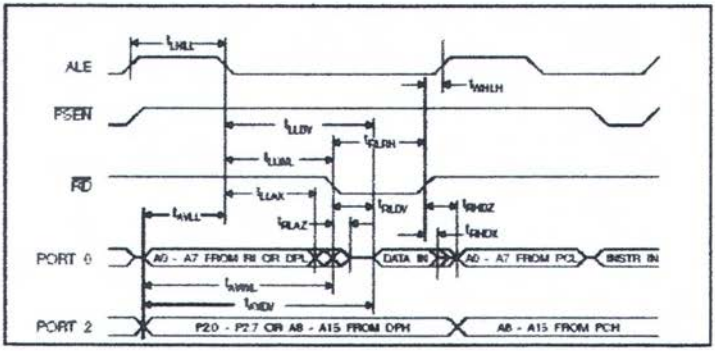


Gambar 2.10 Membaca ROM Eksternal[4]



Gambar 2.11 Menulis ke RAM [4]

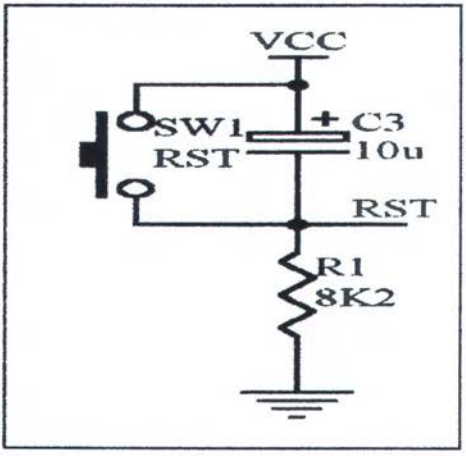
Pin yang digunakan untuk mengakses memori data eksternal ialah WR' dan RD'. Pin WR' dihubungkan pada pin memori eksternal yang berfungsi sebagai *write enable* (W'), sedangkan pin RD' dihubungkan dengan pin pada memori eksternal yang berfungsi sebagai *output enable* (OE').



Gambar 2.12 Membaca dari RAM [4]

➢ **Reset**

Pin RST ini diberi high bila ingin mengembalikan semua kondisi dalam keadaan awal *ON*.



Gambar 2.13 Rangkaian *Power ON Reset* [4]

Lama pemberian *high* minimal 2 *cycle*. Rangkaiannya dapat dilihat pada, Gambar 2.13.

- **ALE/PROG**

  Pulsa output *Address Latch Enable* untuk memberikan alamat *byte* rendah selama mengakses memori eksternal. Pin ini juga merupakan input *pulse* program (PROG) selama pemrograman *Flash*.

  Pada operasi normal ALE dipancarkan pada 1/6 frekuensi *oscillator* secara konstan, dan bisa digunakan untuk tujuan *external timing* atau *clock*. Catatan, bagaimanapun bahwa satu pulsa ALE diloncati antara tiap akses ke memori data eksternal.

  Jika diinginkan, operasi ALE dapat *disabled* dengan mengeset bit 0 lokasi 8EH dari SFR. Dengan set bit, ALE hanya aktif selama ada instruksi MOVX or MOVC. Sebaliknya, pin dengan lemah ditarik jadi *high*. Mengeset *disable* bit-ALE tidak mempunyai pengaruh jika mikrokontroler di dalam mode eksekusi eksternal.

- **PSEN**

  Program Store Enable adalah membaca *strobe* ke memori program eksternal. Saat AT89S51 mengeksekusi kode dari memori program eksternal, PSEN teraktifkan dua kali setiap putaran mesin *(machine cycle)*, kecuali bila pengaktifan dua PSEN diloncati selama tiap akses ke memori data eksternal.

- **EA/Vpp**

  External Access Enable. EA harus terjalin ke GND agar komponen *enable*, untuk fetch kode mulai dari 0000H sampai FFFFH di lokasi memori program eksternal. Catatan, bagaimanapun bahwa jika *lock* bit 1 terprogram , EA akan ter-*latch* dibagian dalam pada *reset*.

  EA seharusnya terjalin ke Vcc untuk eksekusi-eksekusi program internal.

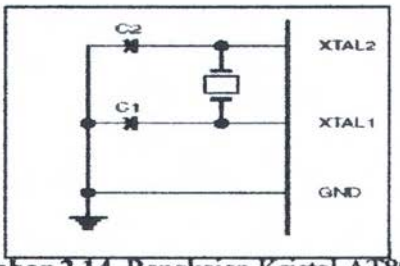
Pin ini juga menerima 12 volt *programming enable voltage* (Vpp) selama pemrograman *flash*, untuk bagian-bagian yang membutuhkan Vpp 12 volt.

➢ **Pin Oscillator**

Sepasang pin ini diberi kristal eksternal yang di *bypass* dengan dua buah kapasitor kira-kira 30 pF ± 10%. Kristal yang dipakai umumnya 12 MHz, tetapi apabila ada komunikasi serial lebih mudah menggunakan kristal dengan ukuran 11,059 MHz dalam penentuan *Baud Rate*-nya. XTAL1 adalah Input ke pembalikan penguat *oscillator (inverting oscillator amplifier)* dan input ke clock internal pengoperasian rangkaian. Sedangkan XTAL2 adalah Output dari pembalikan penguat *oscillator*.

- **Oscilator**

XTAL1 dan XTAL2 adalah input dan output, berturut-turut dari *inverting amplifier* yang mana dapat dikonfigurasi sebagai misal pada *chip oscillator*, seperti terlihat pada gambar 1. Salah satu kristal kwarsa atau resonator keramik bisa digunakan. Untuk menjalankan alat dari sumber *clock* eksternal (*external clock source*), XTAL2 seharusnya tidak terhubung kiri saat XTAL1 dijalankan seperti terlihat pada gambar 2.13 di bawah ini. Di mana tidak membutuhkan *duty cycle* di sinyal *clock* eksternal, semenjak input ke internal *clocking circuitry* melalui satu dibagi dua *flip-flop*, tetapi spesifikasi waktu *low* dan *high* tegangan maksimum atau minimum harus terpantau.



Gambar 2.14 Rangkaian Kristal AT89S51 [4]

➢ **Special Function Registers (SFRs)**

Pada mikrokontroller AT89S51 juga memiliki beberapa *Spesial Function Register* (SFRs) sebagai berikut :

Tabel 2.2 Tabel *Special Functions Registers* [4]

Simbol	Nama	Alamat
* ACC	Akumulator	0E0H
* B	Register B	0F0H
* PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer (DPH & DPL)	83H & 82H
* P0	Port 0	80H
* P1	Port 1	90H
* P2	Port 2	0A0H
* P3	Port 3	0B0H
Simbol	Nama	Alamat
* IP	Interrupt Priority Control	0B8H
* IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
Simbol	Nama	Alamat
* TCON	Timer/Counter 2 Control	88H
TH0	Timer/Counter 0 (high byte)	8CH
TL0	Timer/Counter 0 (low byte)	8AH
TH1	Timer/Counter 1 (high byte)	8DH
TL1	Timer/Counter 1 (low byte)	8BH
* SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

Catatan : SFRs (*Spesial Function Register*) yang ditandai dengan tanda * adalah SFRs yang baik secara bit maupun *byte*-nya dapat dialamati.

Fungsi masing-masing SFR dapat dijelaskan sebagai berikut :

- **Akumulator (ACC)**

  Register akumulator merupakan register 8 bit yang khusus, dimana instruksi aritmetika dan logika berlangsung di register ini. Akumulator dapat digunakan sebagai sumber (*source*) maupun tujuan (*destinition*) dalam suatu operasi aritmetika maupun logika. Instruksi-instruksi yang berhubungan dengan memori eksternal memerlukan akumulator sebagai sumber maupun tujuannya.

- **Register B**

  Register B digunakan bersama-sama dengan akumulator pada saat operasi perkalian dan pembagian. Jika suatu operasi nilainya melebihi 1 *byte*, maka MSB hasil operasi tersebut akan disimpan di register B sementara bagian LSB-nya di akumulator. Untuk instruksi lainnya, register B juga dapat diperlakukan sebagai register biasa yang digunakan untuk menyimpan data.

- **Program Status Word**

  Didalam register PSW tersimpan informasi mengenai status program.

- **Stack Pointer (SP)**

  Register SP mempunyai lebar 8 bit dan berfungsi untuk menunjukkan *byte* alamat sebelumnya yang didorong (*Push*) masuk ke dalam tumpukkan (*Stack*). Stack pointer secara otomatis akan ditambah satu (*increment*) pada saat instruksi *Push/Call* dieksekusi. Walaupun *stack* bisa disimpan dimana saja di dalam RAM namun *stack pointer* selalu diinisialisasi pada alamat 07H setelah *reset* (harga *default*). Hal ini menyebabkan *stack* dimulai pada alamat 08H dalam RAM

internal. Oleh karena itu jika menginginkan *stack* dimulai pada alamat yang telah kita tentukan maka harus dilakukan proses peng-inisialisasian.

- **Data Pointer**

  Data pointer (DPTR) terdiri dari *byte* atas (DPH) dan *byte* bawah (DPL). Fungsinya adalah untuk menyimpan alamat sepanjang 16-bit. Dapat dimanipulasikan sebagai satu register 16-bit atau sebagai dua register 8-bit yang masing-masing berdiri sendiri.

- **Port 0 – port 3 (P0 – P3)**

  P0, P1, P2 dan P3 adalah SFRs *latches* dari port 0,1,2 dan 3. Masing-masing terdiri dari 8-bit saluran.

➢ **Serial Data Buffer (SBUF)**

Serial data buffer (SBUF) merupakan register yang berhubungan dengan operasi data serial. Register ini digunakan sebagai *buffer* pada saat data serial akan dikirim atau diterima. Register ini dioperasikan bersama-sama dengan akumulator. Pada saat data dari akumulator dipindahkan ke SBUF maka operasi yang dilakukan adalah pengiriman data, demikian pula sebaliknya jika data dari SBUF dipindahkan ke akumulator maka operasi yang dilakukan adalah penerimaan data.

➢ **Control Register (IP, IE, TMOD, TCON, SCON, PCON**

Special function Register IP, IE, TMOD, TCON, SCON dan PCON mengandung bit control dan status untuk mengatur sistem *interupt, timer/counter, power control* dan *serial port.*

25.2 Memori AT 89S51

Pada mikrokontroler AT 89S51 mempunyai daerah memori yang dipisahkan antara program memori, eksternal data memori dan internal data memori.

➢ **Program Memori**

Program memori hanya dapat dibaca dan tidak dapat ditulis. Program mempunyai kapasitas sebesar 64 *Kbyte*. Mikrokontoler AT89S51 mempunyai internal program memori sebesar 4096 *byte* (4 *Kbyte*) yang berbentuk *flash program memory*. Penggunaan internal program memori diatur oleh pin /EA, jika pin /EA=VCC maka program pada alamat 0000H – 0FFFH terletak pada internal memori sedangkan alamat 1000H – FFFFH terletak pada eksternal memori. Jika pin /EA=GND maka semua program dieksekusi oleh eksternal memori.

➢ **Data Memori**

Data memori pada mikrokontroler AT89S51 memiliki internal data memori sebesar 128 *byte* dan juga menggunakan eksternal data memori berkapasitas sampai 64 *Kbyte*.

➢ **Eksternal Memori**

Eksternal memori pada mikrokontroler AT89S51 terdiri dari eksternal program memori dan eksternal data memori. Untuk mengakses eksternal program memori kita memerlukan sinyal /PSEN (*Program Strobe Enable*) yang berfungsi sebagai *read strobe*. Sedangkan eksternal data memori diakses melalui sinyal /RD dan /WR yang secara berturut-turut berfungsi sebagai *read* dan *write strobe*.

Eksternal program memori diakses dalam dua kondisi, yaitu saat /EA aktif, atau saat *Program Counter* (PC) mengandung alamat terakhir dari internal program memori. Karena mikrokontroler AT89S51 memiliki internal program memori maka pin /EA dapat juga tidak diaktifkan sehingga mengakses internal program memori.

Akses ke eksternal program memori selalu menggunakan lebar alamat 16-bit. Sedangkan akses ke eksternal data memori bisa menggunakan lebar alamat 8-bit atau 16-bit, tergantung dari mode pengalamatan yang digunakan.

Karena program dan data memori diakses oleh *strobe* yang berbeda (eksternal program memori oleh /PSEN

MILIK PERPUSTAKAAN ITS

sedangkan eksternal data memori oleh /RD dan /WR) maka untuk aplikasi tertentu yang mengijinkan untuk mengeksekusi program dari eksternal data memori, daerah kerja (*space*) eksternal program dan data memori harus digabungkan. Dalam mikrokontroler AT89C51 hal ini bisa dilakukan dengan cara meng-AND-kan /PSEN dan /RD. Logika positif dari operasi AND kedua sinyal tersebut akan menghasilkan *active-low read strobe* yang dapat digunakan untuk menggabungkan daerah kerja kedua jenis eksternal memori tersebut.

2.5.3 Bahasa Assembly AT89S51

Mikrokontroler dan mikroprosesor menggunakan kode biner sebagai perintahnya. Namun pada saat membuat suatu program, sangat sulit bila menggunakan bilangan biner untuk mempresentasikan perintah mesin. Cara yang mudah yaitu dengan menggunakan bahasa assembly. Bila setiap perintah ditampilkan dalam bentuk ini program akan lebih mudah untuk dibaca dan dipahami. Setiap perintah mesin diberi nama yang sederhana dan singkat yang disebut mnemonik. Karena terdapat korespondensi satu-satu antara perintah bahasa assembly dasar dan perintah mesin, pengubahan dari suatu bentuk ke bentuk lainnya dapat dilakukan.

Pemetaan satu ke satu ini memiliki pengecualian untuk teknik assembly dengan bantuan komputer. Hampir semua bahasa assembly menyertakan *directive* (pengarah) yang menyediakan data penting bagi komputer. Pengarah ini muncul dalam program seperti mnemonik biasa namun sebenarnya hanya digunakan untuk menentukan segi tertentu dalam proses assembly.

Program bahasa assembly dapat diubah kedalam bahasa mesin. Dalam kedua bentuk (bahasa assembly dan bahasa mesin) perintah program secara umum disebut kode. Istilah kode sumber (*source code*) diperuntukan bagi program asli

yang ditulis dalam bahasa assembly. Hasil perintah mesinya disebut kode objek (*object code*).

Pemakaian paling sederhana dari bahasa assembly adalah sebagai bantuan dalam pengembangan kode untuk mikroprosesor secara manual. Namun setelah program assembly yang dapat dikerjakan dalam komputer berhasil dibuat, maka jarang sekali pemrograman ditulis langsung kode objeknya.

2.5.4 Mode Pengalamatan

Dalam keluarga mikrokontroler ATMEL khususnya AT89S51 bahasa assembly-nya mempunyai kesamaan dengan mikrokontroler dari keluarga MCS-51 dari INTEL. Bahasa assembly pada mikrokontroler AT89S51 terdapat beberapa mode pengalamatan yang berkaitan dengan instruksi yang digunakan.

- Mode Pengalamatan Langsung (*direct addressing*)
  Data yang ditunjukkan oleh alamat diberikan oleh operandnya. Mode ini hanya dapat digunakan pada internal RAM dan SFRs.
- Mode Pengalamatan Tidak Langsung (*indirect addressing*) Alamat tidak ditunjukkan langsung, disimpan dalam register. Mode ini dapat digunakan pada internal dan eksternal RAM. Alamat yang disimpan dapat berjumlah 8 atau 16 bit.
- Mode Pengalamatan Register (*register addressing*)
  Data yang akan dieksekusi disimpan dalam register yang bisa digunakan adalah R0 sampai R7.
- Mode Pengalamatan *Immediate Constant*
  Data (konstanta) yang akan dieksekusi diberikan langsung, tanpa disimpan terlebih dahulu.

2.5.5 Instruksi Bahasa Assembly AT89C51

Bahasa Assembly AT89S51 mempunyai empat group fungsi instruksi. Fungsi-fungsi instruksi tersebut antara lain :

- Data transfer
- Aritmetika
- Logika
- Kontrol transfer

Semua fungsi instruksi tersebut merupakan kumpulan dari kode-kode mesin untuk melakukan operasi-operasi umum yang melibatkan immediate data, register, internal RAM, internal ROM, eksternal data memori dan eksternal program memori.

➢ **Aritmetika**

Bahasa Assembly AT89S51 memiliki empat instruksi dasar aritmetika. Karena AT89C51 menggunakan prosesor 8-bit, maka semua operasi aritmetika yang dapat langsung dieksekusi memiliki nilai 8-bit termasuk tanda *(sign)*. Bit *overflow* yang terdapat pada register PSW akan memberikan tanda untuk operasi aritmetika yang menggunakan komplemen 2 dan tidak mempresentasikan operasi penjumlahan 8-bit. Sebuah instruksi koreksi ke nilai BCD *(Binary Code Decimal)* disediakan, yang akan langsung mengkoreksi data yang terdapat di akumulator.

Register PSW merupakan register yang sangat penting dalam operasi aritmetika ini. Bit-bit yang terdapat dalam register ini akan terpengaruh langsung pada setiap pelaksanaan eksekusi instruksi aritmetika. Instruksi-instruksi yang merupakan fungsi aritmetika adalah ; INC *(increment)*, ADD *(Adder)*, ADDC (*Adder With Carry*), DA (*Decimal-add-Adjust for BCD addition*), SUBB (*SUBtract with Borrow*), DEC (*DECrement*), MUL (*Multiplication*), DIV (*Division*).

➢ **Logika**

Operasi logika dibagi menjadi dua bagian instruksi dasar yaitu *single operand operation* dan *two operand operation.*

- ***Single Operand Operation***

Merupakan instruksi yang memiliki satu operand contohnya adalah ; CLR *(CLeaR)*, SETB *(SET Bit)*, CPL *(ComPLement)*, RL *(Rotate Left)*, RLC *(Rotate Left with Carry)*. RR *(Rotate Right)*, RRC *(Rotate Right with Carry)*, SWAP *(rotate left four)*.

- ***Two Operand Operation***

Merupakan instruksi-instruksi yang memiliki dua operand contohnya adalah : ANL (*And Logic*), ORL (*OR Logic*), XRL (*eXclusive oR Logic*).

➢ **Kontrol Transfer**

Terdapat tiga kelas operasi yang termasuk ke dalam operasi kontrol transfer yaitu ; *unconditional calls, returns* dan *jumps*; *conditional jumps*; dan *interrupts*. Semua operasi kontrol transfer diperlukan pada saat kondisi-kondisi tertentu, dimana jalannya program tidak sekuensial.

➢ **Unconditional calls, returns dan jumps**

Merupakan operasi yang akan melakukan transfer kontrol dari alamat asal program *counter* ke alamat target tanpa syarat loncatan. Dapat dilakukan baik secara langsung maupun tidak langsung contohnya adalah ; ACALL (*Absolute CALL*), LCALL *(Long CALL)*, RET (*RETurn from subroutine*), AJMP (*Absolute JuMP*), LJMP (*Long JuMP*), SJMP (*Short JuMP*). JMP @A+DPTR.

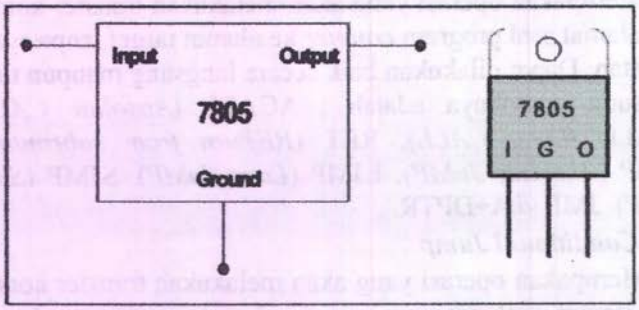
➢ ***Conditional Jump***

Merupakan operasi yang akan melakukan transfer kontrol dari alamat asal program *counter* ke alamat target dengan syarat loncatan dan dilakukan secara langsung. Pada operasi ini alamat boleh melebihi -128 byte ruang program memori untuk loncatan mundur dan +128 byte untuk loncatan maju. Instruksi tersebut adalah : JZ (*Jump if accumulator Zero*),

JNZ (*Jump if accumulator Not Zero*), JC (*Jump if Carry flag set*). JNC (*Jump if Not Carry flag set*), JB (*Jump if Bit set*), JNB (*Jump If Not BIT set*), JBC (*Jump if Bit is set and Clear bit*), CJNE (*Compare Jump Not Equal*), DJNZ (*Decrement Jump Not Zero*), RETI (*RETurn from Interrupt*).

2.6 Regulator

Pemakaian regulator pada pencatu daya berfungsi sebagai stabilitas tegangan. Komponen aktif ini mampu meregulasi tegangan menjadi stabil. Komponen ini sudah dikemas dalam sebuah IC regulator tegangan tetap yang biasanya sudah dilengkapi dengan pembatas arus *(current limiter)* dan juga pembatas suhu *(thermal shutdown)*. Jenis IC regulator tegangan tetap yang sering dipakai adalah jenis 78xx atau 79xx. IC regulator 78xx menghasilkan output tegangan dengan polaritas positif sedangkan 79xx menghasilkan output tegangan dengan polaritas negatif . Konstruksi dari regulator ini dapat dilihat pada Gambar 2.11. input dari 9 VDC yang berasal dari kapasitor sebagai *filter* dan output 5 VDC yang dihubungkan ke rangkaian sensor.



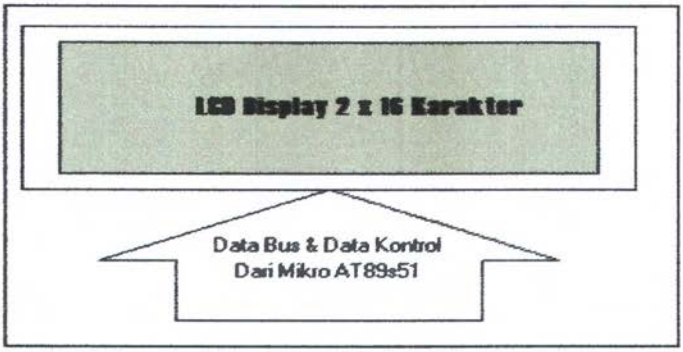
Gambar 2.15 IC Regulator 7805 [2]

Hanya saja perlu diketahui supaya rangkaian regulator dengan IC tersebut bisa bekerja, tegangan input harus lebih besar dari tegangan output regulatornya. Biasanya perbedaan tegangan

V_{in} terhadap V_{out} yang direkomendasikan ada di dalam datasheet kompenen tersebut. Pemakaian alumunium pendingin *(heatsink)* dianjurkan jika komponen ini dipakai untuk mencatu arus yang besar.

2.7 LCD 2 x 16

Untuk menampilkan hasil dari pengukuran kecepatan, inputan setpoint kecepatan yang kita inginkan kita membutuhkan tampilan (display), LCD (*Liquid Cristal Display*) merupakan sejenis *crystal* yang akan berpendar jika diberi tegangan tertentu, sehingga perpendaran tersebut dapat diatur untuk membentuk angka, huruf dan lain sebagainya. LCD yang digunakan dalam percobaan ini adalah menggunakan LCD dengan banyak baris dan karakter adalah 2 x 16.

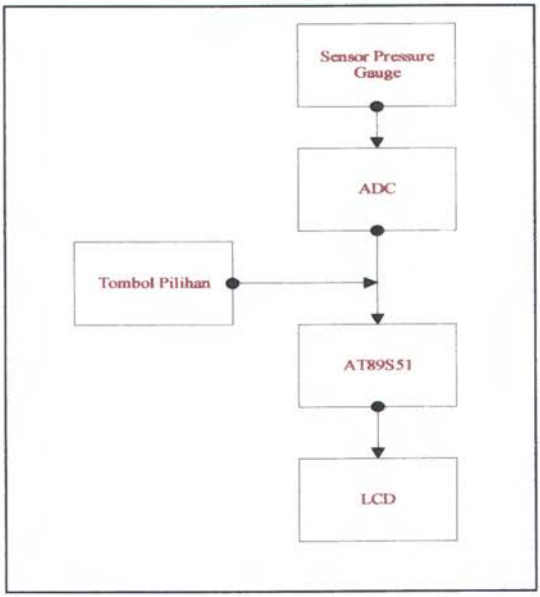


Gambar 2.16 LCD 2 x 16 [2]

BAB III
PERANCANGAN ALAT

Dalam bab ini akan dibahas perencanaan perangkat keras dan perangkat lunak guna mewujudkan tugas akhir ini, yang berjudul "Rancang Bangun Alat Ukur Tekanan "Portable" berbasiskan Mikrokontroler ".

Alat ini direncanakan agar dapat mengukur tekanan yang diproses dengan menggunakan Mikrokontroler AT89S51. Data pengukuran akan diproses dengan menggunakan mikrokontroler dan ditampilkan pada papan *display* yang berupa LCD. Perencanaan alat ini bertujuan untuk menciptakan alat ukur tekanan yan dikemas secara *portable pocket* dan dalam tampilan digital. Diagram blok sistem alat ukur tekanan dapat digambarkan sebagai berikut.



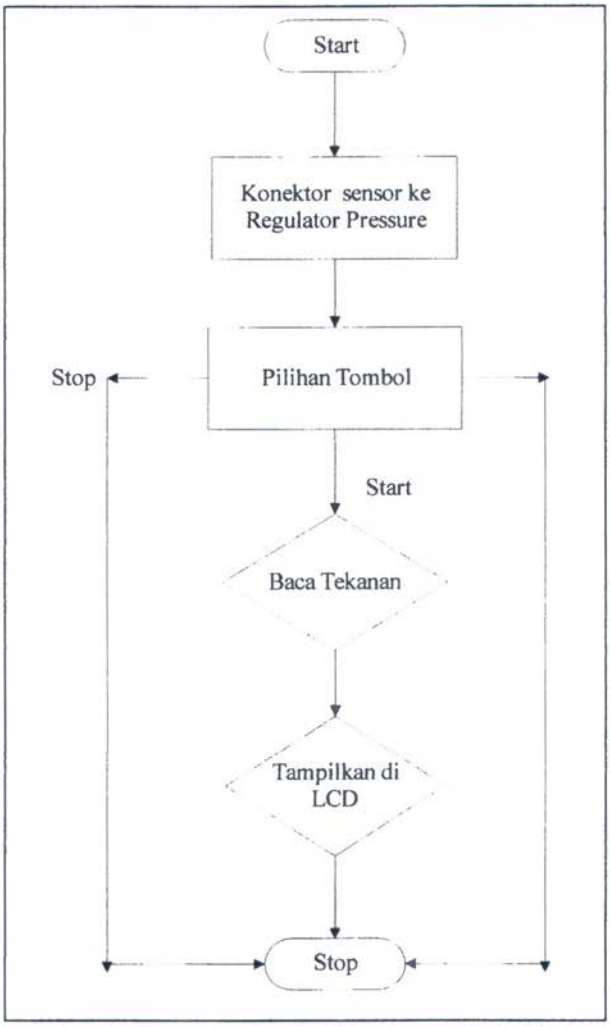
Gambar 3.1 Diagram blok sistem

3.1 Mekanisme Kerja Sistem

Mekanisme kerja sistem pada alat ukur tekanan berbasis Mikrokontroler AT89S51, adalah didasarkan pada pengukuran sensor *pressure gauge*. Dalam hal ini *pressure gauge* telah dimodifikasi dengan LDR dan led yang dihubungkan dengan rangkaian pembagi. Keluaran dari sensor berupa tegangan yang masuk ke ADC. Setelah masuk ke ADC maka akan disalurkan ke mikrokontroler. Setelah data dari ADC dimasukkan ke Mikrokontroler maka hasil pengukuran tersebut akan didisplaykan ke LCD berupa pengukuran digital. Sistem pembuatan progresif kerja digambarkan desain algoritma dibawah ini:

- Instalasi sensor, komponen *hardware*, memulai adanya akusisi data.
- Kemudian sensor dikoneksikan ke kompresor sebagai sumber tekanan.
- Sensor pressure gauge menerima tekanan dari kompresor.
- *Signal Conditioning* (rangkaian pembagi) mengkondisikan pengukuran sinyal untuk pengukuran minimum berupa data 0 V dan pengukuran level maksimun berupa 5 V
- ADC mengkonversi data sinyal analog pengukuran 0 – 5 V menjadi data sinyal digital 0 – 255 bit
- Mikrokontroller membaca sinyal ADC 0-255 untuk dimasukkan ke LCD.
- LCD menampilkan hasil pengukuran

Untuk memperjelas mekanisme kerja dari alat ukur ini maka dapat dilihat flowchart di bawah ini.

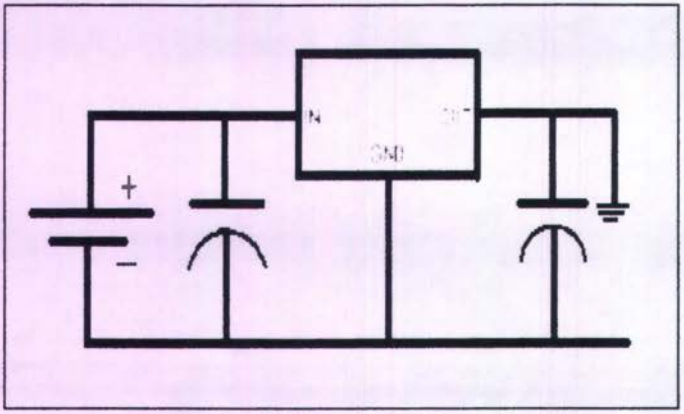


Gambar 3.2 Flowchart mekanisme kerja sistem

3.2 Perancangan Perangkat Keras *(Hardware)*

3.2.1 Power Supply

Agar alat ini mudah dibawa dan fleksibel maka untuk catu daya menggunkan baterai 9 Volt. Dari baterai masuk ke rangkaian *power supply* dengan menggunakan regulator 7805 untuk mengghasilkan outputan tegangan 5 Volt untuk mensupply rangkaian ADC dan mikrokontroler.



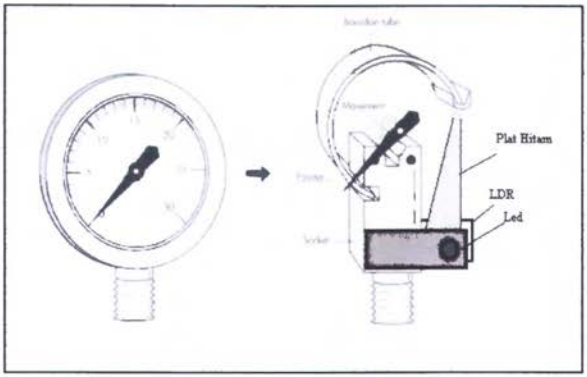
Gambar 3.3 Rangkaian Power Supply

3.2.2 Perencanaaan Sensor Tekanan

Dalam tugas akhir ini menggunkan sensor *pressure gauge* yang telah dimodifikasi dengan menggunakan LDR dan led. LDR dan led dipasang berhadapan yang dihalangi oleh plat hitam. LDR dan led ini dipasang pada bagian leher *pressure gauge*, sedangkan plat hitam dipasang pada *link*. LDR dan led dihubungkan dengan resistor sebesar 470 Ω dan multitone sebagai pengatur outputan dari sensor.

Prinsip kerja dari sensor ini adalah ketika ada tekanan maka *bourdon* akan mengembang. Mengembangnya bourdon membuat *link* terangkat sehingga menyebabkan plat hitam juga terangkat. Dengan terangkatnya plat hitam maka cahaya yang diterima oleh LDR semakin banyak. Sesuai dengan

prinsip kerja dari LDR sendiri semakin banyak cahaya yang diterima maka semakin kecil tegangan yang terukur.

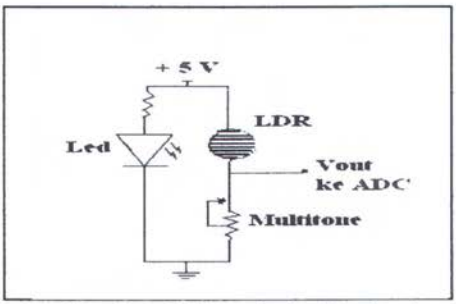


Gambar 3.4 Rangkaian Sensor

Oleh karena itu tekanan dapat diukur dengan tegangan yang terukur yang kemudian dimasukkan ke ADC. Untuk mendapatkan inputan sensor 0 – 5 Volt maka *pressure gauge* disetting dengan mengubah-ubah multitone.

3.2.3 Rangkaian Pembagi

Rangkaian pembagi ini digunakan untuk mengeset tegangan keluaran dari LDR untuk memenuhi range 0 sampai 5 Volt sebelum masuk ke ADC.



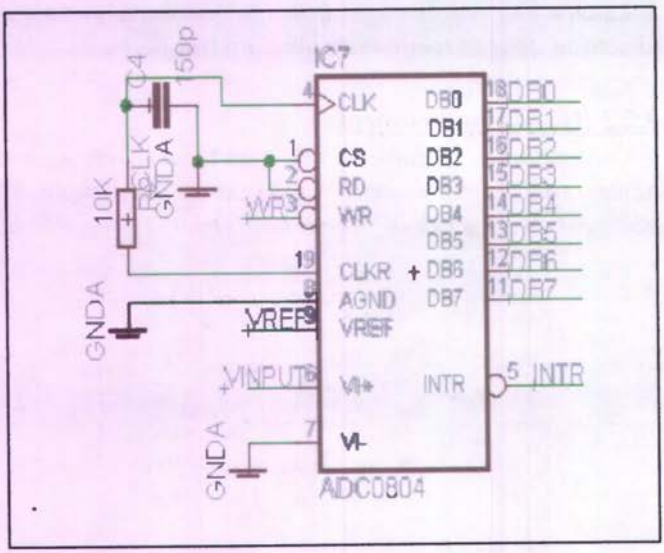
Gambar 3.5 Rangkaian Pembagi

Dengan menggunakan rumus (2.1) maka dapat kita setting keluaran rangkaian pembagi yang masuk ke ADC dengan menyesuaikan buka tutupnya plat hitam. Perubahan hambatan LDR dihasilkan oleh besar kecilnya cahaya yang jatuh pada LDR, yang tergantung buka utupnya plat hitam. Oleh karena itu semakin besar tekanan maka semaki banyak pula cahaya yang datang.

3.2.4 Rangkaian Analog to Digital (ADC)

Rangkaian *Analog to Digita*l (ADC) menggunakan ADC 0804 buatan National Semiconductor . ADC ini merupakan ADC 8 bit dengan kapasitas $2^8 = 255$. Adapun spesifikasinya adalah :

- ADC 8 bit = $2^8 = 256$ resolusinya
  Keluarannya 8 bit pararel
- Mempunyai waktu konversi 100 mikro detik
- Input 0 sampai 5 volt dengan *supply* tunggal



Gambar 3.6 Rangkaian ADC 0804

Pada rangkaian ini ADC berfungsi untuk mengubah sinyal analog yaitu berupa tegangan menjadi biner yang nantinya akan diolah oleh mikrokontroler.

Pembacaan ADC dilakukan dengan mengirimkan sinyal /WR pada pin ADC oleh mikrokontroler dan apabila ADC telah selesai melakukukan konversi pada inputan saat itu maka dia akan mengubah level tegangan pada pin /INTR dari level 1 ke level 0 dan bila hal ini terjadi maka mikrokontroler akan membaca atau memasukkan data dari ADC, dalam hal ini *port* yang dipakai untuk membaca data *output* dari ADC adalah port 0 dan control ADC dilakukan oleh port 1.4 untuk sinyal /INTR dan port 1.4 untuk sinyal /WR. Pin *Read* dan *Enable* pada rangkaian ADC dihubungkan dengan *ground,* dan ADC ini menggunakan *supply* tunggal yaitu tegangan 5 Volt.

3.2.5 Mikrokontroler

Mikrokontroler merupakan unsur yang paling penting dalam alat ini yaitu sebagai pemroses data yang masuk ADC 0804, pengolah data dan menampilkannya dalam *display* LCD.

Tabel 3.1 Jenis Kapasitas ROM Mikrokontroler Atmel

Jenis	Kapasita ROM	Kapasitas Ram
AT89C51	4 KB	128
AT89S51	4 KB	128
AT89S53	20 KB	3 x 128

Pada mikrokontroler ini digunakan internal ROM dan RAM sehingga kaki EA/VPP diberi level tegangan HIGH (5 Volt), untuk tiap jenis mikrokontroler keluaran Atmel seri 89S memiliki perbedaan dalam hal kapasitas ROM dan RAM.

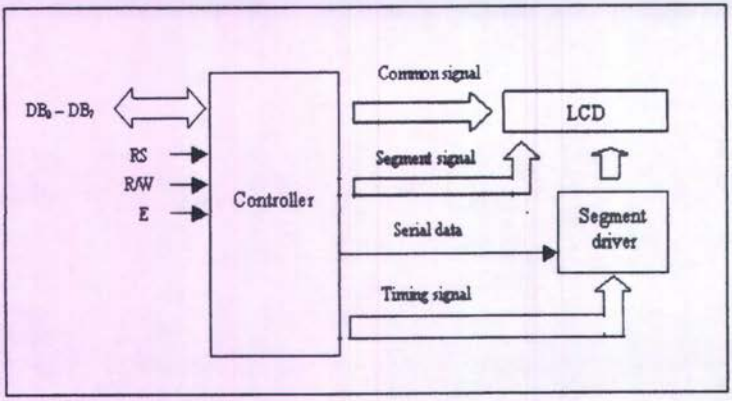
Jenis lain dari mikrokontroler keluaran Atmel masih banyak lagi. Ketiga mikrokontroler tersebut banyak terdapat di pasaran.

Adapun penggunaan port dalam mikrokontroler ini adalah :

1. Port 1.0 – Port 1.7 = Data ADC
2. Port 0 = Data LCD
3. Port 3.0 = Tombol awal
4. Port 3.1 = Tombol Start
5. Port 3.3 = Tombol stop
6. Port 1.5 – Port 1.7 = Downloader

3.2.6 Display LCD *(Liquid Crystal Display)*

LCD yang digunakan 2 baris x 16 kolom. LCD memiliki memori internal yang berisi definisi karakter sesuai dengan standar ASCII (CGROM – *Character Generator ROM*) dan memori sementara (RAM) yang bisa digunakan bila memerlukan karakter khusus (berkapasitas 8 karakter).



Gambar 3.7 Diagram blok LCD

RAM ini juga berfungsi untuk menyimpan karakter yang ingin ditampilkan di LCD.

- **Fungsi Pin-pin LCD**

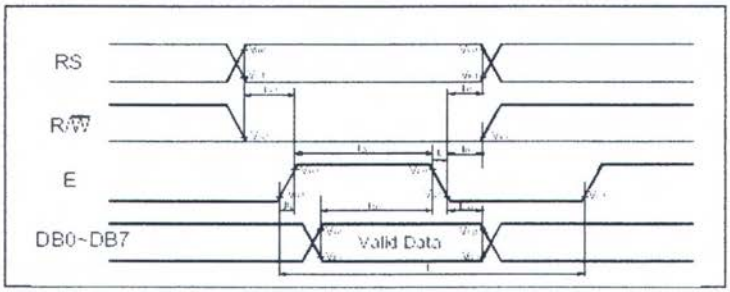
Tabel 3.2 Tabel fungsi pin LCD

Sinyal	Input/Output	Fungsi
DB_0 - DB_7	Input/Output	Data
E	Input	Sinyal operasi
R/W	Input	Sinyal seleksi *Read* and *Write* 0 : *Write* 1 : *Read*
RS	Input	Sinyal seleksi Register 0 : Register instruksi 1 : Register data
V_{LC}	-	Pengaturan contrast
V_{DD}	-	+5V
V_{SS}	-	Ground

Operasi Write adalah waktu penulisan data pada saat data dimasukkan ke LCD.

- **Operasi Write**

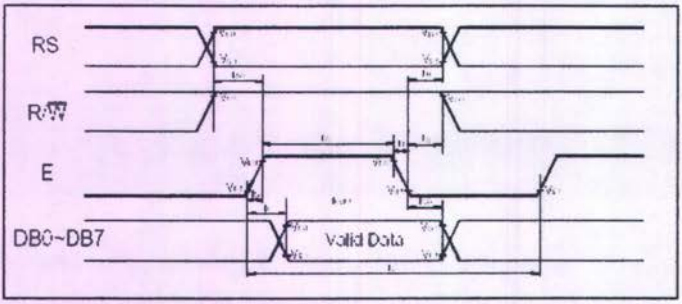
Pada Gambar 3.8 dapat dilihat diagram pada waktu pengisian data ke Mikrokontroller.



Gambar 3.8 Timing diagram data write

- **Operasi Read**

Sedangkan Operasi read adalah pembacaan pada saat data masuk ke dalam LCD



Gambar 3.9 Timing diagram data read

Pin-pin yang digunakan untuk control LCD adalah :

1. RS : Pin 2.0
2. E : Pin 2.1

Pin untuk LCD sendiri adalah pin pengisian data agar data dapat diterima dan diolah melalui Mikrokontroller ke LCD. Sebelum data dibaca oleh LCD maka data diisi oleh RS.

Untuk pemograman mikrokontroler menggunakan software atmel ISP yang merupakan *software* untuk download ke mikrocontroler.

Program yang kita tulis dalam notepad harus disimpan dalam ekstensi .asm dan kemudian dikonversikan dalam bentuk bilangan heksadesimal yang secara otomatis akan mengoreksi error program. Bila *error* adalah nol maka program bisa di downloadkan menuju mikrokontroler.

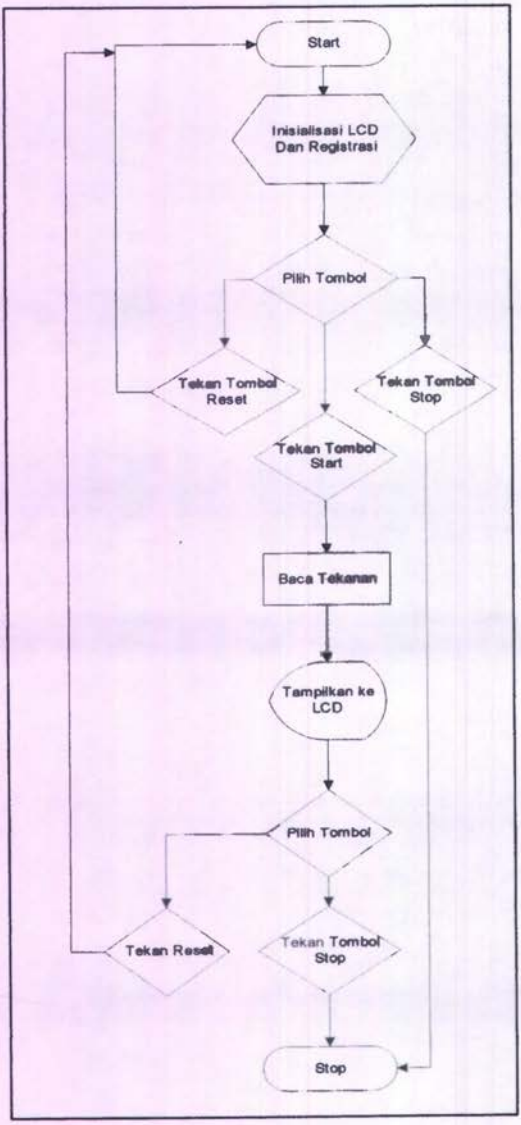
3.3 Perancangan Perangkat Lunak *(Software)*

Perancangan *software* alat ukur tekanan dengan Mikrokontroller AT89S51 dibuat dan disesuaikan dengan

keberadaan *hardware* yang digunakan. Perancangan *software* pada dasarnya terdiri dari beberapa bagian pokok yaitu :

1. *Software* yang dipakai untuk pembacaan Mikrokontroler
2. *Software* yang dipakai untuk pembacaan ADC oleh mikrokontroler yang digunakan sebagai data pengolahan data
3. *Software* pengkonversi nilai desimal ke nilai tegangan
4. *Software* penampil ke display LCD

*A*dapun flowchart dari perancangan *software* adalah :



Gambar 3.10 Flowchart Perancangan Software

3.3.1 Perancangan Bahasa Assembly

Perancangan bahasa assembly pada sistem ini terbagi menjadi beberapa yaitu :

1. Pengisian Assembly ke Mikrokontroller AT89S51

Program Atmel MCS51 merupakan program yang pengisian yang dijalankan secara serial pada *operating system windows* melalui *port* 1.5, 1.6 dan 1.7. Pertama kita buat listing program *assembly* terlebih dahulu pada NotePad yang kemudian disimpan dengan extension .asm. Kemudian file .asm *dicompile* atau diuji kebenarannya dengan menggunakan program asm51 yang dijalankan pada *operating system* DOS. Proses pengujian selesai dengan menghasilkan file yang berakhiran .Hex dan .LST. Serta jumlah kesalahan dari Program yang kita buat. Jika kesalahan yang ada *(Error)* sama dengan Nol (Tidak ada kesalahan pada penulisan Listing Program), maka selanjutnya kita lakukan pengisian *(Download* program) ke IC Mikro AT89S51. Algoritma *software* monitoring tekanan bisa terlihat dibawah ini:

- Lakukan proses pengecekan *hardware*
- Bila kondisi baik lakukan pengukuran
- Alokasi data memori pengukuran dan

Alokasi data port input listing program terlihat dbawah ini

```
Data_ADC    equ    60h
Rd_ADC      bit    P3.6
Wr_ADC      bit    P3.7
T_Start     bit    p3.1
T_Stop      bit    p3.2
Data_P      equ    61h
H_rb        equ    63h
H_r         equ    64h
H_p         equ    65h
H_s         equ    66h
```

❖ Baca data inputan dan tampilan besar tekanan terukur, dengan listing program :

```
;((22 * D)/100 ) - ((435)/100)
mov             r6,#12
Loop_50:        mov     r7,#29
Loop_10:        mov     a,H_S
                jz      Zero_Sat
                dec     H_S
                jmp     K_K
Zero_sat:       mov     a,H_P
                jz      Zero_Pul
                dec     H_P
                mov     H_S,#9
                jmp     K_K
Zero_pul:       mov     a,H_R
                jz      Zero_Rat
                dec     H_R
                mov     H_P,#9
                mov     H_S,#9
                jmp     K_K
Zero_Rat:       mov     a,H_Rb
                jz      Zero_Rib
                dec     H_Rb
                mov     H_R,#9
                mov     H_P,#9
                mov     H_S,#9
                nop
K_K:            djnz    R7,Loop_10
                djnz    r6,Loop_50
Zero_Rib:       nop
                ret
Konversi_ADC:
                clr     c
```

```
                Mov     a,Data_Adc
                mov     b,#100
                div     ab
                mov     H_rb,a
                mov     a,b
                mov     b,#10
                div     ab
                mov     H_R,a
                mov     H_P,b
                ret
Tampil_LCD:     mov     Dptr,#Karakter
                Mov     R1,#0c5h
                Call    Write_ins
                Mov     a,H_Rb
                Movc    a,@a+Dptr
                Mov     R1,a
                Call    Write_Data
                Mov     R1,#0c6h
                Call    Write_ins
                Mov     a,H_R
                Movc    a,@a+Dptr
                Mov     R1,a
                Call    Write_Data
                Mov     R1,#0c8h
                Call    Write_ins
                Mov     a,H_p
                Movc    a,@a+Dptr
                Mov     R1,a
                Call    Write_Data
                Mov     R1,#0c9h
                Call    Write_ins
                Mov     a,H_s
                Movc    a,@a+Dptr
```

```
Mov     R1,a
Call    Write_Data
Mov     R1,#0cfh
Call    Write_ins
ret
```

❖ Untuk pemilihan dalam menentukan perintah tombol dengan listing program dibawah ini

```
jnb     T_Start,M_Start
jnb     T_Stop,M_Stop
```

Pada awal program diberikan inisialisasi untuk komponen yang dipakai antara lain LCD dan register.

Untuk LCD, data hasil *encoding* tersebut langsung dijadikan data untuk LCD. Hal-hal yang perlu diperhatikan dalam meulis sebuah karakter pada LCD yaitu :

- Pin **R/W** di serikan logic "0", sehingga operasi yang dilakukan adalah operasi *write*/tulis ke LCD.
- Pin **E,** untuk melakukan semua operasi LCD maka pin ini haruslah ditoggle (dari logic "0" ke "1").
- Untuk pin **RS** dilakukan :
  - **RS** diberikan logic "0" untuk melakukan mengirimkan instruksi pada LCD. Perintah ini biasanya digunakan untuk memesan "lokasi" pada LCD, "lokasi" mana yang akan dituliskan sebuah karaker.
  - **RS** diberikan logic "1" untuk mengirimkan data pada LCD, yaitu memberikan data yang akan dituliskan ke LCD.

Pengujian LCD dilakukan secara *software*, dimana pada layar LCD akan menampilkan kalimat yang sudah diset didalam mikrokontroler. Adapun Listing Program Test LCD sebagai berikut:

Listing Test LCD

```
Start:      mov     r1,#03fh
```

```
            call    write_ins
            mov     r1,#0dh
            call    write_ins
            mov     r1,#06h
            call    write_ins
            mov     r1,#01h
            call    write_ins
            mov     r1,#0c0h
            call    write_ins
            nop
            jmp     cek_ukur
write_ins:  mov     p0,r1
            clr     p2.0
            call    delcd
            setb    p2.1
            clr     p2.1
            call    delcd
            ret
write_data: mov     p0,r1
            setb    p2.0
            call    delcd
            setb    p2.1
            clr     p2.1
            call    delcd
            ret
delcd:      mov   20h,#0fh
delcd1:     mov   19h,#0fh
            djnz  19h,$
            djnz  20h,delcd1
            ret
delcdx:     mov     20h,#0f4h
delcdy:     mov     19h,#0ffh
            djnz    19h,$
```

```
        djnz    20h,delcdy
        ret
barisx: mov     r4,#16
        mov     r1,#80h
        call    write_ins
tulisx: clr     a
        movc    a,@a+dptr
        mov     r1,a
        inc     dptr
        call    write_data
        call    delcdx
        djnz    r4,tulisx
barisy: mov     r4,#16
        mov     r1,#0c0h
        call    write_ins
tulisy: clr     a
        movc    a,@a+dptr
        mov     r1,a
        inc     dptr
        call    write_data
        call    delcdx
        djnz    r4,tulisy
        call    delcd
        djnz    r3,barisx
        ret
```

3.3.2 Pengukuran dan Deteksi Sinyal

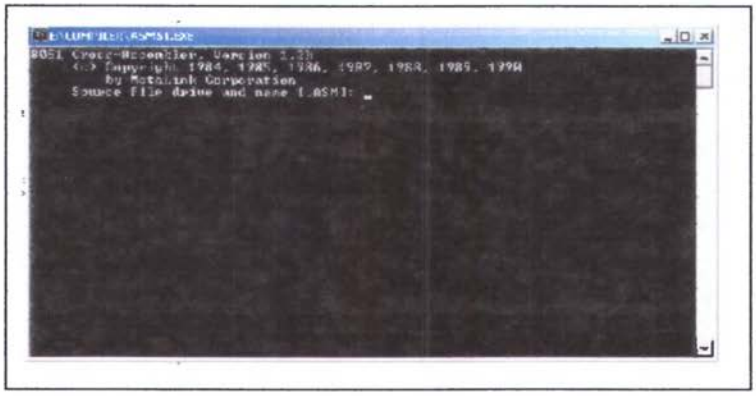
Software pengukur pada dasarnya hanya terdiri dari pengukuran dan perhitungan tekanan sensor. Data yang masuk dari ADC ke mikrokontroler berupa data pararel yang bila dibaca oleh mikrokontroler berupa bilangan desimal antara 0 sampai 255 maka diperlukan program untuk konversi tegangan ke tekanan.

Pengukuran dan perhitungan tekanan tergantung pada tegangan output dari sensor. Output dari sensor *pressure* mulai dari 0,13 sampai dengan 1,70 Volt untuk tekanan antara 0 sampai 15 Psi.

Untuk mengkondisikan pada display sesuai dengan tekanan yang terukur pada *pressure gauge*, maka saat tekanan 15 Psi yang terukur pada *pressure gauge* pada pembacaan ADC dikondisikan 170 dengan cara mengubah nilai V_{Ref} ADC.

Pengetesan sensor menggunakan *software* pada mikrokontroler, dimana listing programnya sama dengan program test ADC, tekanan dari sensor ditampilkan ke LCD. Mikrokontroler ISP Program yang dijalankan pada *Operating System Windows*. Pertama kali pada proses pengisian kita melakukan cek terhadap IC AT89S51 baik itu komunikasinya ataupun keberadaannya. Setelah komunikasi antara komputer dengan *Minimum System* AT89S51 terdeteksi, maka selanjutnya kita mengambil program.hex yang akan kita isi kemudian disimpan di *Buffer* dari program ISP.

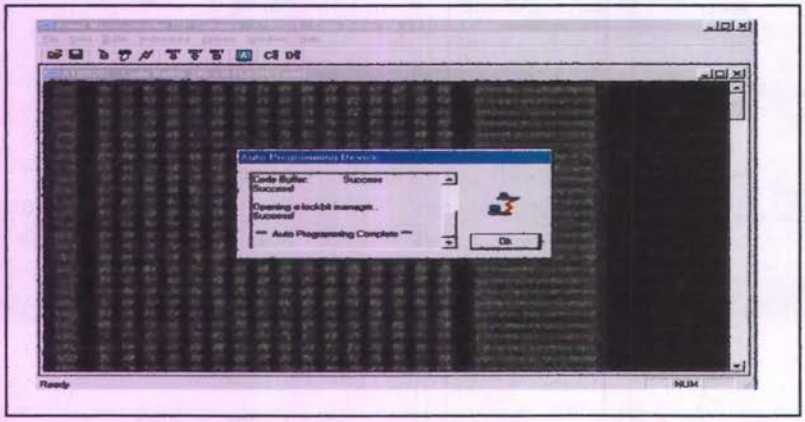
Setelah selesai membuat assembly program dilanjutkan dengan mengkompail assembley dengan kompiler asm51 dengan tampilan dos seperti terlihat pada Gambar 3.11:



Gambar 3.11 Kompile Program Asm Dengan Asm51

MILIK PERPUSTAKAAN ITS

Hasil kompiler akan menghasilkan program.Hex yang akan kita isi sudah ada pada *Buffer*, kemudian kita lanjutkan dengan melakukan pengisian. Pengisian dilakukan dengan menekan tombol auto (*Erase, Write* dan *Verify*) maka proses pengisian akan berlangsung, sampai jumlah byte program yang akan diisi komplit. Maka dengan sendirinya program ISP akan menghentikan pengisian. Setelah selesai program pengisian maka, IC AT89S51 yang ada sudah terisi program yang kita inginkan, selanjutnya IC tersebut bisa kita aplikasikan pada rangkaian yang kita buat.



Gambar 3.12 Program Pengisian Mikro AT9S51

BAB IV
PENGUJIAN DAN ANALISA

Setelah melalui proses perancangan dan pembuatan alat dalam tugas akhir ini, maka perlu dilakukan suatu pengujian untuk mengetahui karakteristik dari alat tersebut.

4.1 Pengukuran Data

Setelah melakukan serangkaian pengujian alat yang akan didapatkan data pengukuran maka ada beberapa point yang mana pada nantinya akan dibahas. Dan untuk mengetahui baik tidaknya kerja alat maka perlu dilakukan kalibrasi dengan mencari *uncertainly type* U_A.

4.1.1 Analisa Tipe A,U_a

Pada tipe ini biasanya ditandai dengan adanya data pengukuran, misalnya n kali pengukuran, maka selanjutnya dari dat tersebut akan ditemukan nilai rata-ratanya, standart deviasi, atau repetabilitynya. Bentuk kurva dari tipe ini adalah sebaran Gauss. Rumus umum untuk tipe A:

$$U_a = \frac{D}{\sqrt{n}} \quad \text{.....(4.1)}$$

Dimana D = standart deviasi

4.1.2 Standart Deviasi

Standart deviasi adalah pengembangan dari suatu pengukuran, dimana dengan memberikan nilai n yang menunjukkan banyaknya data. Deviasi ini merupakan perbedaan antara masing-masing data dan nilai rata-rata, deviasi dapat dirumuskan sebagai berikut :

$$D = (\bar{x} - x_i) \quad \text{.....(4.2)}$$

4.1.3 Simpangan Baku

Adalah nilai dari akar dari jumlah kuadrat deviasi dibagi dengan (n -1).

$$S(x_i) = \sqrt{\frac{\sum D^2}{n-1}} \quad \text{.....(4.3)}$$

4.1.4 Akurasi

Akurasi didefinisikan sebagai keterdekatan hasil pengukuran suatu alat ukur terdapat suatu nilai standart yang disepakati, atau terhadap suatu nilai yang benar (*true value).*

$$A = 1 - \left| \frac{Yn - Xn}{Yn} \right| \quad \text{.....(4.4)}$$

Dimana A = Akurasi
Yn = Nilai sebenarnya
Xn = nilai yang terukur

4.1.5 Presisi

Presisi adalah keterdekatan hasil pengukuran yang dilakukan berulang-ulang terhadap rata-rata pengukuran.

$$\text{Presisi} = 1 - \left| \frac{Xn - \overline{Xn}}{Xn} \right| \quad \text{.....(4.5)}$$

Dimana Xn = nilai dari pengukuran ke-n

$\overline{Xn}$ = nilai rata-rata pengukuran

4.1.6 Histerisis

Gejala histerisis muncul berupa perbedaan hasil-hasil pengukuran yang dilakukan secara naik-turun.

Histerisis = Data naik-data turun .....(4.6)

4.1.7 Resolusi

Resolusi suatu alat ukur adalah perubahan terkecil dalam nilai yang diukur dimana instrument ukur dapat memberikan respon atau besarnya perubahan input minimum yang perlu dimasukkansedemikian sehingga pada keluaran terdapat perubahan yang dapat dibaca dengan benar.

4.1.8 Sensitivitas

Sensitivitas alat ukur adalah perbandingan antara perubahan keluaran alat ukur terhadapa perubahan masukan terkecil.

4.2 Pengambilan data sensor tekanan

Pengambilan data pada sensor tekanan ini dimaksudkan untuk mencari nilai *uncertainly* dari pengukuran *pressure gauge sensor* yang dibandingkan dengan *regulator pressure.* Pengambilan data dilakukan dengan mengukur setiap kenaikan 5 Psi sebanyak 6 kali setiap tingkat. Data pengukuran seperti pada tabel 4.1 di bawah ini.

Tabel 4.1 Data pengukuran naik pressure gauge tekanan 0 Psi

Data ke-	Regulator Pressure (Psi)	Pressure Gauge (Psi)	D	D^2
1.	0	0	0	0
2.	0	0	0	0
3.	0	0	0	0
4.	0	0	0	0
5.	0	0	0	0
6	0	0	0	0
	$\bar{x} = 0$	$\bar{x} = 0$		$\Sigma = 0$

$$S(x_i) = \frac{0}{5} = 0$$

$$U_a = \frac{0}{\sqrt{6}} = 0$$

Jadi hasil pengukuran = 0 ± 0

Tabel 4.2 Data pengukuran naik pressure gauge tekanan 5 Psi

Data ke-	Regulator Pressure (Psi)	Pressure Gauge (Psi)	D	D^2
1.	6	5	0	0
2.	6	5	0	0
3.	6	5	0	0
4.	6	5	0	0
5.	6	5	0	0
6.	6	5	0	0
	$\bar{x} = 6$	$\bar{x} = 5$		$\Sigma = 0$

$$S(x_i) = \frac{0}{5} = 0$$

$$U_a = \frac{0}{\sqrt{6}} = 0$$

Jadi hasil pengukuran = 5 ± 0

Tabel 4.3 Data pengukuran naik pressure gauge tekanan 10 Psi

Data ke-	Regulator Pressure (Psi)	Pressure Gauge (Psi)	D	D^2
1.	11	10	0	0
2.	11	10	0	0
3.	11	10	0	0
4.	11	10	0	0
5.	11	10	0	0
6.	11	10	0	0
	$\bar{x} = 11$	$\bar{x} = 10$		$\Sigma = 0$

$$S(x_i) = \frac{0}{5} = 0$$

$$U_a = \frac{0}{\sqrt{6}} = 0$$

Jadi hasil pengukuran = 10 ± 0

Tabel 4.4 Data pengukuran naik pressure gauge tekanan 15 Psi

Data ke-	Regulator Pressure (Psi)	Pressure Gauge (Psi)	D	D^2
1.	17	15	0	0
2.	17	15	0	0
3.	17	15	0	0
4.	17	15	0	0
5.	17	15	0	0
6.	17	15	0	0
	$\bar{x} = 17$	$\bar{x} = 15$		$\Sigma = 0$

$$S(x_i) = \frac{0}{5} = 0$$

$$U_a = \frac{0}{\sqrt{6}} = 0$$

Jadi hasil pengukuran = 15 ± 0

Tabel 4.5 Data pengukuran turun pressure gauge tekanan 15 Psi

Data ke-	Regulator Pressure (Psi)	Pressure Gauge (Psi)	D	D^2
1.	17	15	0	0
2.	17	15	0	0
3.	17	15	0	0
4.	17	15	0	0
5.	17	15	0	0
6.	17	15	0	0
	$\bar{x} = 17$	$\bar{x} = 15$		$\Sigma = 0$

$$S(x_i) = \frac{0}{5} = 0$$

$$U_a = \frac{0}{\sqrt{6}} = 0$$

Jadi hasil pengukuran = 15 ± 0

Tabel 4.6 Data pengukuran turun pressure gauge tekanan 10 Psi

Data ke-	Regulator Pressure (Psi)	Pressure Gauge (Psi)	D	D^2
1.	11	10	0	0
2.	11	10	0	0
3.	11	10	0	0
4.	11	10	0	0
5.	11	10	0	0
6.	11	10	0	0
	$\bar{x} = 11$	$\bar{x} = 10$		$\Sigma = 0$

$$S(x_i) = \frac{0}{5} = 0$$

$$U_a = \frac{0}{\sqrt{6}} = 0$$

Jadi hasil pengukuran = 10 ± 0

Tabel 4.7 Data pengukuran turun pressure gauge tekanan 5 Psi

Data ke-.	Regulator Pressure (Psi)	Pressure Gauge (Psi)	D	D^2
1.	6	5	0	0
2.	6	5	0	0
3.	6	5	0	0
4.	6	5	0	0
5.	6	5	0	0
6.	6	5	0	0
	$\bar{x} = 6$	$\bar{x} = 5$		$\Sigma = 0$

$$S(x_i) = \frac{0}{5} = 0$$

$$U_a = \frac{0}{\sqrt{6}} = 0$$

Jadi hasil pengukuran = 5 ± 0

Tabel 4.8 Data pengukuran turun pressure gauge tekanan 0 Psi

Data ke-	Regulator Pressure (Psi)	Pressure Gauge (Psi)	D	D^2
1.	0	0	0	0
2.	0	0	0	0
3.	0	0	0	0
4.	0	0	0	0
5.	0	0	0	0
6	0	0	0	0
	$\bar{x} = 17$	$\bar{x} = 15$		$\Sigma = 0$

$$S(x_i) = \frac{0}{5} = 0$$

$$U_a = \frac{0}{\sqrt{6}} = 0$$

Jadi hasil pengukuran = 0 ± 0

4. 3 Pengambilan data display

Pengujian ini dilakukan dengan mengambil data setiap kenaikan 5 Psi sampai 15 Psi untuk mencari data penyesuaian besarnya tekanan yang terukur pada *pressure gauge* dengan tekanan yang ditampilkan di LCD.data pengujian seperti pada tabel di bawah ini.

Tabel 4.9 Data pengukuran naik tekanan 0 Psi pada display

Data ke-	Pressure Gauge (Psi)	Display (Psi)	D	D^2	A	Presisi
1.	0	0.00	0	0	1	1
2.	0	0.00	0	0	1	1
3.	0	0.00	0	0	1	1
4.	0	0.00	0	0	1	1
5.	0	0.00	0	0	1	1
6.	0	0.00	0	0	1	1
	$\bar{x} = 0$	$\bar{x} = 0.00$		$\Sigma = 0$		

$$S(x_i) = \sqrt{\frac{0}{5}} = 0$$

$$U_a = \frac{0}{\sqrt{6}} = 0$$

Jadi hasil pengukuran = 0 ± 0

Tabel 4.10 Data pengukuran naik tekanan 5 Psi pada display

Data ke-	Pressure Gauge (Psi)	Display (Psi)	D	D^2	A	Presisi
1.	5	4.34	0.06	0.004	0.87	0.99
2.	5	4.34	0.06	0.004	0.87	0.99
3.	5	4.34	0.06	0.004	0.87	0.99
4.	5	4.34	0.06	0.004	0.87	0.99
5.	5	4.51	-0.11	0.012	0.90	0.98
6.	5	4.51	-0.11	0.012	0.90	0.98
	$\bar{x} = 5$	$\bar{x} = 4.40$		$\Sigma = 0.040$		

$$S(x_i) = \sqrt{\frac{0.04}{5}} = 0,09$$

$$U_a = \frac{0.9}{\sqrt{6}} = 0,37$$

Jadi hasil pengukuran = 4,40 ± 0,37

Tabel 4.11 Data pengukuran naik tekanan 10 Psi pada display

Data ke-	Pressure Gauge (Psi)	Display (Psi)	D	D^2	A	Presisi
1.	10	9.10	0.34	0.116	0.91	0.96
2.	10	9.61	-0.17	0.029	0.96	0.98
3.	10	9.61	-0.17	0.029	0.96	0.98
4.	10	9.61	-0.17	0.029	0.96	0.98
5.	10	9.10	0.34	0.116	0.91	0.96
6.	10	9.61	-0.17	0.029	0.96	0.98
	$\bar{x} = 10$	$\bar{x} = 9.44$		$\Sigma = 0.348$		

$$S(x_i) = \sqrt{\frac{0.348}{5}} = 0,26$$

$$U_a = \frac{0.26}{\sqrt{6}} = 0,11$$

Jadi hasil pengukuran = 9,44 ± 0,11

Tabel 4.12 Data pengukuran naik tekanan 15 Psi pada display

No.	Pressure Gauge (Psi)	Display (Psi)	D	D^2	A	Presisi
1.	15	15.05	0.03	0.001	0.10	0.10
2.	15	15.05	0.03	0.001	0.10	0.10
3.	15	15.22	-0.2	0.040	0.99	0.99
4.	15	14.88	0.14	0.020	0.10	0.99
5.	15	14.88	0.14	0.020	0.10	0.99
6.	15	15.05	0.03	0.001	0.10	0.10
	$\bar{x} = 15$	$\bar{x} = 15.02$		$\Sigma = 0.083$		

$$S(x_i) = \sqrt{\frac{0.083}{5}} = 0,13$$

$$U_a = \frac{0.13}{\sqrt{6}} = 0,05$$

Jadi hasil pengukuran = 15,02 ± 0,05

Tabel 4.13 Data pengukuran turun tekanan 15 Psi pada display

No.	Pressure Gauge (Psi)	Display (Psi)	D	D^2	A	Presisi
1.	15	15.05	0	0	0.10	0
2.	15	15.05	0	0	0.10	0
3.	15	15.05	0	0	0.10	0
4.	15	15.05	0	0	0.10	0
5.	15	15.05	0	0	0.10	0
6.	15	15.05	0	0	0.10	0
$\bar{x}$	15	15.05		$\Sigma = 0$		

$$S(x_i) = \sqrt{\frac{0}{5}} = 0$$

$$U_a = \frac{0}{\sqrt{6}} = 0$$

Jadi hasil pengukuran = 15,02 ± 0

Tabel 4.14 Data pengukuran turun tekanan 10 Psi pada display

No.	Pressure Gauge (Psi)	Display (Psi)	D	D^2	A	Presisi
1.	10	11.48	0	0	0.85	0
2.	10	11.48	0	0	0.85	0
3.	10	11.48	0	0	0.85	0
4.	10	11.48	0	0	0.85	0
5.	10	11.48	0	0	0.85	0
6.	10	11.48	0	0	0.85	0
	$\bar{x} = 10$	$\bar{x} = 11.48$		$\Sigma = 0$		

$$S(x_i) = \sqrt{\frac{0}{5}} = 0$$

$$U_a = \frac{0}{\sqrt{6}} = 0$$

Jadi hasil pengukuran = 9,44 ± 0

Tabel 4.15 Data pengukuran turun tekanan 5 Psi pada display

No.	Pressure Gauge (Psi)	Display (Psi)	D	D^2	A	Presisi
1.	5	4.00	-0.06	0.004	0.80	0.99
2.	5	4.00	-0.06	0.004	0.80	0.99
3.	5	4.00	-0.06	0.004	0.80	0.99
4.	5	4.00	-0.06	0.004	0.80	0.99
5.	5	3.49	0.45	0.203	0.70	0.87
6.	5	4.17	-0.23	0.053	0.83	0.95
	$\bar{x} = 5$	$\bar{x} = 3.94$		$\Sigma = 0.270$		

$$S(x_i) = \sqrt{\frac{0.270}{5}} = 0,23$$

$$U_a = \frac{0.23}{\sqrt{6}} = 0,09$$

Jadi hasil pengukuran = 3.94 ± 0,09

Tabel 4.16 Data pengukuran turun tekanan 0 Psi pada display

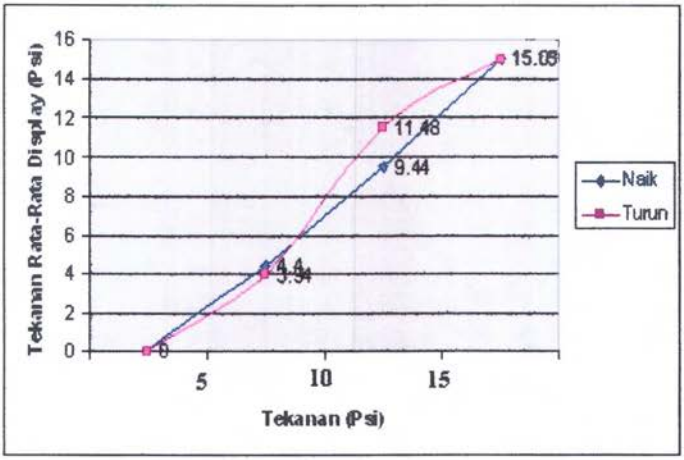
No.	Pressure Gauge (Psi)	Display (Psi)	D	D^2	A	Presisi
1.	0	0.00	0	0	1	1
2.	0	0.00	0	0	1	1
3.	0	0.00	0	0	1	1
4.	0	0.00	0	0	1	1
5.	0	0.00	0	0	1	1
6.	0	0.00	0	0	1	1
	$\bar{x} = 0$	$\bar{x} = 0.00$		$\Sigma = 0$		

$$S(x_i) = \sqrt{\frac{0}{5}} = 0$$

$$U_a = \frac{0}{\sqrt{6}} = 0$$

Jadi hasil pengukuran = 0 ± 0

Dari data pengukuran naik Dan pengukuran turun maka dapat dicari histerisnya yaitu dengan cara mengurangi rata-rata data naik dengan rata-rata data turun.



Gambar 4.1 Grafik Histerisis

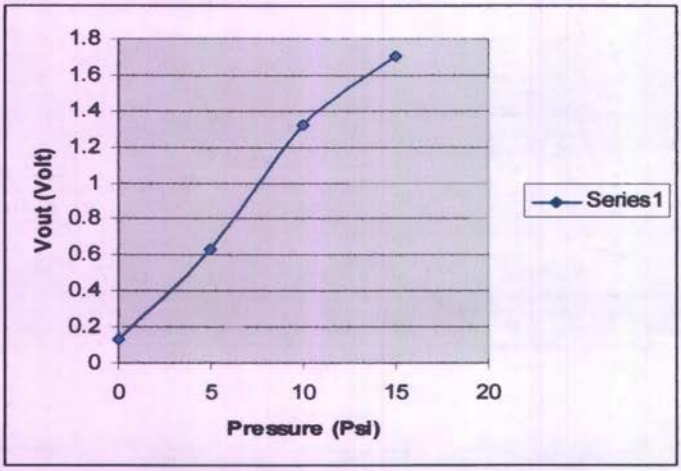
4.4 Pengambilan data tegangan sensor

Keluaran dari sensor yang berupa tegangan langsung masuk ke ADC yang mempunyai range sinyal inputan 0 – 5 Volt. Oleh karena itu perlu diambil data tegangan keluaran sensor seperti tabel 4.17.

Tabel 4.17 Pengukuran sensor

No.	Tekanan (Psi)	Tegangan Rata-rata (Volt)
1.	0	0.13
2.	5	0.63
3.	10	1.32
4.	15	1.70

Dari hasil pengukuran sensor diketahui bahwa range pengukuran antara 0.17 – 1.70 Volt dan tekanan antara 0 – 15 psi Dari data diatas maka dapat dibuat grafik antara tekanan terukur dalam hal ini adalah tekanan pada *pressure gauge* sensor dengan tegangan yang keluar dari sensor seperti terlihat pada grafik di bawah ini.



Gambar 4.2 Grafik hubungan tegangan – tekanan

Dari data pengambilan data tegangan pada sensor maka dapat dikonverrsikan ke dalam bentuk Psi dengan melakukan konversi pada *software.*

4.5 Analisa Data

Setelah melakukan pengambilan data maka dapat dianalisa dengan mencari nilai *uncertainty,* akurasi dan presisi. Untuk data perbandingan antara *pressure gauge* dengan regulator *pressure* tidak perlu mencari nilai akurasi dan presisi. Yang paling penting untuk dianalisa adalah data perbandingan antara *pressure gauge* pada sensor dengan tekanan yang ditampilkan di LCD.

4.5.1 Analisa Perhitungan

Untuk perhitungan *uncertainty* seperti rumus (4.1). maksud dari ilai *uncertainty* adalah untuk mencari range *truevalue* dari hasil pengukuran. Misalnya pada pengukuran naik pada tekanan 5 Psi didapatkan hasil pengukuran 3.94 ± 0,09. Artinya tekanan yang sebenarnya *(truevalue)* adalah antara 3,85 Psi sampai 4,03 Psi.

Sedangkan untuk nilai akurasi dari pencuplikan data dimaksudkan untuk mengetahui batas penyimpangan pengukuran atau batas kesalahan intrinsik yaitu kesalahan yang muncul pada saat alat ukur digunakan pada kondisi referensi dan diakibatkan kesalahan pada komponen-komponen pembentuk dalam hal ini seluruh rangkaian alat. Misalnya pada pengukuran naik tekanan 5 Psi terhitung skurasi pada pencuplikan data pertama adalah 0,87 dan pada pencuplikan data kelima nilai akurasinya adalah 0,90. artinya tekanan yang sebenarnya (*truevalue)* pada pencuplikan data pertama adalah antara 3,44 Psi sampai dengan 5,21 Psi.

Untuk perhitungan presisi dilakukan untuk melihat kedekatan hasil pengukuran yang dilakukan berulang-ulang terhadap rata-rata pengukuran. Suatu alat ukur yang presisi belum tentu menjamin akurasinya karena presisi ini hanya ditentukan oleh nilai yang dihasilkan oleh alat ukur yang bersangkutan tanpa harus dibandingkan dengan besaran terukur yang sebenarnya yang dihasilkan oleh peralatan standart. Misalnya pada data pengukuran naik 5 Psi mempunyau presisi antara 0,98 sampai dengan 0,99. Semakin besar nilai presisi suatu alat maka semakin badus pula alat ukur tersebut.

Untuk resolusi dari alat ukur ini adalah 0,01, yang didapatkan dari alat ukur ini yang mempunyai perubahan terkecil 0,01. Hal ini disebabkan karena alat ukur ini merupakan alaut ukur tekanan digital dengan tampilan dia digit di belakang koma. Sedangkan nilai sensitivitasnya adalah 0,6 yang terlihat saat pertama kali terjadi perubahan tekanan, pada tampilan pertama kali tekanan yang terbaca adalah 0,6 Psi.

4.5.2 Analisa Plat Hitam

Semua hasil pengujian dan pengambilan data serta perhitungan data dipengaruhi oleh kualitas alat ukur ini. Dari analisa data diperoleh nilai *uncertainty* setiap pencuplikan data

kecil. Semakin kecil nilai *uncertainty* maka semakin bagus kualitas alat ukur tersebut. Kualitas dari alat ukur ini juga dipengaruhi oleh rangkaian dasarnya seperti sensor. Kerja sensor alat ukur ini dipengaruhi oleh penempatan plat hitam yang diletakkan diantara LDR dan led. Dari dua bentuk yang telah dicoba antara bentuk lurus dengan bentuk yang melengkung, lebih bagus bentuk melengkung dalam hal luasan. Luasan plat hitam berpengaruh dengan intensitas cahaya yang diterima oleh LDR. Selain itu ketebalan plat hitam juga berpengaruh, jika plat semakin tebal maka kemungkinan kebocoran cahaya led sedikit. Dari tiga jenis plat hitam yaitu piringan hitam, *casing* disket, dan *negative film* yang dipakai adalah piringan hitam. Alasan memakai piringan hitam karena diantara ketiga macam bahan tersebut piringan hitam sangat tipis dan tidak tembus cahaya dan ringan sehingga mudah terangkat seiring dengan mengembangnya *bourdon*.

4.5.3 Analisa Hubungan Penunjukan Analog dan Digital

Alat ukur ini mempunyai dua tampilan yaitu tampilan analog dan digital. Tampilan analog adalah *pressure gauge* yang sudah dimodifikasi dengan LDR dan led yang diletakkan di dalam *box*. Sedangkan untuk tampilan digital bisa dilihat dengan LCD. Dari sensor *pressure gauge* mempunyai keluaran dalam bentuk sinyal analog yaitu resistansi. Untuk masuk ke dalam rangkaian Adc diperlukan konversi resistansi ke tegangan dengan menggunakan rangkaian pembagi tegangan dengan range 0,13 – 1,70 Volt. Dari tengan yang masuk ke ADC dikonversikan ke bilangan biner dengan cara tes ADC. Dari tes ADC diketahui bahwa pada tegangan 0,13 terbaca ADC biner 22 dan tegangan 1,70 terbaca ADC 180. Dari data biner ADC dikirim ke mikrokontroler dan diolah dengan konversi tegangan ke tekanan dan kemudian ditampilkan ke LCD.

BAB V
PENUTUP

5.1 Kesimpulan

Beberapa kesimpulan yang dapat ditarik dari penelitian ini diantaranya :

- Sensor alaut ukur ini dibuat dari *pressure gauge* yang dimodifikasi dengan LDR dan led.
- Range pengukuran alat ukur adalah 0 – 15 Psi.
- *Uncertainty* pengukuran tekanan minimum adalah 0,00 ± 0,00 dan tekanan maksimum adalah 15,02 ± 0,05 untuk pengukuran naik dan *Uncertainty* pengukuran tekanan minimum adalah 0,00 ± 0,00 dan tekanan maksimum adalah 15,02 ± 0.
- Untuk pemasangan led dengan LDR harus sejajar dan led harus fokus sehingga cahaya yang jatuh tepat di LDR.
- Penempatan plat hitam serta ketebalan sangat berpengaruh dengan outputan sensor.

5.2 Saran

- Untuk membuat alat ukur tekanan dengan menggunakan sensor ini, pemasangan plat hitam dan tebal tipisnya serta kemampuan dapat meneruskan cahaya sangat berpengaruh dengan outputan yang nantinya akan langsung dimasukkan ke ADC tanpa menggunakan *signal conditioning.*
- Dalam perancangan dan pembuatan alat ukur ini bisa dikemas lebih kecil dan rapi.

DAFTAR PUSTAKA

[1] **Diktat Kerja Praktek Instrumen**, PT Petro Kimia Gresik Pabrik III Instrumen, 2003.

[2] http://www.google .com

[3] Modul Praktikum mikrokontroler, D3 Teknik Instrumentasi, Workshop Instrumentasi,2006.

[4] Nalwan, Paulus Andi.2003. **Panduan Teknik Antar Muka dan Pemrograman Mikrokontroler.**

[5] Musyafa',Ali, Ir., MSc. Modul **Kuliah Instrumentasi Industri RF 0591**. Jurusan Teknik Fisika. FTI-ITS.

Lampiran A

SOFTWARE PORTABLE POCKET ALAT UKUR PRESSURE BERBASIS MIKROKONTROLER

DESIGN BY : NATRIA IKE KUSDIANINGSIH

(2403 030 025)

```
$mod51

;konfigurasi adc
Data_adc        equ     60h
Rd_adc          bit     p3.6
Wr_adc          bit     p3.7

T_start         bit     p3.1
T_stop          bit     p3.2

Data_p          equ     61h
H_rb            equ     63h
H_r             equ     64h
H_p             equ     65h
H_s             equ     66h

                org     0h
                jmp     start

Start:  mov     R1,#03fh
        call    write_ins
        mov     R1,#0dh
        call    write_ins
        mov     R1,#06h
        call    write_ins
        mov     R1,#01h
        call    write_ins
```

```
        mov     R1,#0c0h
        call    write_ins
        jmp     M_start

Write_ins:
        mov     P0,R1
        clr     P2.0
        call    delcd
        setb    P2.1
        clr     P2.1
        call    delcd
        ret
Write_data:
        mov     P0,P1
        setb    P2.0
        call    delcd
        setb    P2.1
        clr     P2.1
        call    delcd
        ret

Delcd:   mov     20h,#0fh
Delcd1:  mov     19h,#0fh
         djnz    19h,$
         djnz    20h,delcd1
         ret
Delcdx:  mov     20h,#0f4h
Delcdy:  mov     19h,#0ffh
         djnz    19h,$
         djnz    20h,delcdy
         ret
Barisx:  mov     R4,#16
         mov     R1,#80h
```

```
        call        write_ins

Loop_show:
        jnb         T_start,M_mulai
        nop
        jmp         loop_show
M_mulai:
        jnb         T_start,$
        mov         dptr,#ukur_ne
        mov         R3,#1
        call        barisx
        nop
Loop_ukur:
        call        baca_adc
        call        konversi_adc
        call        konversi
        call        tampil_lcd
        jnb         T_stop,M_stop
        jmp         loop_ukur

M_stop:
        jnb         T_stop,$
        jmp         lagi_ukur


Baca_adc:
        mov         P1,#0ffh
        clr         wr_adc
        nop
        setb        wr_adc
        nop
        nop
        clr         rd_adc
```

MILIK PERPUSTAKAAN ITS

```
        call    write_ins
Tulisx: clr     a
        movc    a,@a+dptr
        mov     R1,a
        inc     dptr
        call    write_data
        call    delcdx
        djnz    R4,tulisx
Barisy: mov     R4,#16
        mov     R1,#0c0h
        call    write_ins
Tulisy: clr     a
        movc    a,@a+dptr
        mov     R1,a
        inc     dptr
        call    write_data
        call    delcdx
        djnz    R4,tulisy
        call    delcd
        djnz    R3,barisx
        ret


M_start:
        mov     dptr,#show_ne
        mov     R3,#2
        call    barisx

Lagi_ukur:
        mov     dptr,#lagi_ne
        mov     R3,#1
        call    barisx
        mov     R1,#0cfh
```

```
                djnz    R6,loop_kali_1

;((22 * d)/100 ) - ((435)/100)
;---------------------
                mov     R6,#12
Loop_50:        mov     R7,#29
Loop_10:        mov     a,h_s
                jz      zero_sat
                dec     h_s
                jmp     k_k
Zero_sat:       mov     a,h_p
                jz      zero_pul
                dec     h_p
                mov     h_s,#9
                jmp     k_k
Zero_pul:       mov     a,h_r
                jz      zero_rat
                dec     h_r
                mov     h_p,#9
                mov     h_s,#9
                jmp     k_k
Zero_rat:       mov     a,h_rb
                jz      zero_rib
                dec     h_rb
                mov     h_r,#9
                mov     h_p,#9
                mov     h_s,#9
                nop
K_k:            djnz    R7,loop_10
                djnz    R6,loop_50
Zero_rib:       nop
                ret
```

```
        mov             data_adc,P1
        setb            rd_adc
        ret
Zero_data:      nop
                ret

Konversi:       mov     h_rb,#0
                mov     h_r,#0
                mov     h_p,#0
                mov     h_s,#0
                mov     a,data_adc
                jz      zero_data
                mov     R6,data_adc
Loop_kali_1:
                mov     R7,#17
Loop_kali_2:    inc     h_s
                mov     a,h_s
                cjne    a,#10,not_10_k
                mov     h_s,#0
                inc     h_p
                mov     a,h_p
                cjne    a,#10,not_10_k
                mov     h_p,#0
                inc     h_r
                mov     a,h_r
                cjne    a,#10,not_10_k
                mov     h_r,#0
                inc     h_rb
                mov     a,h_rb
                cjne    a,#10,not_10_k
                mov     h_rb,#0
Not_10_k:       nop
                djnz    R7,loop_kali_2
```

```
        mov     R1,#0c9h
        call    write_ins
        mov     a,h_s
        movc    a,@a+dptr
        mov     R1,a
        call    write_data
        mov     R1,#0cfh
        call    write_ins
        ret

Delay10ms:
        mov     R5,#0
D10msb: mov     R6,#0
        djnz    R6,$
        djnz    R5,d10msb
        ret

Karakter:
        db      00110000b       ;0
        db      00110001b       ;1
        db      00110010b       ;2
        db      00110011b       ;3
        db      00110100b       ;4
        db      00110101b       ;5
        db      00110110b       ;6
        db      00110111b       ;7
        db      00111000b       ;8
        db      00111001b       ;9
        db      01000001b       ;a(10)
        db      01000010b       ;b(11)
        db      01000011b       ;c(12)
        db      01000100b       ;d(13)
        db      01000101b       ;*(14)
```

```
Konversi_adc:  clr    c
               mov    a,data_adc
               mov    b,#100
               div    ab
               mov    h_rb,a
               mov    a,b
               mov    b,#10
               div    ab
               mov    h_r,a
               mov    h_p,b
               ret
Tampil_lcd:    mov    dptr,#karakter
               mov    r1,#0c5h
               call   write_ins
               mov    a,h_rb
               movc   a,@a+dptr
               mov    R1,a
               call   write_data
;
               mov    r1,#0c6h
               call   write_ins
               mov    a,h_r
               movc   a,@a+dptr
               mov    R1,a
               call   write_data
;
               mov    r1,#0c8h
               call   write_ins
               mov    a,h_p
               movc   a,@a+dptr
               mov    R1,a
               call   write_data
```

```
        db      00100000b               ;#(15)

Show_ne:db      '◇<natria ike>◇'        ;1
        db      '◇-----◇-----◇ '
        db      '< alat ukur  >  '      ;2
        db      '◇-<tekanan>--◇'
Lagi_ne:
        db      '<press t.start.>     '  ;3
        db      '◇◇◇----◇◇◇'
Ukur_ne:
        db      '◇-<tekanan>--◇'        ;1
        db      '◇<[  ,  ]psi>    '

End
```

Features

- Compatible with MCS®-51 Products
- 4K Bytes of In-System Programmable (ISP) Flash Memory
  - Endurance: 1000 Write/Erase Cycles
- 4.0V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Watchdog Timer
- Dual Data Pointer
- Power-off Flag
- Fast Programming Time
- Flexible ISP Programming (Byte and Page Mode)
- Green (Pb/Halide-free) Packaging Option

8-bit Microcontroller with 4K Bytes In-System Programmable Flash

AT89S51

1. Description

The AT89S51 is a low-power, high-performance CMOS 8-bit microcontroller with 4K bytes of In-System Programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with In-System Programmable Flash on a monolithic chip, the Atmel AT89S51 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

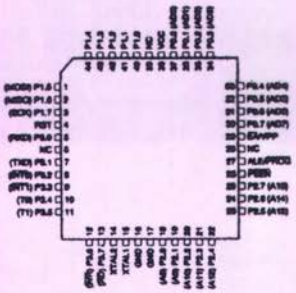
The AT89S51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, two 16-bit timer/counters, a five-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next external interrupt or hardware reset.

2487C–MICRO–03/06

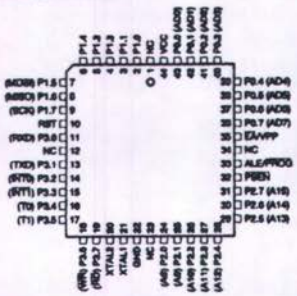
2. Pin Configurations

2.1 40-lead PDIP

2.2 44-lead TQFP

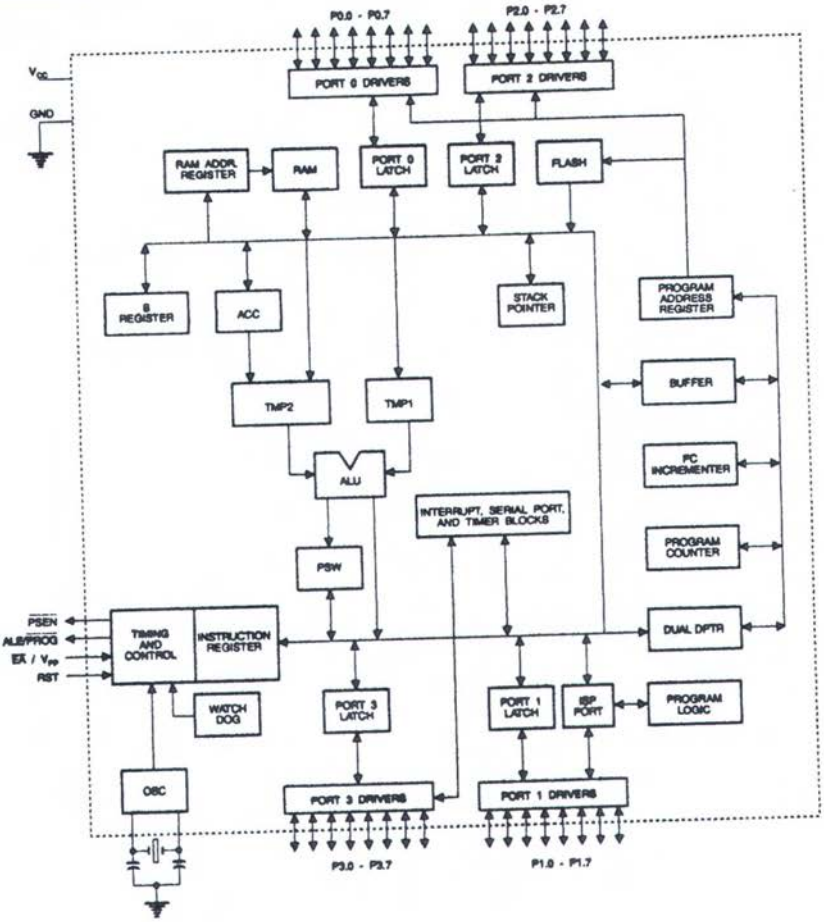


2.3 44-lead PLCC



2.4 42-lead PDIP

Block Diagram



4. Pin Description

4.1 VCC

Supply voltage (all packages except 42-PDIP).

4.2 GND

Ground (all packages except 42-PDIP; for 42-PDIP GND connects only the logic core and the embedded program memory).

4.3 VDD

Supply voltage for the 42-PDIP which connects only the logic core and the embedded program memory.

4.4 PWRVDD

Supply voltage for the 42-PDIP which connects only the I/O Pad Drivers. The application board MUST connect both VDD and PWRVDD to the board supply voltage.

4.5 PWRGND

Ground for the 42-PDIP which connects only the I/O Pad Drivers. PWRGND and GND are weakly connected through the common silicon substrate, but not through any metal link. The application board MUST connect both GND and PWRGND to the board ground.

4.6 Port 0

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

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

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. **External pull-ups are required during program verification.**

4.7 Port 1

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

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

Port Pin	Alternate Functions
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

4.8 Port 2

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

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

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

4.9 Port 3

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

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

Port 3 also serves the functions of various special features of the AT89S51, as shown in the following table.

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

4.10 RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 98 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

4.11 ALE/$\overline{PROG}$

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

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

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

4.12 $\overline{\text{PSEN}}$

Program Store Enable ($\overline{\text{PSEN}}$) is the read strobe to external program memory.

When the AT89S51 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twic each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to exte nal data memory.

4.13 $\overline{\text{EA}}$/VPP

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

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

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programmin

4.14 XTAL1

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

4.15 XTAL2

Output from the inverting oscillator amplifier

5. Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown i Table 5-1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be imple- mented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

LAMPIRAN C : SPESIFIKASI ALAT

A. Spesifikasi alat keseluruhan

- Range : 0 – 15 Psi
- Sensitivitas : 0,60 Psi
- Resolusi : 0,01 Psi
- Akurasi maksimum : 1 Psi

B. Sensor

- Pressure gauge range 0-38 Psi merek Migishita produk Jerman
- LDR (Light Dependent Resistor)
- Led superbright 16 Volt
- Resistor 470 Ω

C. Rangkaian Pembagi Tegangan

- Multitone 10^4 Ω

D. Power Supply

- Kapasitor 1000μF/25 Volt, 1000μF/16 Volt
- Regulator 7805
- Resistor 470 Ω

E. Minimum Sistem

- IC AT89S51
- Push Button
- Resistor 470 Ω, 1 kΩ
- Kapasitor keramik 30 pF, 10 μF
- Crystal 11,0982

F. Rangkaian ADC

- IC ADC 0804
- Resistor 10 kΩ
- Kapasitor 150 pF
- Multitone 10^3 Ω

G. Display

- LCD 2 x 16
- Kabel data
- Dioda IN4002

H. Casing/Box

- Box ukuran 17 x 10 cm
- Push on
- Tombol On/Off

I. Plant

- Compressor merek Unoair range 0-115 Psi
- Konektor pressure gauge 20 pF
- Konektor pressure 40 pF

LAMPIRAN D

Terima kasih kepada :

1. Keluarga besar di Nganjuk atas dukungan serta doa yang mengalir dengan tulus.
2. M'Yupi dan M'Niko, matur nuwun adhek ngrepoti terus.
3. Pak Fitri yang banyak membantu dalam menyelesaikan tugas akhir ini. Suwun guyonane.
4. Someone yang pernah ada, makasih sudah memberi warna baru dalam hari-hariku.
5. "Five Little Stars" {m'dD, Mba'Iez, M'Bunder, Ya2nK} awal cahaya hidupku di awal dunia bebasku.
6. My BoyFriends : M'Antok (Novan), Aan (Andreas), Restu (TuTu), ChoCho (Singgih), Smile (Ismail), Zai-Zai (arek Malang), Amen, Ku2n, Alvan, Listyo, Aris (Cepu), dan On Yanuar. Makasih banyak dan maafkan Ike.
7. My GirlFriends : Mba Tika (makasih banyak atas persaudaran ini), Imunk, Sita She~mBrut, Sari, Citra. Makasih banyak dan maafin Ike.
8. Tama (makasih candanya), Yaya (Rambutan), Widityo, Bina, Agnes, Dony, Rahmat, Arifai, Fahmi, Asterias, Elok, Andhin, ChiChi, Vani, Dotty (Dokter Robot), Indra, Micha, Vita, Ummi. Makasih banyak dan maafin Ike.
9. My sisters: Farit (PakDe), sidiq, Richo, Rahmat, Salam, Elak (Bantal), Ucil, Mustakim, Indra, Heri, Aden, Al Haq, Awang, Budi, Virgo, Dewi (Rong2), Lelita (Bona)~kapan jalan2 lagi, Yani, Linda, Nova, Sunyi, Esa, Lia, Ifa. Selamat berjuang! Jaga dan rawat WS ya! Makasih dan maafin Ike.
10. My Brothers : M'Munir (MuMun), M' Dido, M' Anwar, M' Pradanggapasti, mba' Rury, mba' Dina, mba' Ari. Makasih dukungaannya.
11. D9 Family : MeMe, mami Erick, Mama Lina, Tante Day, Mba' Yeni, Oke (Endel), mama Tari, mba' Dyah, Lilis, Harti, Descy. Makasih semuanya dan maafin Ike.

12. Pak Sam parkir makasih atas info2nya (jangan nggossip terus)
13. Seluruh karyawan Teknik Fisika yang telah memberi kelancaran selama kuliah.
14. semua orang yang ada dalam kehidupan Ike yang belum sempat tersebutkan, terima kasih banyak dan maafin Ike.

Terima kasih atas segala warna indah yang hadir dalam hari-hariku. Teruntuk WS q yang kusayangi serta seluruh warga WS, terima kasih yang sebesar-besarnya karena telah mengajarkan arti kehidupan, persaudaraan dan persahabatan, duka gembira bersama, dan yang terlebih mengajarkanku berpikir lebih dewasa , lebih berani menghadapi semuanya. Maafkan ike. Gomenasai.