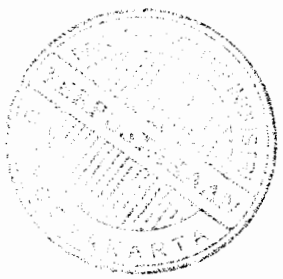
PENGENDALIAN SUHU BENDA KONDUKSI MENGGUNAKAN LOGIKA FUZZY BERBASIS MIKROKONTROLER

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat

untuk Memperoleh Gelar Sarjana Jurusan Teknik Elektro





Oleh :

Nama	**: Wahyu Setiawan**
No. Mahasiswa	**: 00 524 019**

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2007

LEMBAR PENGESAHAN PEMBIMBING

PENGENDALIAN SUHU BENDA KONDUKSI MENGGUNAKAN LOGIKA FUZZY BERBASIS MIKROKONTROLER

TUGAS AKHIR

Oleh :

Nama : Wahyu Setiawan

No. Mahasiswa : 00 524 019

Jogjakarta, 15 Februari 2007

Pembimbing I,	Pembimbing II,
Wahyudi Budi Pramono, ST	Medilla Kusriyanto, ST

LEMBAR PENGESAHAN PENGUJI

PENGENDALI SUHU BENDA KONDUKSI MENGGUNAKAN LOGIKA FUZZY BERBASIS MIKROKONTROLER

TUGAS AKHIR

Oleh :

Nama : Wahyu Setiawan

No. Mahasiswa : 00 524 019

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Jurusan Teknik Elektro
Fakultas Teknologi Industri Universitas Islam Indonesia

Jogjakarta, 2 Maret 2007

Tim Penguji

Wahyudi Budi Pramono, ST.
Ketua

Medilla Kusriyanto, ST.
Anggota I

Dwi Ana Ratnawati, ST.
Anggota II

Mengetahui,
Ketua Jurusan Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia

Tito Yuwono, ST.,M.Sc

HALAMAN PERSEMBAHAN

DAFTAR ISI

HALAMAN JUDUL i

LEMBAR PENGESAHAN PEMBIMBING ii

LEMBAR PENGESAHAN PENGUJI iii

HALAMAN PERSEMBAHAN iv

MOTTO v

KATA PENGANTAR vi

ABSTRAKSI ix

DAFTAR ISI x

DAFTAR GAMBAR xiii

DAFTAR TABEL xv

BAB I PENDAHULUAN 1

1.1. Latar Belakang 1

1.2. Rumusan Masalah 2

1.3. Batasan Masalah 2

1.4. Tujuan Penelitian 3

1.5. Manfaat Penelitian 3

1.6. Sistematika Penulisan 4

BAB II LANDASAN TEORI 6

2.1. Logika *Fuzzy* 6

2.1.1. Himpunan *fuzzy* 8

2.1.2. Fungsi keanggotaan 10

2.2. Mikrokontroler AT 89S51 11

DAFTAR GAMBAR

Halaman

Gambar 2.1 Diagram blok mikrokontroler..... 11

Gambar 2.2 Konfigurasi pin dari chip mikrokontroler..... 13

Gambar 2.3 Arsitektur perangkat keras AT89S51..... 19

Gambar 2.4 Peta memori data.....21

Gambar 2.5 Peta *Special Function Register* (SFR)..... 26

Gambar 2.6 Konfigurasi pin IC ADC ICL 7109..... 37

Gambar 2.7 TRIAC 40

Gambar 2.8 Daerah operasi TRIAC..... 40

Gambar 2.9 MOC 3021 41

Gambar 3.1 Diagram blok rangkaian 45

Gambar 3.2 Rangkaian sensor suhu.....46

Gambar 3.3 Rangkaian Pembagi Tegangan 48

Gambar 3.4 Rangkaian ADC 49

Gambar 3.5 Rangkaian Display.....51

Gambar 3.6 Gambar koneksi *keypad* ke mikrokontroler..... 52

Gambar 3.7 Rangkaian *TRIAC driver*..... 54

Gambar 3.8 Diagram alir fuzzifikasi..... 56

Gambar 3.9 Grafik derajat keanggotaan *error* hasil proses fuzzifikasi ... 58

Gambar 3.10 Diagram alir program kendali suhu 62

Gambar 4.1 Pengujian rangkaian sensor suhu..... 69

Gambar 4.2 Grafik setpoint pada 50°C..... 74

Gambar 4.3 Grafik setpoint pada 100°C 75

DAFTAR TABEL

Halaman

Tabel 2.1 Fungsi khusus port 3..... 15

Tabel 2.2 Nilai register setelah direset..... 17

Tabel 2.3 Tipe IC Mikrokontroler keluarga Atmel..... 33

Tabel 3.1 Hasil pengecekan aturan *fuzzy* pada pemanas menggunakan Grafik derajat keanggotaan *error* 60

Tabel 3.2 Hasil pengecekan aturan *fuzzy* pada kipas menggunakan Grafik derajat keanggotaan *error* 61

Tabel 4.1 Hasil pengukuran ADC pada rangkaian..... 71

Tabel 4.2 Hasil pengukuran pada setpoint 50°C..... 74

Tabel 4.3 Hasil pengukuran pada setpoint 100°C 75

MOTTO

"Hidup adalah serangkaian pengalaman; Setiap pengalaman membuat kita lebih besar, walaupun kita tidak pernah menyadarinya".

(*Henry Ford*)

"Tidak ada suatu musibahpun yang menimpa seseorang kecuali dengan izin Allah. Dan barang siapa yang beriman kepada Allah, niscaya Dia akan memberi petunjuk kepada hatinya. Dan Allah Maha mengetahui segala sesuatu".

(*At Taghaabun: 11*)

"Hidup adalah soal keberanian; menghadapi tanda tanya, tanpa kita bisa mengerti, tanpa kita bisa menawar. Terimalah dan hadapilah".

(*Soe Hoek-gie*)

"Tetapkanlah pikiranmu pada apa yang sudah menjadi cita-citamu, dan sanubarimu tidak akan salah menunjukkan jalan padamu untuk mencapainya".

(*Ekos*)

"Pernahkah kamu berjuang selama hidupmu ini?
Berjuang untuk suatu cita-citamu?
Hanya kau yang tahu, dan tak dapat kau tipu hatimu!"

(Edi-P)

KATA PENGANTAR

Assalamu'alaikum Wr.Wb.,

Pertama-tama penulis panjatkan puji syukur ke hadirat Allah SWT yang senantiasa melimpahkan rahmat dan karunia-Nya, sehingga penulis bisa menyelesaikan tugas akhir. Dan berkat hidayah-Nya pula penulis dapat menyusun tugas akhir dengan baik.

Dengan melaksanakan tugas akhir ini, penulis berharap dapat memberikan pengalaman serta menambah wawasan dalam bidang sistem kendali terutama dalam pembuatan pengendalian suhu yang mengimplementasikan logika *fuzzy* serta mikrokontroler didalamnya. Melalui tugas akhir ini pula, penulis dapat mengaplikasikan ilmu yang diperoleh dari bangku kuliah dengan kenyataan yang ada dilapangan.

Penulis mengucapkan terima kasih kepada Allah SWT, Nabi Muhammad SAW dan tidak lupa pula kepada semua pihak yang telah membantu dan mendukung dalam proses penyusunan laporan tugas akhir ini, yaitu :

1. Bapak Fachtul Wahid, ST., M.Sc., selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
2. Bapak Tito Yuwono, ST., M.Sc., selaku Kepala Jurusan Teknik Elektro Universitas Islam Indonesia.
3. Bapak Wahyudi Budi Pramono, ST., selaku dosen pembimbing I, yang telah membantu dan memberikan bimbingan kepada penulis dalam penyusunan laporan ini.

12. Temen-temen bulutangkis, Eko 'blangkon', Firman, Fael, Dwi', Arif Arsitek, Jecky, Anggoro, Pangky 'Suwito', Daus, Dimas, Cepot, Ebit yang sadar akan pentingnya olahraga. Terima kasih untuk dorongan semangat dari kalian.
13. Teman-teman Elektro'00, Ibnu, Dewi, Erlan, Adi, Ali, Bayu, Agung Magelang, Didit, Ruli, Arif Cirebon, Ghana, Asep, Adi Purnomo, Arif Lukman, Irwan, Rosi, Rofik, Arif Anwar, pokoknya semua anak Elektro angkatan 2000. Terima kasih telah menjadikan semuanya begitu bersahabat dan indah.

Menyadari akan banyak kekurangan yang ada karena keterbatasan pengetahuan maupun pengalaman dari penulis maka penulis mengharapkan kritik dan saran yang membangun dari pembaca. Akhirnya semoga laporan ini banyak berguna dan bermanfaat bagi kita semua.

Wassalamu'alaikum Wr.Wb.,

Yogyakarta, Maret 2007

Penulis

4. Bapak Medilla Kusriyanto, ST., selaku dosen pembimbing II, yang telah membantu serta memberikan arahan kepada penulis dalam penyusunan laporan ini.
5. Ibu Dwi Ana Ratnawati, ST., selaku dosen penguji , yang memberikan arahan dan masukan.
6. Ayahanda tercinta Bapak H. Bachrudin B.A dan Ibunda tercinta Ibu Hj. Sri Hastuti, serta Mba Ana, Mba Ade, Emah, Mas Iwan, Cici, keluarga besarku, terima kasih atas cinta, dorongan semangat yang tiada henti dan do'anya yang tulus.
7. My Mia, atas cinta, pengorbanan dan doa yang tulus. Semoga Allah SWT mewujudkan semua doa kita.
8. Adik-adikku Daus + Ririn , Delia, Opit + Indri, Wiwin, Chen-Chen, Tias, Pupunk, terimakasih atas perhatian, bantuan, dan kenangan yang indah.
9. Temen-temen dekatku, Arif Ainur 'tiger abang', Eko Joko S 'lare karanganyar', Amin Husni 'jejaka pekalongan', Ghana 'game maniac', Gombloh 'preman boyolali', Rabitah 'playboy cap duren', Lia 'Xepo girl' *thanks for all*... "*your make everything simple*".
10. Temen-temen Kost yang dulu, Anggi, Ijal, Okem, Amsal, pokoknya semua yang pernah tinggal bareng, makasih untuk persahabatan dan semua kenangan yang indah.
11. Temen-temen KKN angkatan 27 , Aan, Aang, Argha, Alan, Dita, Erlina, Lis, Rara, Ratna, terimakasih untuk kerjasama dan persahabatannya selama ini.

2.2.1. Konstruksi dasar Mikrokontroler AT89S51 18

2.2.2. RAM, ROM, dan Register dalam AT89S51 18

2.2.3. *Timer* pada AT89S51 27

2.2.4. Mode pengalamatan 27

2.2.5 . Dasar kerja program 31

2.3. Sensor Suhu Termokopel 34

2.4. *Analog to Digital Converter* 35

2.4.1. Ketelitian dan Resolusi ADC 37

2.5. *TRIAC* 39

2.6. Opto-oupler (MOC 3021) 40

2.7. IC TL 074 (*Operational Amplifier*) 42

2.8. Modul LCD (*Liquid Crystal Display*) 42

BAB III PERANCANGAN SISTEM ELEKTRONIS 45

3.1. Perancangan Perangkat Keras 45

3.1.1. Rangkaian sensor suhu 46

3.1.2. Rangkaian ADC ICL 7109 47

3.1.3. Rangkaian LCD 49

3.1.4. Rangkaian *keypad* 51

3.1.5. Rangkaian TRIAC *driver* 52

3.2.Perancangan Perangkat Lunak 54

3.2.1. Penerapan logika *fuzzy* pada sistem 55

3.2.1.1. Fuzzifikasi 56

3.2.1.2. Inferensi 58

3.2.1.3. Defuzzifikasi 59

3.2.2. Routine program pengambil data suhu 63

3.2.3. Routine pemberian instruksi dan penulisan data ke LCD 64

3.2.4. Routine program *keypad* 65

3.2.5. Routine kendali suhu 66

BAB IV ANALISIS SISTEM ELEKTRONIS 69

4.1. Pengujian Rangkaian Sensor 69

4.2. Pengujian Rangkaian ADC 70

4.3. Analisa Logika *Fuzzy* 71

4.4. Pengujian Alat Keseluruhan 72

BAB V PENUTUP 77

5.1. Kesimpulan 77

5.2. Saran 78

DAFTAR PUSTAKA xvi

LAMPIRAN

ABSTRAKSI

Pengendalian suhu banyak dimanfaatkan dalam berbagai bidang, baik untuk mempercepat suatu proses ataupun untuk memperlambat suatu proses. Oleh karenanya dibutuhkan suatu pengendali temperatur yang dapat beroperasi secara otomatis dan mampu mengendalikan suhu . Karakteristik suatu *plant* temperatur adalah tidak linier, sehingga pengendali konvensional (On/Off) tidak sesuai untuk sistem yang tidak linier.

Dalam tugas akhir ini akan dirancang sistem pengendali suhu benda konduksi yang berbasis logika *fuzzy* dengan pengendali mikrokontroler AT89S51. Sistem pengendalian ini dipilih karena pembuatan model yang lebih sederhana jika dibandingkan dengan sistem pengendali lainnya. Logika *fuzzy* akan menerapkan pengendalian yang tepat untuk memetakan suatu ruang *input* ke dalam suatu ruang *output*.

Pengaturan dengan menerapkan logika fuzzy yang memiliki 5 derajat keanggotaan Alat ini mengontrol rangkaian pemanas dan pendingin dengan mengendalikan sudut penyulutan TRIAC. Suhu pada benda diukur dengan termokopel, dan suhu setingan ditampilkan pada LCD. Hasil penelitian menunjukkan *setpoint* suhu tidak pernah tercapai tapi akan berselisih beberapa derajat celcius terhadap suhu yang diinginkan. Adanya osilasi sebesar ± 8°C. namun akan semakin berkurang pada suhu yang lebih tinggi Hal ini disebabkan adanya aturan-aturan *fuzzy* yang diterapkan mikrokontroler..

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Aplikasi logika *fuzzy* sudah mulai dirasakan pada beberapa bidang. Salah satu aplikasi terpentingnya adalah untuk membantu manusia dalam melakukan pengambilan keputusan. Aplikasi logika *fuzzy* untuk pengambilan keputusan ini semakin diperlukan tatkala semakin banyak kondisi yang menuntut adanya keputusan yang tidak hanya bisa dijawab dengan 'ya' atau 'tidak'.

Salah satu aplikasi dari logika *fuzzy* yaitu dipergunakan untuk pengendalian suhu pada benda konduksi yang berbasiskan mikrokontroler. Pengendalian suhu pada benda konduksi itu dimaksudkan untuk menstabilkan suhu benda tersebut pada suatu nilai suhu yang diinginkan, dengan demikian dibutuhkanlah suatu alat pemanas *(heater)* dan pendingin *(cooler)*, dimana pemanas bekerja pada saat suhu dibawah nilai dan pendingin pada kondisi sebaliknya.

Untuk dapat bekerjanya peralatan pemanas dan pendingin ini secara otomatis dibutuhkanlah suatu alat kontrol yang akan memeriksa suhu benda konduksi apakah berada dibawah atau diatas nilai, dan kemudian memerintahkan pemanas atau pendingin untuk bekerja. Sistem seperti ini mempunyai kelemahan yaitu untuk mencapai suhu yang diinginkan akan sulit karena pada saat dihidupkan peralatan pemanas dan pendingin langsung pada posisi maksimal sehingga pada saat peralatan dimatikan suhu benda konduksi telah melampaui nilai yang diinginkan, dan hal ini

dapat mengakibatkan peralatan pemanas dan pendingin pada saat mendekati nilai suhu yang diinginkan akan bekerja bergantian terus menerus.

Untuk mengatasi hal tersebut maka sistem kontrol dari peralatan pemanas dan pendingin harus dapat memerintahkan bekerjanya kedua alat tersebut tidak secara hidup/mati atau *on/off* akan tetapi kedua alat tersebut dapat bekerja (pada saat hidup) dengan posisi bekerja maksimal, dan kemudian turun terus hingga akhirnya mati pada saat nilai suhu yang diinginkan telah dicapai, dengan demikian pengendalian suhu dapat diatur secara lebih luas.

1.2 Rumusan Masalah

Dari ulasan mengenai masalah diatas, dapat ditarik rumusan masalah " bagaimana merancang pengendali suhu benda konduksi menggunakan logika fuzzy berbasis mikrokontroler."

1.3 Batasan Masalah

Untuk tidak meluasnya lingkup pembahasan masalah dalam memberikan keterangan dalam tugas akhir ini, penulis akan memberikan batasan-batasan masalah yang akan dibahas.

- Alat ini akan mengendalikan perubahan suhu .
- Sensor suhu yang digunakan adalah termokopel.

- Menggunakan logika fuzzy berbasis mikrokontroler Atmel 89S51 dengan tampilan peraga LCD untuk tampilan suhu riil.
- Pemanas yang dipakai adalah elemen pemanas dan pendingin menggunakan kipas angin, mengingat alat ini hanya sebagai simulasi.
- Pengaturan suhu panas benda dibatasi antara 26°C hingga 250°C

1.4 Tujuan Penelitian

Tujuan dari penulisan tugas akhir ini adalah merancang dan membuat suatu rangkaian pengendali suhu benda panas konduksi dengan menggunakan logika *fuzzy* berbasis Mikrokontroler Atmel 89S51. yang akan dipergunakan untuk mengendalikan suhu benda konduksi agar stabil pada suatu nilai suhu yang telah ditentukan.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Mengimplementasikan pengetahuan tentang mikrokontroler dan pemrogramannya terutama dari keluarga ATMEL 89S51.
2. Hasil dari tugas akhir ini dapat dimanfaatkan untuk keperluan sehari-hari.
3. Mengaplikasikan logika *fuzzy* untuk pengendalian suhu suatu benda konduksi dengan menggunakan mikrokontroler.

1.6 Sistematika Penulisan

BAB I PENDAHULUAN

Bagian ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan tugas akhir, manfaat tugas akhir, dan sistematika penulisan tugas akhir ini.

BAB II LANDASAN TEORI

Bagian ini memuat teori-teori yang berhubungan dengan tugas akhir ini.

BAB III PERANCANGAN SISTEM ELEKTRONIS

Bagian ini menjelaskan objek perancangan serta tahap-tahap perancangan yang dilakukan.

BAB IV ANALISIS SISTEM ELEKTRONIS

Bagian ini berupa uraian tentang analisis sebuah sistem elektronis yang telah terpasang ada. Penjelasan tentang cara kerja alat, keunggulan, metode-metode yang digunakan.

BAB V PENUTUP

Bagian ini berupa kesimpulan-kesimpulan dari proses perancangan, implementasi, dan terutama pada analisis kerja sistem elektronis.

Bagian ini juga berisi saran-saran yang perlu diperhatikan berdasar keterbatasan-keterbatasan yang ditemukan dan asumsi-asumsi yang dibuat selama melakukan Tugas Akhir.

BAB II

DASAR TEORI

2.1 Logika Fuzzy

Pada tahun 1965, Zadeh memodifikasi teori himpunan yaitu setiap anggotanya memiliki keanggotaan yang bernilai kontinu antara 0 sampai 1. Himpunan ini disebut dengan Himpunan *Fuzzy (Fuzzy Set)*.

Pada prinsipnya logika *fuzzy* mencoba menjawab keterbatasan-keterbatasan yang dimiliki oleh struktur logika biner boolean yang hanya memiliki dua kondisi pernyataan yaitu benar *(true)* atau salah *(false)*. Logika *fuzzy* mencoba menjembatani kondisi-kondisi yang tidak hanya bisa diselesaikan dengan pernyataan ya atau tidak. Dan logika *fuzzy* juga mencoba mendeskripsikan kondisi-kondisi pertengahan, kondisi antara situasi ya dan tidak ke dalam formulasi matematis.

Logika *fuzzy* dikatakan sebagai logika baru yang lama, sebab ilmu tentang logika *fuzzy* modern dan metode baru ditemukan beberapa tahun yang lalu, padahal sebenarnya konsep tentang logika *fuzzy* itu sendiri sudah ada sejak lama.

Ada beberapa alasan mengapa logika *fuzzy* banyak digunakan dalam mengambil keputusan, antara lain :

1. Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.
2. Sangat fleksibel.
3. Memiliki toleransi terhadap data-data yang tidak tepat.
4. Mampu memodelkan fungsi-fungsi nonlinier yang sangat kompleks.
5. Dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
7. Didasarkan pada bahasa alami.

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy*, yaitu:

1. Variabel *fuzzy*

Variabel *fuzzy* merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*. Contoh: umur, temperatur.

2. Himpunan *fuzzy*

Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*. Contoh Variabel temperatur

dapat mengakibatkan peralatan pemanas dan pendingin pada saat mendekati nilai suhu yang diinginkan akan bekerja bergantian terus menerus.

Untuk mengatasi hal tersebut maka sistem kontrol dari peralatan pemanas dan pendingin harus dapat memerintahkan bekerjanya kedua alat tersebut tidak secara hidup/mati atau *on/off* akan tetapi kedua alat tersebut dapat bekerja (pada saat hidup) dengan posisi bekerja maksimal, dan kemudian turun terus hingga akhirnya mati pada saat nilai suhu yang diinginkan telah dicapai, dengan demikian pengendalian suhu dapat diatur secara lebih luas.

1.2 Rumusan Masalah

Dari ulasan mengenai masalah diatas, dapat ditarik rumusan masalah " bagaimana merancang pengendali suhu benda konduksi menggunakan logika fuzzy berbasis mikrokontroler."

1.3 Batasan Masalah

Untuk tidak meluasnya lingkup pembahasan masalah dalam memberikan keterangan dalam tugas akhir ini, penulis akan memberikan batasan-batasan masalah yang akan dibahas.

- Alat ini akan mengendalikan perubahan suhu .
- Sensor suhu yang digunakan adalah termokopel.

- Menggunakan logika fuzzy berbasis mikrokontroler Atmel 89S51 dengan tampilan peraga LCD untuk tampilan suhu riil.
- Pemanas yang dipakai adalah elemen pemanas dan pendingin menggunakan kipas angin, mengingat alat ini hanya sebagai simulasi.
- Pengaturan suhu panas benda dibatasi antara 26°C hingga 250°C

1.4 Tujuan Penelitian

Tujuan dari penulisan tugas akhir ini adalah merancang dan membuat suatu rangkaian pengendali suhu benda panas konduksi dengan menggunakan logika *fuzzy* berbasis Mikrokontroler Atmel 89S51. yang akan dipergunakan untuk mengendalikan suhu benda konduksi agar stabil pada suatu nilai suhu yang telah ditentukan.

1.5 Manfaat Penelitian

Manafaat dari penelitian ini adalah:

1. Mengimplementasikan pengetahuan tentang mikrokontroler dan pemrogramannya terutama dari keluarga ATMEL 89S51.
2. Hasil dari tugas akhir ini dapat dimanfaatkan untuk keperluan sehari-hari.
3. Mengaplikasikan logika *fuzzy* untuk pengendalian suhu suatu benda konduksi dengan menggunakan mikrokontroler.

1.6 Sistematika Penulisan

BAB I PENDAHULUAN

Bagian ini berisi latar belakang masalah, rumusan masalah, batasan masalah, tujuan tugas akhir, manfaat tugas akhir, dan sistematika penulisan tugas akhir ini.

BAB II LANDASAN TEORI

Bagian ini memuat teori-teori yang berhubungan dengan tugas akhir ini.

BAB III PERANCANGAN SISTEM ELEKTRONIS

Bagian ini menjelaskan objek perancangan serta tahap-tahap perancangan yang dilakukan.

BAB IV ANALISIS SISTEM ELEKTRONIS

Bagian ini berupa uraian tentang analisis sebuah sistem elektronis yang telah terpasang ada. Penjelasan tentang cara kerja alat, keunggulan, metode-metode yang digunakan.

BAB V PENUTUP

Bagian ini berupa kesimpulan-kesimpulan dari proses perancangan, implementasi, dan terutama pada analisis kerja sistem elektronis.

Bagian ini juga berisi saran-saran yang perlu diperhatikan berdasar keterbatasan-keterbatasan yang ditemukan dan asumsi-asumsi yang dibuat selama melakukan Tugas Akhir.

BAB II

DASAR TEORI

2.1 Logika Fuzzy

Pada tahun 1965, Zadeh memodifikasi teori himpunan yaitu setiap anggotanya memiliki keanggotaan yang bernilai kontinu antara 0 sampai 1. Himpunan ini disebut dengan Himpunan *Fuzzy (Fuzzy Set)*.

Pada prinsipnya logika *fuzzy* mencoba menjawab keterbatasan-keterbatasan yang dimiliki oleh struktur logika biner boolean yang hanya memiliki dua kondisi pernyataan yaitu benar *(true)* atau salah *(false)*. Logika *fuzzy* mencoba menjembatani kondisi-kondisi yang tidak hanya bisa diselesaikan dengan pernyataan ya atau tidak. Dan logika *fuzzy* juga mencoba mendeskripsikan kondisi-kondisi pertengahan. kondisi antara situasi ya dan tidak ke dalam formulasi matematis.

Logika *fuzzy* dikatakan sebagai logika baru yang lama. sebab ilmu tentang logika *fuzzy* modern dan metode baru ditemukan beberapa tahun yang lalu, padahal sebenarnya konsep tentang logika *fuzzy* itu sendiri sudah ada sejak lama.

Ada beberapa alasan mengapa logika *fuzzy* banyak digunakan dalam mengambil keputusan, antara lain :

1. Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.
2. Sangat fleksibel.
3. Memiliki toleransi terhadap data-data yang tidak tepat.
4. Mampu memodelkan fungsi-fungsi nonlinier yang sangat kompleks.
5. Dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
7. Didasarkan pada bahasa alami.

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy*, yaitu:

1. Variabel *fuzzy*

Variabel *fuzzy* merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*. Contoh: umur, temperatur.

2. Himpunan *fuzzy*

Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*. Contoh Variabel temperatur

terbagi menjadi 5 himpunan *fuzzy* yaitu: dingin, sejuk, normal, hangat, dan panas.

3. Semesta pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya. Contah: Semesta pembicaraan untuk variabel temperatur: [0 40]

4. Domain

Domain himpunan *fuzzy* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif.

2.1.1 Himpunan *Fuzzy*

Pada himpunan tegas *(crisp)*, nilai kenggotaan suatu item x dalam suatu himpunan A, yang sering ditulis dengan $\mu_A[x]$, memiliki 2 kemungkinan, yaitu:

1. Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
2. Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Kalau pada himpunan crisp, nilai keanggotaan hanya ada dua kemungkinan, yaitu 0 atau 1. Pada himpunan fuzzy nilai kenggotaannya terletak pada rentang 0 sampai 1. Apabila x memiliki nilai keanggotaan fuzzy $\mu_A [x]=0$ berarti x tidak menjadi anggota himpunan A, demikian pula apabila x memiliki nilai keanggotaan fuzzy $\mu_A [x]=1$ berarti x menjadi anggota penuh pada himpunan A.

Terkadang kemiripan antara keanggotaan fuzzy dengan probabilitas menimbulkan kerancuan. Keduanya memiliki nilai pada interval [0,1], namun interpretasi nilainya sangat berbeda antara kedua kasus tersebut. Keanggotaan fuzzy memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang.

Himpunan fuzzy memiliki 2 atribut, yaitu:

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel.

2.1.2 Fungsi Keanggotaan

Fungsi keanggotaan *(membership function)* adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai kenggotaannya (disebut juga derajat keanggotaan) yang memiliki interval antara 0 dan 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi. Ada beberapa fungsi yang bisa digunakan:

a. Representasi Linear

Pada representasi linear, pemetaan input ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas.

Ada dua keadaan himpunan fuzzy yang linier. Pertama, kenaikan himpunan dimulai pada nilai dominan yang memiliki derajat keanggotaan nol [0] bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi.

b. Representasi Kurva Segitiga

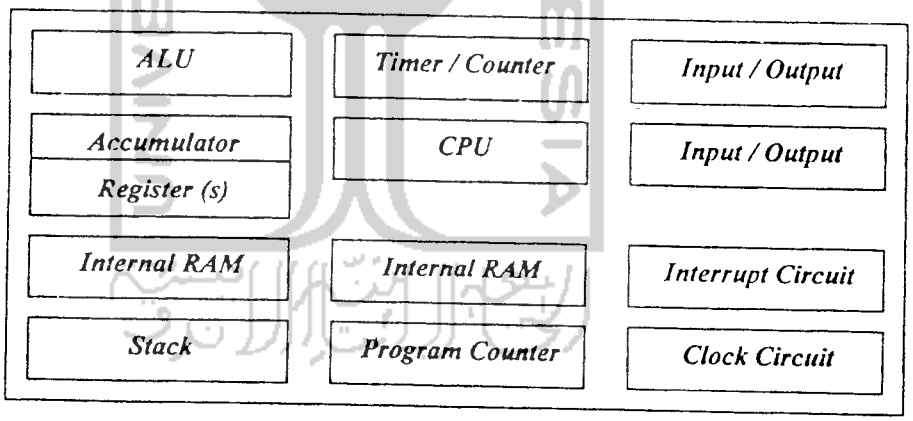
Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linier).

c. Representasi Kurva Trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1.

2.2 Mikrokontroler AT 89S51

Mikrokontroler adalah suatu rangkaian terintegrasi yang tersusun atas beberapa komponen, antara lain : *Central Processing Unit (CPU), Read Only Memory (ROM), Random Acces Memory (RAM), Timer* dan *Input/Output (I/O)* yang dikemas dalam satu keping tunggal (*Chip*). Jadi sebenarnya mikrokontroler merupakan sebuah piranti pengembangan mikroprosesor dengan teknik fabrikasi dan konsep pemrograman yang sama memungkinkan pembuatan mikroprosesor multiguna juga menghasilkan mikrokontroler. Pada gambar dibawah ini memperlihatkan diagram blok dari suatu rangkaian mikrokontroler secara umum.



Gambar 2.1. Diagram blok suatu mikrokontroler

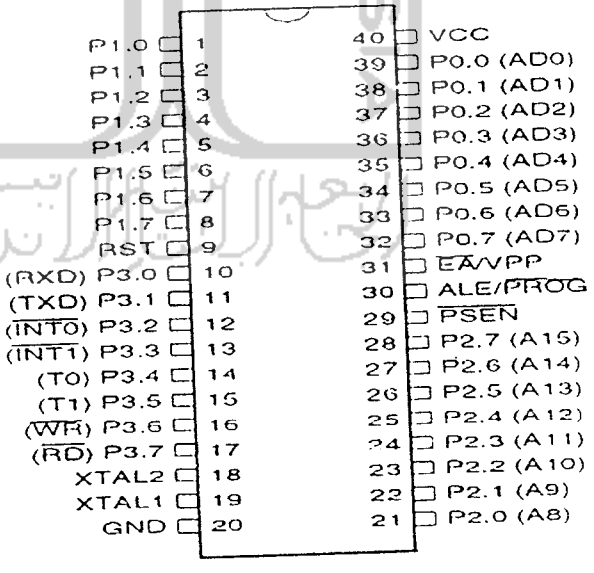
Mikrokontroler merupakan sistem mikroprosesor yang dirancang secara khusus untuk aplikasi dengan kendali sekuensial, yaitu digunakan untuk mengatur dan memonitor suatu sistem dengan urutan kerja tertentu. Gambar diatas memperlihatkan blok diagram mikrokontroler lengkap dengan komponen-

komponennya dalam satu *chip*. Dari gambar tersebut terlihat bahwa mikrokontroler merupakan gabungan antara mikroprosesor dengan *Read Only Memory (ROM), Random Acces Memory (RAM), Paralel I/O, Serial I/O, Counter, Timer* dan pembangkit isyarat pulsa detak. Sedangkan suatu mikroprosesor sendiri atau CPU terdiri atas beberapa komponen, antara lain : *Aritmathic Logic Unit (ALU), Program Counter (PC), Stack Pointer (SP)* dan register-register.

Dalam sejarahnya mikrokontroler MCS-51 merupakan jenis Mikrokontroler yang termasuk tua, keluarga Mikrokontroler MCS-51 adalah merupakan Mikrokontroler yang paling populer saat ini. Keluarga ini diawali oleh Intel yang mengenalkan IC Mikrokontroler type 8051 pada awal tahun 1980 an. Sampai saat ini sudah ada lebih 100 macam Mikrokontroler turunan 8051, sehingga terbentuklah keluarga besar Mikrokontroler dan biasa disebut MCS-51. Belakangan ini, pabrik IC Atmel ikut menambah anggota keluarga MCS-51. Produksi Mikrokontroler MCS-51 Atmel dibagi dua macam, yang pertama yaitu Mikrokontroler dengan jumlah pin 40 setara dengan jumlah pin 8051 yang asli, dan yang kedua adalah Mikrokontroler dengan jumlah pin 20 yang merupakan Mikrokontroler MCS-51 yang disederhanakan. Perbedaan keduanya adalah dalam hal kapasitas *Flash* PEROM (*Programmable and Erasable Read Only Memory*) nya. AT89C51/S51 mempunyai *Flash* PEROM dengan kapasitas 4 Kbyte, AT89C52/S52 dengan kapasitas 8 Kbyte, AT89C53 dengan kapasitas 12 Kbyte, AT89C55 dengan kapasitas 20 Kbyte dan AT 89C8252 berisikan 8 Kbyte *Flash* PEROM dan 2 Kbyte EEPROM (*Electrical Erasable and Programable ROM*). Penyederhanaan dilakukan pula pada Mikrokontroler ukuran kecil ini, yaitu

dengan cara mengurangi jalur I/O paralel, kemampuan yang lain sama sekali tidak mengalami pengurangan, penyederhanaan ini dimaksudkan untuk membentuk Mikrokontroler yang ukurannya fisiknya kecil akan tetapi dengan kemampuan yang sama.

AT89S51 adalah Mikrokontroler 8 bit keluaran Atmel dengan 4 Kbyte *Flash* PEROM yang merupakan memori dengan teknologi *high density nonvolatile memory* dan kompatibel dengan Mikrokontroler standar industri MCS-51, isi memori tersebut dapat diisi ulang atau dihapus berkali-kali sampai batas 1000 kali, Mikrokontroler ini merupakan *high performance* teknologi CMOS (*Complementary Metal Oxide Semiconductor*) dan dikemas dalam paket 40 pin dengan catu daya tunggal. Diagram susunan kaki dan simbol logika dari Mikrokontroler AT89S51 dalam bentuk PDIP (*Plastic Dual In Line Package*) dapat dilihat pada gambar berikut ini.



Gambar 2.2. Konfigurasi pin dari Chip Mikrokontroler AT89S51.

Masing-masing pin pada gambar tersebut memiliki fungsi tersendiri. Satu kumpulan pin memiliki fungsi sama dan diwakili oleh sebuah register atau alamat tersendiri pada internal CPUnya disebut juga port. Fungsi dari pin-pin tersebut adalah sebagai berikut :

1. Port 0

Port 0 dapat berfungsi sebagai I/O biasa, *low order multiplex address/* data atau menerima kode byte pada saat *Flash Programing*. Pada fungsi sebagai I/O biasa port ini dapat memberikan *output sink* ke delapan buah TTL input atau dapat diubah sebagai input dengan memberikan logika 1 pada port tersebut. Pada fungsi sebagai *low order multiplex address/* data port ini akan mempunyai *internal pull up*. Pada saat *Flash Programing* diperlukan *external pull up* terutama saat verifikasi program. Port 0 terdapat pada pin no. 32-39.

2. Port 1

Port 1 berfungsi sebagai I/O biasa atau menerima *low order address bytes* pada saat *Flash Programing*. Port ini mempunyai *internal pull up* dan berfungsi sebagai input dengan memberikan logika 1. Sebagai output, port ini dapat memberikan *output sink* ke empat buah input TTL. Port 1 terdapat pada pin no. 1-8.

3. Port 2

Port 2 berfungsi sebagai I/O biasa atau menerima *high order address bytes* pada saat mengakses memori secara 16 bit (Movx @DPTR). Pada saat mengakses memori secara bit. (Mov @Rn) port ini akan mengeluarkan isi dari P2 *Special Function Register (SFR)*. Port ini mempunyai *internal pull up*

dan berfungsi sebagai input dengan memberikan logika 1. Sebagai output, port ini dapat memberikan *output sink* ke empat buah input TTL. Port 1 terdapat pada pin no. 21-28.

4. Port 3

Port 3 yang terdapat pada pin 10-17 berfungsi sebagai *input/output* (I/O) yang mempunyai sifat sama dengan port 1 maupun port 2, sedangkan sebagai fungsi spesial, port-port ini mempunyai keterangan yang ditunjukkan pada tabel berikut ini :

Tabel 2.1. Fungsi Khusus Port 3

PORT PIN	NO PIN	FUNGSI KHUSUS
P3.0	10	RXD (masukan data port serial)
P3.1	11	TXD (keluaran data port serial)
P3.2	12	INT0' (masukan interupsi 0 dari luar)
P3.3	13	INT1' (masukan interupsi 1 dari luar)
P3.4	14	T0 (masukan ke pencacah 0)
P3.5	15	T1 (masukan ke pencacah 1)
P3.6	16	WR' (sinyal baca untuk memori luar)
P3.7	17	RD' (sinyal tulis untuk memori luar)

5. PSEN (*Program Strobe Enable*)

PSEN adalah kontrol sinyal yang mengijinkan untuk mengakses program (code) memori *eksternal*. Pin ini dihubungkan ke pin OE (*output enable*) dari EPROM. Sinyal PSEN akan 0 pada tahap *fetch* (penjemputan) instruksi. PSEN akan selalu bernilai 0 pada pembacaan program memori *internal*. PSEN terdapat pada pin 29.

6. ALE (*Address Latch Enable*)

Pin ini dapat berfungsi sebagai *Address Latch Enable* (ALE) yang me –*latch low byte address* pada saat mengakses memori *eksternal.* Sedangkan pada saat *Flash Programing* berfungsi sebagai *pulse input,* pada operasi normal ALE akan mengeluarkan sinyal clock sebesar 1/16 frekuensi oscilator kecuali pada saat mengakses memori *eksternal*, sinyal clock pada pin ini dapat pula di *disable*

dengan men-set bit 0 dari SFR dialamat 8Eh. ALE hanya akan aktif pada saat mengakses memory *eksternal* (Movx dan Movc). ALE terdapat pada pin 30.

7. EA (*External Access*)

Pada kondisi logika rendah, pin ini akan berfungsi sebagai EA yaitu Mikrokontroler akan menjalankan program yang ada pada memori *eksternal* setelah sistem direset. Jika berkondisi logika tinggi, pin ini akan berfungsi untuk menjalankan program yang ada pada memori *internal*. Pada saat *Flash Programing* pin ini akan mendapat tegangan 12 Volt (VP), EA terdapat pada pin 31.

8. *On-Chip Oscillator*

AT89S51 telah memiliki *on-chip oscillator* yang dapat bekerja dengan menggunakan kristal *eksternal* yang dihubungkan ke pin XTAL1 dan XTAL2. Tambahan kapasitor untuk menstabilkan oscilator tersebut. Nilai kristal yang biasa dipakai oleh keluarga MCS-51 adalah 12 MHz. O*n-chip oscillator* juga dapat menggunakan isyarat pulsa detak dari luar, misalnya AFG (*eksternal oscilator*) yang cukup dihubungkan pada pin XTAL1.

9. RST (*Reset*)

RST pada pin 9 merupakan reset dari AT89S51. Jika pada pin ini diberi masukan logika tinggi selama 2 *machine cycle* maka register-register *internal* pada AT89S51 akan berisi nilai *default* setelah sistem di reset seperti ditunjukkan pada tabel berikut ini :

Tabel 2.2. Nilai Register Setelah Direset

REGISTER	ISI
Program counter	0000H
Accumulator	00H
B register	00H
PSW	00H
SP	07H
DPTR	0000H
Port 0-3	FFH
IP (8031/8051)	XXX00000B
IP (8032/8052)	XX000000B
IE (8031/8051)	0XX00000B
IE (8032/8052)	0X000000B
Timer Register	00H
SCON	00H
SBUF	00H
PCON (HMOS)	0XXXXXXXB
PCON (CMOS)	0XXX0000B

10. Koneksi Catu Daya

Beroperasi pada tegangan 5 Volt. Pin Vcc terdapat pada pin 40 sedangkan Vss (*ground*) terdapat pada pin 20.

Untuk merancang suatu sistem yang menggunakan Mikrokontroler AT89S51 sebagai basis utamanya, diperlukan pemahaman terhadap konstruksi, instruksi dan pendukung operasi yang dimiliki oleh Mikrokontroler tersebut. Berikut ini adalah sekilas gambaran dari Mikrokontroler AT89S51 :

a. Konstruksi dasar Mikrokontroler AT89S51

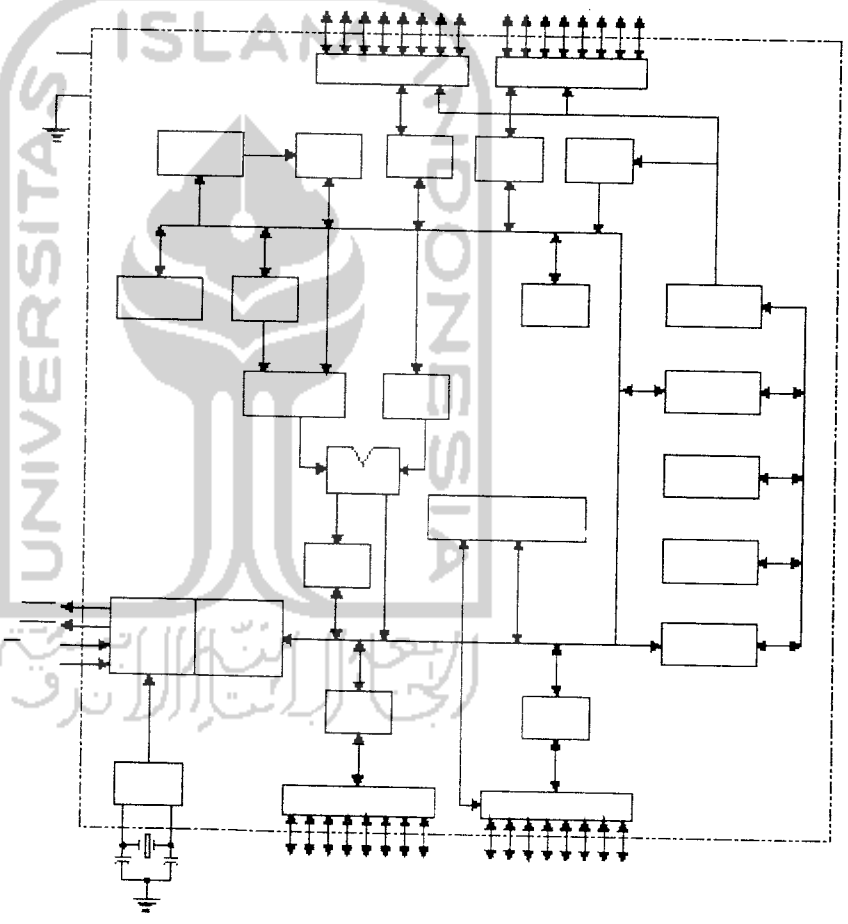
Mikrokontroler ini dilengkapi dengan berbagai fasilitas diantaranya adalah :

1. *Central Processing Unit* (CPU)
2. Memori data (RAM) didalam chip 128 byte, *Flash* PEROM didalam chip sebesar 4 Kbyte yang dapat diisi ulang sebanyak 1000 kali.
3. Pengendali interupsi (*Interrupt Control*)
4. Bus Kendali (*Bus Control*)
5. UART (*Universal Asynchronous Receiver/Transmitter*) yang digunakan untuk komunikasi data secara serial (Jalur seri berada pada pin RXD dan TXD)
6. Rangkaian oscilator didalam chip dengan frekuensi maksimal 24 MHz.
7. Empat buah port yang masing-masing berisi 8 bit, sifatnya 2 arah (I/O) dan setiap bitnya dapat dialamati, salah satu portnya yaitu port 3 (P3) juga dapat berfungsi untuk komunikasi data secara serial. Interupsi, masukan untuk pencacah dan masukan isyarat perintah baca dan tulis (R/W).
8. Pewaktu (*Timer*)/ pencacah (*Counter*) 16 bit sebanyak 2 buah.

Dari uraian tersebut, letak dan konstruksi dari komponen-komponen yang telah disebutkan dapat dilihat secara jelas seperti pada gambar 2.3.

b. RAM, ROM dan register dalam AT89S51

Dalam pengertian MCS-51, *Flash* PEROM merupakan memori penampung program pengendali AT89C51, dikenal sebagai memori program (nomor $0000 sampai $07FF, lihat gambar 4). Sedangkan RAM dalam chip AT89S51 adalah memori data (dialamati dari alamat $00 sampai $FF), yaitu memori yang dipakai untuk menyimpan data.



Gambar 2.3. Arsitektur Perangkat Keras AT89S51

Memori data pada AT89S51 dibagi menjadi 2 bagian yaitu :

1). Memori alamat $00 sampai $7F merupakan memori seperti layaknya RAM yang dipakai sebagai memori penyimpan data biasa, terdiri atas :

a). Register serbaguna (*General Purpose Register*) terdapat pada memori alamat $00 sampai $18, memory sebanyak 32 byte ini dikelompokkan menjadi 4 kelompok register (*Register Bank*), 8 byte memory dari masing-masing kelompok itu dikenali sebagai R0, R1, R2, R3, R4, R5, R6 dan R7.

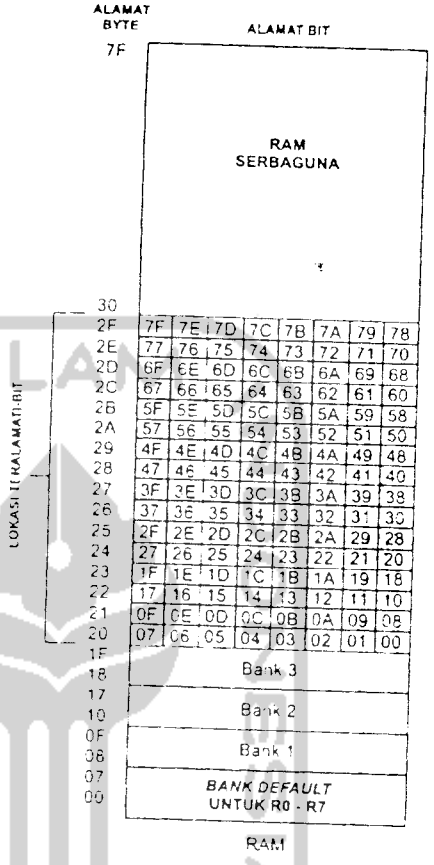
b). Memori level bit (memori alamat $20 sampai $2F), setiap byte memori didaerah ini bisa dipakai untuk menampung bit informasi yang masing-masing dialamati sendiri, dengan demikian dari 16 byte memori yang ada dapat dipakai untuk menyimpan 128 bit (16 X 8 bit) yang dialamati dengan bit alamat $00 sampai $7F.

c). Memori alamat $30 sampai $7F (sebanyak 80 byte) merupakan memori data biasa, dapat dipakai untuk menyimpan data maupun dipakai sebagai stack.

2. Memori alamat $80 sampai $FF dipakai sangat khusus yang dinamakan *Special Function Register* (SFR).

Pada AT89S51 terdapat register-register baku seperti yang dapat kita jumpai pada semua jenis Mikrokontroler ataupun mikroprosesor seperti *Program Counter* (PC), *Accumulator*, *Stack Pointer* (SP) *Register* dan Program Status Register. Juga terdapat register-register khas yang hanya terdapat pada keluarga

MCS-51, adapun register-register dan fungsi dari register dalam AT89S51 tersebut adalah sebagai berikut :



Gambar 2.4. Peta Memori Data

1) *Program Counter* (PC)

Register yang ditempatkan di tempat tersendiri didalam inti prosesor. PC mempunyai kapasitas 16 bit, didalam PC dicatat nomor memori program yang menyimpan instruksi berikutnya yang akan diambil (*fetch*) sebagai instruksi untuk dikerjakan.

2) Akumulator

Sesuai dengan namanya, akumulator adalah sebuah register yang berfungsi untuk menampung (*Accumulate*) hasil-hasil pengolahan data dari banyak instruksi MCS-51, register ini berada pada alamat E0h, akumulator dapat menampung data 8 bit (1 byte) dan merupakan register yang paling banyak kegunaannya, lebih dari setengah instruksi-instruksi MCS-51 melibatkan akumulator.

3) *Stack Pointer Register(SP Register)*

Merupakan salah satu bagian dari memori data yang dipakai sebagai stack (Tumpukan), yaitu tempat yang dipakai untuk menyimpan sementara nilai PC sebelum prosesor menjalankan subrutin, nilai tersebut akan diambil kembali dari stack dan dikembalikan ke PC saat prosesor selesai menjalankan subrutin. Stack Pointer adalah register yang yang berada pada alamat 81h dan berfungsi untuk mengatur kerja stack, dalam SP disimpan alarnat memori data yang dipakai untuk operasi stack berikutnya.

4) *Program Status Word* (PSW)

Program Status Word terletak pada alamat D0h dan berfungsi untuk mencatat kondisi prosesor setelah melaksanakan instruksi.

5) Register B

Merupakan register dengan kapasitas 8 bit, sebagai register pembantu akumulator saat menjalankan instruksi perkalian dan pembagian, register ini dapat dialamati secara bit.

6) DPH dan DPL

Data Pointer High Byte (DPH) dan *Data Pointer Low Byte* (DPL) masing-masing merupakan register dengan kapasitas 8 bit, tetapi dalam pemakaiannya kedua register ini digabungkan menjadi satu register 16 bit yang dinamakan sebagai *Data Pointer Register* (DPTR). Sesuai dengan namanya, Register ini dipakai untuk mengalamati data dalam jangkauan yang luas.

7) P0, PI, P2 dan P3

Seluruh port dapat diakses dengan pengalamatan secara bit, merupakan sarana 1/0 dua arah, selain dipakai sebagai port 1/0, PO dan P2 dapat pula dipakai untuk saluran data (*Data Bus*) dan saluran alamat (*Address Bus*) yang diperlukan AT89C51 untuk dapat menambah memori diluar chip.

8) SBUF (*Serial Buffer*)

Register Serial Buffer dipergunakan untuk mengirim data dan menerima data dengan UART yang terdapat dalam IC AT89S51. data yang disimpan di dalam SBUF akan dikirim keluar secara serial lewat kaki TX1), sebaliknya data seri yang diterima di kaki RXD dapat diambil di register SBUF, jadi SBUF akan berfungsi sebagai port *output* pada saat register ini diisi data dan SBUF akan menjadi port *input* apabila isinya diambil.

9) SCON (*Serial Control*)

Register SCON dipergunakan untuk mengatur mode operasi UART didalam IC AT89S51. hal-hal yang diatur meliputi penentuan kecepatan pengiriman data serial (*baud rate*), mengaktifkan fasilitas penerimaan data serial (fasilitas

ini tidak perlu diatur), register SCON digunakan pula untuk memantau proses pengiriman dan penerimaan data serial. Register ini dapat dialamati secara bit.

10) TL0/ TH0 (Timer 0 Low/High)

Kedua register ini bersama membentuk timer 0, yang merupakan pencacah naik (*up counter*), mode operasi kedua register ini diatur oleh register TMOD dan register TCON. Hal-hal yang dapat diatur antara lain adalah sumber detak untuk pencacah, nilai awal pencacah bilamana proses pencacahan mulai atau berhenti dan lain sebagainya.

11) TL 1 / TH 1 (*Timer 1 Low/High*)

Kedua register ini bersama membentuk *timer* 1, yang merupakan pencacah naik (*up counter*), mode operasi kedua register ini diatur oleh register TMOD dan register TCON. Hal-hal yang dapat diatur antara lain adalah sumber detak untuk pencacah, nilai awal pencacah bilamana proses pencacahan mulai atau berhenti dan lain sebagainya.

12) TMOD (*Timer Mode*)

Register TMOD dipergunakan untuk mengatur mode operasi *timer* 0 dan *timer* 1, lewat register ini masing-masing timer dapat diatur menjadi *timer* 16 bit, *timer* 13 bit, *timer* 8 bit yang dapat diisi ulang secara otomatis, atau 2 buah *timer* 8 bit yang terpisah. selain itu dapat diatur pula agar proses-proses pencacahan *timer* dapat dikendalikan lewat sinyal dari luar IC AT89S51 atau *timer* dipakai untuk mencacah isyarat-isyarat dari luar IC.

13) TCON (*Timer Control*)

Register TCON dipergunakan untuk memulai atau menghentikan proses pencacahan *timer* dan dipakai untuk memantau apakah terjadi limpahan dalam proses pencacahan. Selain itu masih tersisa 4 bit yang lain dalam register TCON yang tidak digunakan untuk mengatur *timer*, melainkan digunakan untuk mengatur isyarat interupsi yang diterima di pin INT0 atau INT1 dan digunakan untuk memantau apakah ada permintaan interupsi pada kedua pin tersebut. Register ini dapat dialamati secara bit.

14) IE (*Interrupt Enable*)

Register ini dipergunakan untuk mengaktifkan atau me-non aktifkan sarana interupsi, bit 0 sampai bit 6 dari register IE (IE.0, IE.1, ..., IE.6) digunakan untuk mengatur masing-masing sumber interupsi (sesungguhnya IE.6 tidak dipergunakan) sedangkan IE.7 digunakan untuk mengatur sistem interupsi secara keseluruhan, jika IE.7 = 0 mengakibatkan sistem interupsi non aktif tidak mempedulikan IE.0, IE.1, ..., IE6. Register ini dapat dialamati dengan cara pengalamatan bit.

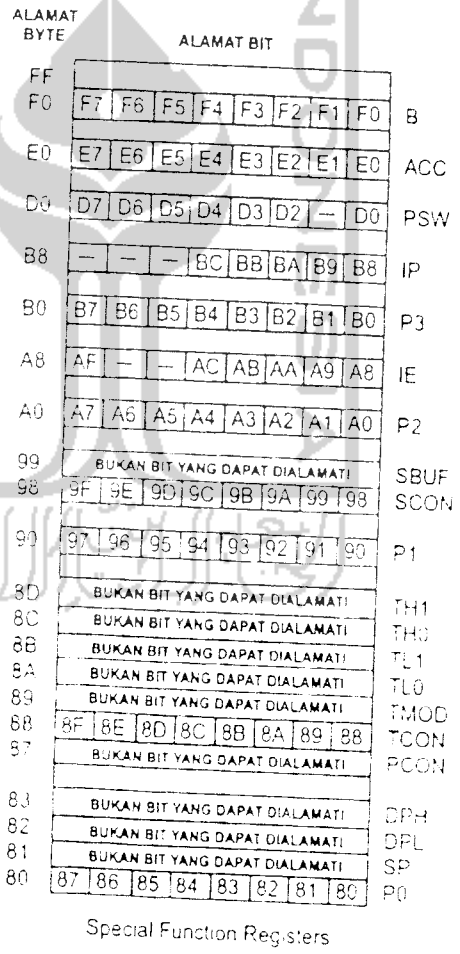
15) IP (*Interrupt Priority*)

Register ini dipergunakan untuk mengatur prioritas dari masing-masing sumber interupsi, masing-masing sumber interupsi dapat diberi prioritas tinggi dengan memberikan logika '1' pada bit bersangkutan dalam register ini. Sumber interupsi yang prioritasnya tinggi dapat menginterupsi proses dari sumber interupsi yang prioritasnya lebih rendah. Register ini dapat dialamati dengan cara pengalamatan bit.

16) PCON (*Power Control*)

Register PCON dipergunakan untuk mengatur pemakaian daya IC AT89S51, dengan cara 'menidurkan' IC tersebut sehingga hanya memerlukan arus kerja yang sangat kecil. Salah satu bit dalam register ini digunakan untuk menggandakan kecepatan pengiriman data seri (*baud rate*) dari UART didalam IC AT89S51.

Adapun letak register-register tersebut pada *Special Function Register* (SFR) dapat kita lihat pada gambar 2.5 berikut ini :



Gambar 2.5. Peta *Special Function Register* (SFR)

c. *Timer* Pada AT89S51

Mikrokontroler AT89S51 dibuat dengan dibekali 2 buah *timer*, keduanya dapat di kendalikan, diset, dibaca dan dikonfigurasi sendiri-sendiri. *Timer* AT89S51 memiliki 3 fungsi umum, yaitu :

1) Menghitung waktu antara 2 kejadian (*event*)

2) Menghitung jumlah kejadian itu sendiri

3) Membangkitkan *baud rate* untuk port serial.

Sebuah *timer* bekerja dengan mencacah, tidak tergantung pada fungsi sebagai register, *counter* ataupun *generator baud rate*, sebuah *timer* akan selalu membentuk tundaan waktu dengan melakukan operasi penambahan satu (*Increment*) secara terus menerus.

d. Mode Pengalamatan

Program pengendali mikrokontroler disusun dari kumpulan instruksi. Instruksi-instruksi tersebut setara dengan kalimat perintah bahasa manusia yang hanya terdiri atas predikat dan obyek. Dengan demikian tahap pertama pembuatan program pengendali mikrokontroler dimulai dengan pengenalan dan pemahaman predikat (Kata kerja) dan obyek apa saja yang dimiliki oleh mikrokontroler.

Obyek dalam pemrograman mikrokontroler adalah data yang tersimpan didalam memori, register atau alat *Input/ Output*. Sedangkan 'kata kerja' yang dikenal, secara umum dikelompokkan menjadi perintah untuk perpindahan data, operasi aritmatika, operasi logika, pengaturan alur program dan beberapa hal khusus. Kombinasi 'kata kerja' dan 'obyek' inilah yang membentuk perintah pengatur kerja mikrokontroler.

Instruksi Mov A, 7 Fh merupakan contoh sebuah instruksi dasar yang sangat spesifik, Mov merupakan 'kata kerja' yang memerintahkan untuk memindahkan data, merupakan predikat dalam kalimat perintah ini. Sedangkan obyeknya adalah data yang dipindahkan, dalam hal ini adalah data yang berada didalam memori nomor 7Fh dipindahkan ke akumulator A.

Data dapat berada di berbagai tempat yang berlainan, dengan demikian dikenal beberapa cara untuk mengalamati atau menyebut alamat suatu data (dalam bahasa Inggris disebut sebagai *Addressing Mode*) yang setiap, mode pengalamatan memberikan fleksibilitas khusus yang sangat penting, mode pengalamatan ini macamnya antara lain adalah :

1) Pengalamatan Data Konstan (*Immediate Addressing Mode*)

Contoh instruksi: Mov A,#20h.

Data konstan merupakan data yang berada didalam instruksi, contoh instruksi tersebut mempunyai makna data konstan #20h (sebagai data konstan ditandai dengan '#') dipindahkan ke akumulator A. Nilai dari suatu konstanta tersebut dapat segera menyatu dengan *op code* dalam memori program.

2) Pengalamatan Langsung (*Direct Addressing Mode*)

Contoh instruksi : Mov A, 30h

Cara tersebut digunakan untuk menunjuk data yang berada di dalam, memori dengan cara menyebut alamat memori tempat data tersebut berada. Contoh instruksi diatas mempunyai makna data yang berada di dalam alamat memori 30h di pindahkan ke akumulator A. sekilas instruksi tersebut sama dengan instruksi data konstan diatas, perbedaannya instruksi diatas menggunakan

tanda # yang menandai 20h adalah data konstan, sedangkan dalam, instruksi ini karena tidak ada tanda #, maka 30h adalah alamat dari memori. Dalam, pengalamatan ini, *operand-operand* ditentukan berdasar alamat 8 bit (1 *byte*) dalam suatu instruksi. Hanya RAM data internal dan SFR saja yang dapat diakses secara langsung.

3) Pengalamatan Tak Langsung (*Indirect Addressing Mode*)

Contoh instruksi : Mov A, @R0

Cara tersebut digunakan untuk menunjuk data yang berada di dalam alamat memori, apabila memori penyimpan data ini letaknya berubah-ubah sehingga alamat memori tidak disebut secara langsung tetapi di'titip'-kan ke register lain, dalam instruksi ini register serbaguna R0 digunakan untuk mencatat alamat memori, sehingga instruksi ini mempunyai makna memori yang alamatnya tercatat dalam R0 isinya dipindahkan ke akumulator A. tanda @ dipakai untuk menandai alamat memori disimpan di dalam R0. bandingkan dengan pengalamatan secara langsung diatas, dalam instruksi ini alamat memori terlebih dahulu disimpan di R0, dan R0 berperan menunjuk memori mana yang digunakan, sehingga apabila R0 berubah, memori yang ditunjuk juga akan berubah pula. Dalam hal ini R0 berfungsi sebagai register penampung alamat, selain R0, register serbaguna R1 juga dapat digunakan sebagai register penampung alamat. Dalam pengalamatan ini baik RAM *internal* maupun *eksternal* dapat diakses secara tak langsung.

4) Pengalamatan Data dalam Register (*Register Addressing Mode*)

Contoh instruksi : Mov A, R5

Instruksi ini mempunyai makna data dalam register serbaguna R5 dipindahkan ke akumulator A. instruksi ini membuat register serbaguna R0 sampai R7 sebagai tempat penyimpan data yang praktis dan kerjanya sangat cepat.

5) Pengalamatan Kode (*Code Indirect Addressing Mode*)

Contoh instruksi : Movc A, @A+DPTR

Perhatikan dalam instruksi ini Mov digantikan dengan Movc, tambahan huruf c tersebut dimaksudkan untuk membedakan bahwa instruksi ini digunakan didalam memori program (mov tanpa huruf c artinva instruksi dipakai di memori data). Tanda '@' digunakan untuk memberi tanda bahwa A+DPTR digunakan untuk menyatakan alamat memori yang isinya di pindahkan ke akumulator A. dalam hal ini nilai yang tersimpan di dalam DPTR (Data Pointer Register 2 byte) ditambahkan dengan nilai yang tersimpan di akumulator A (1 byte) digunakan untuk menunjuk alamat memori program.

6) Pengalamatan Bit (*Bit Addressable*)

Proses pengalamatan ketika operand menunjuk ke alamat pada RAM internal ataupun register fungsi khusus yang mempunyai kemampuan dialamati secara bit.

Berdasarkan penulisannya, pengalamatan ini terdiri atas beberapa macam:

a) Langsung menunjuk ke alamat bit. contoh Setb 0B0h

b) Menggunakan operator titik. contoh : Setb P3.0

b) Menggunakan lambang standar *Assembler* .Contoh : Clr RXD

c) Menggunakan lambang *Assembler* bebas. contoh : Baca bit P2. 5

Clr Baca

e. Dasar Kerja Program

Program untuk mengendalikan kerja dari mikrokontroler disimpan dalam memori program, program pengendali tersebut merupakan kumpulan dari instruksi kerja mikrokontroler, suatu instruksi MCS-51 merupakan kode yang panjangnya antara 1 sampai 4 Byte.

Selama mikrokontroler bekerja, instruksi kerja tersebut byte demi byte diambil ke CPU dan selanjutnya digunakan untuk mengatur kerja mikrokontroler. Proses pengambilan instruksi dari memori program disebut '*fetch cycles*' dan saat-saat CPU melaksanakan instruksi disebut dengan '*execute cycles*'.

Semua mikrokontroler maupun mikroprosesor dilengkapi dengan sebuah register yang berfungsi khusus untuk mengatur *fetch cycles*, register tersebut dinamakan sebagai *Program Counter* (PC). Nilai PC secara otomatis bertambah satu setiap kali selesai mengambil 1 byte isi memori program, dengan demikian isi memori program dapat berurutan diumpankan ke CPU.

Saat MCS-51 di reset, isi *program counter* di-reset menjadi 0000, artinya sesaat setelah reset, isi dari memori program alamat 0 dan seterusnya akan diambil ke CPU dan diperlakukan sebagai instruksi yang akan mengatur kerja mikrokontroler. Dengan demikian, awal dari program pengendali MCS-51 harus ditempatkan di memori alamat 0, setelah reset MCS-51 menjalankan program mulai dari memori program alamat 0000, dengan melakukan proses *fetch cycles* dan *execute cycles* terus menerus tanpa henti.

Jika sarana interupsi diaktifkan dan tegangan di pin INT0 (pin nomor 6) berubah dari '1' menjadi '0', maka proses menjalankan program diatas, akan

dihentikan sebentar, mikrokontroler melayani dulu permintaan interupsi, CPU akan melanjutkan mengerjakan program utama lagi.

Untuk melaksanakan hal diatas, pertama-tama CPU menyimpan nilai *Program Counter* ke *Stack* (*Stack* merupakan satu bagian kecil dari data memori RAM) kemudian mengganti isi PC dengan 0003. artinya MCS-51 akan melaksanakan program yang ditempatkan di memori program mulai byte ke-3 untuk melayani interupsi yang diterima dari pin INT0. adalah tugas *programmer* untuk mengatur dimana agar program yang dipakai untuk melayani interupsi lewat INT0 harus ditempatkan.

Selesai melayani interupsi, nilai PC yang semula disimpan di dalam *stack* akan dikembalikan ke PC, dengan demikian CPU bisa melanjutkan '*execute cycles*'-nya di program utama.

Selain INT0, AT89C51 dapat menerima interupsi dari INT1, dari UART dan dari *Timer*. Agar permintaan interupsi itu dapat dilayani dengan program yang berlainan, maka masing-masing sumber interupsi itu mempunyai nomor awal program untuk layanan interupsi yang berlainan.

Mikrokontroler AT89S51 ini merupakan salah satu jenis mikrokontroler keluaran 8 bit dan termasuk mikrokontroler keluarga 8051 (MCS-51). Atmel corporation mengembangkan mikrokontroler 8 bit dengan kompatibilitas tinggi, baik dari segi instruksi set maupun arsitektur *hardware*-nya. Untuk kebutuhan mikrokontroler 8 bit tersebut, Atmel mengeluarkan produk dengan kode AT89XXX yang terdiri dari delapan jenis mikrokontroler dengan fitur yang

berbeda. Kedelapan jenis mikrokontroler berikut fiturnya ditunjukkan pada tabel 2.3

Tabel 2.3. Tipe IC Mikrokontroler Keluarga Atmel

Mikrokontroler	Jumlah kaki	Flash PEROM	Jumlah I/O
AT89C1051	20	1 kbyte	15 bit
AT89C2051	20	2 kbyte	15 bit
AT89C4051	20	3 kbyte	15 bit
AT89C51/S51	40	4 kbyte	4X8 bit
AT89C52/S52	40	8 kbyte	4X8 bit
AT89S8252	40	8 kbyte + 2 kbyte EEPROM	4X8 bit
AT89S53	40	12 kbyte	4X8 bit
AT89S55	40	20 kbyte	4X8 bit

AT89S51 diproduksi menggunakan teknologi memori *non volatile* dengan densitas tinggi yang dapat ditulis/hapus ulang sampai seratus kali.

Mikrokontroler AT89S51 mempunyai fitur sebagai berikut:

1. 8 bit CPU yang termasuk keluarga MCS-51
2. 4 kbyte Flash PEROM
3. 128 byte RAM
4. 4X8 bit I/O yang bisa diprogram
5. Osilator internal dan rangkaian pewaktu
6. 2 buah timer/counter 16 bit

7. Sebuah port serial dengan control serial full duplex UART
8. 5 buah jalur interupsi (2 buah interupsi eksternal dan tiga buah interupsi internal). 2 buah mode untuk menghemat catu daya, yaitu Idle Mode yang memungkinkan CPU berhenti tetapi RAM, timer/counter, serial port dan interupai tetap dapat berfungsi. Power Down Mode, memungkinkan RAM tetap menyimpan isinya, mematikan osilator dan fungsi yang lain sampai kemudian ada reset.

2.3 Sensor Suhu Termokopel

Sensor temperatur merupakan bagian dari sistem yang mengubah besaran suhu (temperatur) menjadi sinyal listrik (tegangan dan arus). Sensor digunakan untuk mendeteksi perubahan suhu. Suhu yang telah terdeteksi tersebut kemudian akan dikirim ke bagian pengkondisi sinyal.

Banyak komponen yang bisa digunakan untuk sensor temperatur, misalnya IC temperatur sensor, thermistor, termokopel, pirometer optis, dioda dan transistor dan lain sebagainya. Pemilihannya tergantung dari kebutuhan dan aplikasinya.

Termokopel strukturnya terdiri dari sepasang kawat yang tidak sama dihubungkan bersama-sama pada satu ujung (sebagai ujung pengindera atau ujung panas) dan berakhir pada ujung yang lain (titik referensi atau ujung dingin) yang dipertahankan pada suhu konstan yang diketahui (suhu referensi). Bila antara ujung pengindera dan titik referensi terdapat perbedaan suhu, suatu GGL yang menyebabkan arus dalam rangkaian akan dibangkitkan. Bila titik referensi ditutup oleh sebuah alat ukur, maka penunjukan alat ukur tersebut akan sebanding dengan

selisih suhu antara ujung panas dan titik referensi. Efek termolistrik yang diakibatkan oleh potensial kontak pada titik-titik sambung ini dikenal sebagai efek *Seebeck*.

2.4 *Analog to Digital Converter*

Dalam bidang elektronika dikenal dua jenis rangkaian untuk menjembatani sistem analog ke digital. Jenis pertama dikenal sebagai rangkaian konversi besaran digital ke analog (*Digital Analog Converter* disingkat DAC). Dan yang kedua adalah rangkaian konversi besaran analog ke digital (*Analog Digital Converter* atau disingkat ADC).

Ada tiga spesifikasi yang perlu diperhatikan dalam memilih ADC ataupun DAC, antara lain :

1. Resolusi

   Resolusi adalah perubahan masukan terkecil yang menyebabkan berubahnya *output*. Resolusi dapat dirumuskan sebagai berikut:

$$\text{Resolusi } (r) = \left(\frac{1}{2^n - 1} \right) \times Vref \qquad (2.1)$$

Dimana : n adalah banyaknya bit ADC

Vref adalah tegangan referensi input yang digunakan (*Volt*)

Resolusi ADC dapat juga dianggap sebagai kesalahan kuantisasi. Semakin banyak jumlah kode yang mengkode sinyal maka kesalahan kuantisasi ini

semakin dapat direduksi. Kesalahan kuantisasi secara umum adalah ± LSB. Perumusan resolusi secara umum adalah sebagai berikut:

$$\% \text{ Resolusi} = \left(\frac{1}{2^n - 1} \right) \times 100\ \% \qquad (2.2)$$

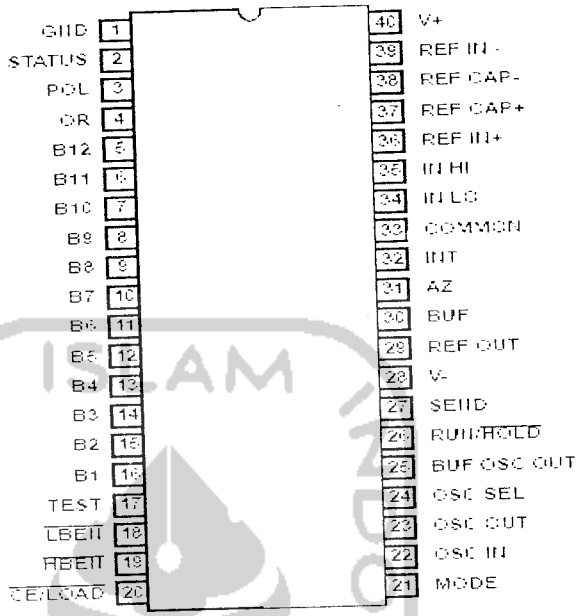
2. Akurasi

Keakuratan sebuah ADC tergantung pada resolusi, dan pada keakuratan komponen-komponen penyusunnya, seperti: komparator, resistor dan level *amplifiernya.*

3. Waktu konversi

Waktu konversi yaitu waktu yang dibutuhkan IC untuk mengubah tegangan masukan analog menjadi data keluaran biner. Waktu konversi (*Conversion Time*) ini sangat penting karena *bandwidth* sinyal *input* merupakan hal yang signifikan. Waktu konversi yang umum untuk sebuah IC berkisar antara 0,05 sampai 100.000 μs dengan keluaran biner. Sedangkan waktu konversi untuk keluaran desimal biasanya berkisar antara 200 sampai 400 ms.

Konfigurasi pin dari ADC ICL7109 ditunjukkan pada Gambar 2.6



Gambar 2.6 Susunan pin ADC ICL 7109

2.4.1 Ketelitian dan Resolusi ADC

Karena pengubah A/D atau juga ADC merupakan suatu sistem simpul tertutup yang mencakup baik sistem analog maupun digital, ketelitian keseluruhan harus mencakup kesalahan-kesalahan dari posisi analog maupun digital. Dalam menentukan ketelitian keseluruhan, cara yang termudah adalah memisahkan kedua sumber kesalahan tersebut.

Jika diasumsikan bahwa semua komponen beroperasi sebagaimana mestinya, sumber kesalahan digital hanya ditentukan oleh resolusi sistem. Dalam mendigitalisasi suatu tegangan analog, biasanya langkah yang harus dilakukan adalah mengupayakan untuk suatu tegangan analog kontinue dengan seperangkat

bilangan ekivalennya. Bila peringkat-peringkat digital diubah kembali ke dalam bentuk analog oleh rangkaian tangga, keluarannya adalah bentuk gelombang tangga. Bentuk gelombang ini merupakan representasi tegangan masukan namun jelas bukan merupakan suatu sinyal tak-kontinue yang tersusun atas sejumlah jenjang-jenjang diskrit. Dalam berupaya untuk mereproduksi sinyal masukan analog, hal terbaik yang dapat dilakukan adalah menggunakan jenjang yang paling mendekati tegangan masukan dalam amplitudonya.

Kenyataannya adalah jenjang-jenjang pada tegangan tangga mengakibatkan timbulnya kesalahan dalam sistem. Jenjang digital terkecil atau *kuantum*, berkaitan dengan LSB dan hanya dapat dikecilkan dengan meningkatkan banyaknya bit dalam pencacah. Kesalahan yang merupakan pembawaan seringkali disebut kesalahan kuantisasi dan pada umumnya sebesar +1 bit.

Sumber utama kesalahan analog adalah dalam pengubah A/D kemungkinan besar adalah pembanding. Sumber-sumber kesalahan lainnya adalah tahanan-tahanan dalam tangga , kerut (*ripple*) pada penyedia tegangan acuan serta derau. Namun demikian, pada umumnya semua ini dapat dianggap sumber kesalahan sekunder dibandingkan kesalahan dalam pembanding.

Sumber-sumber kesalahan terpusat disekitar perubahan-perubahan titik perpindahan DC. Titik perpindahan DC adalah perbedaan antara peringkat-peringkat tegangan masukan yang mengakibatkan keluaran berubah keadaan. Perubahan-perubahan titik perpindahan terutama diakibatkan oleh offset (pengimbangan), penguat dan linieritas penguat yang digunakan dalam

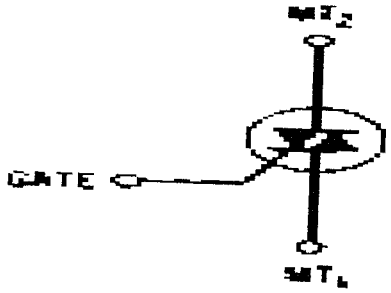
pembanding. Parameter-parameter ini pada umumnya sedikit berubah dengan berubahnya peringkat-peringkat tegangan masukan dan seringkali juga dengan berubah temperaturnya. Perubahan ini yang mengakibatkan kesalahan analog dalam sistem.

2.5. TRIAC

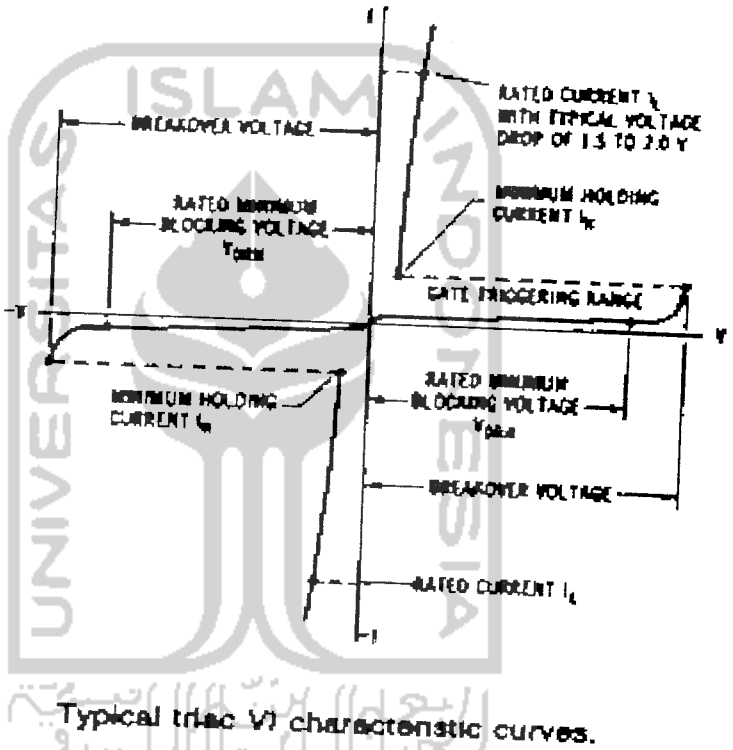
TRIAC adalah semikonduktor yang memiliki tiga terminal (pin), yang digunakan untuk mengontrol arus dari dua arah. TRIAC terlihat seperti dua buah SCR yang disatukan (arah berlawanan) namun dengan satu gerbang pemicu.pin utama dideskripsikan sebagai MT1 dan MT2. Sebuah TRIAC beroperasi hampir sama dengan operasi SCR, namun TRIAC dapat dipicu pada sebelah manapun, baik pada pulsa positif atau negatif pada terminal gerbang tunggalnya. Arus batas minimum harus dipertahankan untuk menjaga agar TRIAC tetap menghantar.

Yang harus diperhatikan dari TRIAC :

1) Tegangan batas maksimum
2) Arus maksimum
3) Arus bawah minimum.
4) Tegangan gerbang dan arus picu gerbang yang dibutuhkan.
5) Switching speed.
6) Nilai dV/dt maksimum.



Gambar 2.7. TRIAC

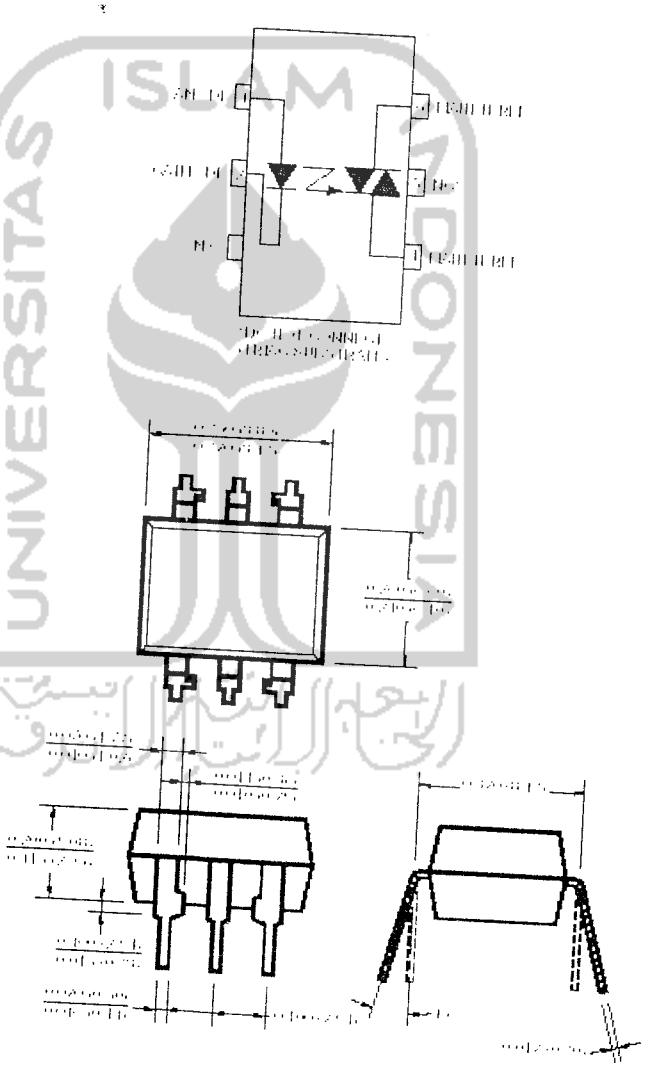


Gambar 2.8 Daerah operasi TRIAC

2.6. Opto-coupler (MOC 3021)

. Opto-coupler adalah alat elektronik yang terbuat dari bahan Ga As (Galium dan Arsen) yang memancarkan cahaya, kemudian cahaya tersebut akan menghidupkan DIAC. Alat ini terpaket dalam bentuk IC dalam hal ini, diharapkan

supaya dapat menyebabkan terjadinya transfer energi listrik dalam satu arah dari tegangan yang berbentuk cahaya ke rangkaian berikutnya, sambil mempertahankan isolasi listrik diantara kedua siklus. Kalau logika 0 diumpankan kemasukan, LED akan menghantar dan memancarkan cahaya yang dapat menghidupkan DIAC. DIAC akan memicu gerbang pada TRIAC.



Gambar 2.9. MOC 3021

2.7 IC TL 074 (*Operational Amplifier*)

IC TL 074 adalah merupakan IC op-amp, dalam satu *package* IC ini didalamnya terdapat empat buah op-amp. Tegangan catu yang umum diberikan pada IC ini adalah merupakan tegangan catu simetris –V, 0V dan +V.

Penguat operasional (Op-Amp) merupakan suatu penguat berperolehan tinggi di kopel langsung, dimana umpan balik ditambahkan untuk mengendalikan karakteristik tanggapan keseluruhannya. Op-Amp digunakan untuk membentuk fungsi-fungsi linier yang bermacam, juga operasi-operasi tak linier dan sering disebut rangkaian analog.

Sejumlah besar penguat Op-Amp mempunyai sebuah masukan differensial dengan tegangan V_2 dan V_1 yang diberikan pada terminal pembalik (inverting) dan bukan pembalik (noninverting). Perolehan antara V_O dan V_1 positif (bukan pembalik/noninverting), sedangkan penguatan antara V_O/V_2 negatif (pembalik/inverting).

Suatu penguat dengan satu ujung dapat dianggap sebagai suatu perisriwa khusus, dimana salah satu terminal masuk dibumikan. Hampir semua Op-Amp hanya mempunyai satu terminal keluaran. Sangat sering dibutuhkan sebuah penguat yang mempunyai keluaran sefasa dengan masukan.

2.8. Modul LCD (*Liquid Crystal Display*)

LCD adalah penampil (*display*) dari bahan cairan kristal yang pengoperasiannya menggunakan sistem *dot matriks*. LCD yang digunakan pada alat ini adalah LCD M1632. LCD ini merupakan modul LCD dengan tampilan

16 x 2 baris dengan konsumsi daya yang rendah. Modul ini di lengkapi dengan LCD mikrokontroller HD44780 buatan Hitachi yang berfungsi sebagai pengendali. LCD ini mempunyai CGROM (*Character Generator Read Onli Memory*), CGRAM (*Character Generator random Access Memory*) dan DDRAM (*Display Data random Access Memory*), dan juga memiliki 3 *bit control* yaitu E yang merupakan input *clock*, R/W sebagai *input* untuk memilih *read* atau *write* dan RS sebagai *register select,* juga memiliki 8 *bit* data yaitu DB0 sampai DB7.

Register untuk membaca perintah-perintah dari mikrokontroler ke HD44780 pada saat proses penulisan data atau tempat status dari HD44780 adalah register perintah. Penulisan data ke register perintah di lakukan dengan tujuan mengatur tampilan LCD, inisialisasi dan mengatur *address counter* maupun *address data*. Kondisi RS berlogika '0' menunjukan akses data ke register perintah. R/W berlogika '0' menunjukan proses penulisan data akan di lakukan. Untuk mode 8 *bit interface*, proses penulisan dapat berlangsung dilakukan secara 8 *bit* dan di awali sebuah pulsa logika '1' pada E *clock*.

Proses pembacaan data pada register perintah bisa digunakan untuk melihat status *busy* dari LCD atau membaca *address counter*. RS diatur pada logika 0 untuk akses ke register perintah. R/W diatur pada logika 1 yang menunjukan proses pembacaan data. Untuk mode 8 *bit interface*, proses penulisan dapat berlangsung dilakukan secara 8 *bit* (*nibble* tinggi dan rendah) dan di awali sebuah pulsa logika '1' pada E *clock*.

Sedangkan register data adalah register dimana mikrokontroler dapat menuliskan atau membaca data ke atau dari DDRAM. Penulisan data pada

register ini akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur sebelumnya. Penulisan data pada register data dilakukan untuk mengirimkan data yang akan di tampilkan pada LCD. Pada mode 8 *bit interface*, proses penulisan dapat berlangsung dilakukan secara 8 *bit* dan di awali sebuah pulsa logika '1' pada E *clock*. Pembacaan data dari register data di lakukan pula untuk membaca kembali data yang ditampikan ke LCD. Proses dilakukan dengan mengatur RS pada logika '1' yang menunjukan adanya akses ke Register Data. Kondisi R/W diatur pada lgika tinggi yang menunjukan adanya proses pembacaan data. Data 8 *bit* dibaca dengan menggunakan pulsa logika '1' yang juga di awali dengan pulsa logika '1' pada E *Clock*.

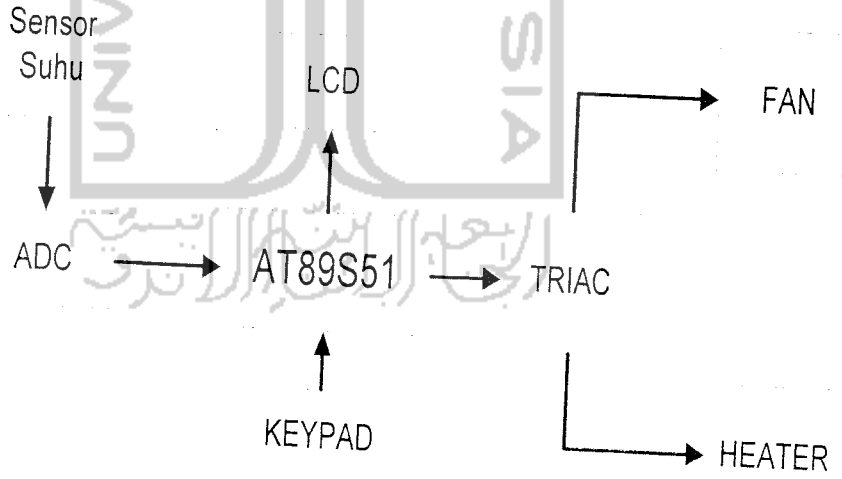
BAB III

PERANCANGAN PERANGKAT KERAS DAN PERANGKAT LUNAK

Dalam bab ini akan dibahas mengenai perancangan alat, dikarenakan pada alat ini digunakan Mikrokontroler yang membutuhkan program (*software*) yang harus diisikan (*download*) kedalam *chip* Mikrokontroler tersebut, maka mengenai perancangan alat ini dibagi menjadi dua bagian yaitu perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*).

3.1 Perancangan Perangkat Keras

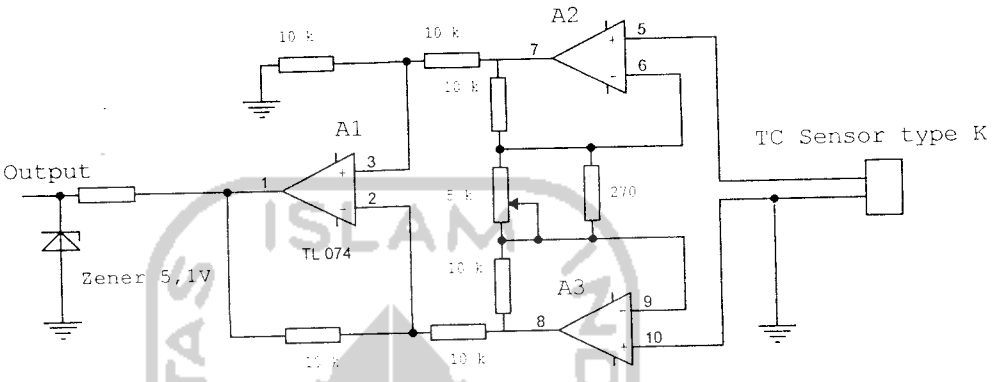
Diagram Blok perangkat keras dari rangkaian alat yang dibuat :



Gambar 3.1 Diagram Blok Rangkaian

3.1.1 Rangkaian Sensor Suhu

Sensor suhu yang digunakan dalam rangkaian ini adalah sensor suhu termokopel tipe K. Keluaran dari sensor ini harus diperkuat agar dapat dibaca oleh rangkaian selanjutnya. Rangkaian penguat untuk sensor adalah sebagai berikut :



Gambar 3.2 Rangkaian sensor suhu

IC yang digunakan pada rangkaian sensor adalah IC TL 074. IC ini berisi 4 buah op-amp didalamnya. Rangkaian ini membutuhkan suplay catu simetris yaitu –12V, 0V dan +12V, yang dipasang pada pin 11 dan 4 dari IC tersebut.

Sensor suhu termokopel type k ini mempunyai tegangan keluaran yang sangat kecil sekali yaitu dalam orde mikro Volt. Op-amp bekerja sebagai pendeteksi kenaikan tegangan yang dihasilkan oleh termokopel sesuai dengan kenaikan suhu yang diterimanya

Pada *output* nampak terpasang dioda zener dengan tegangan pembatas sebesar 5,1 Volt ini dimaksudkan agar tegangan yang dihasilkan oleh rangkaian sensor ini terbatas hanya sebesar 5 Volt, dikarenakan tegangan referensi yang digunakan pada rangkaian ADC adalah sebesar 5 Volt

3.1.2 Rangkaian ADC ICL 7109

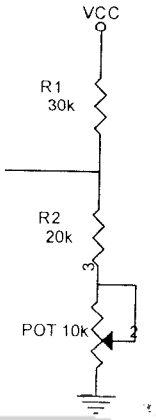
Data analog suhu yang dihasilkan oleh rangkaian sensor suhu agar dapat dibaca oleh IC Mikrokontroler haruslah diubah terlebih dahulu ke dalam bentuk biner digital. Untuk itu maka diperlukan pengubah besaran analog ke digital yang biasa disebut ADC (*Analog to Digital Converter*).

ICL7109 adalah ADC dengan keluaran 12 *bit* dan memerlukan kristal dengan frekuensi 3,579545 MHz untuk memproses sinyal. ADC ini membutuhkan suplai tegangan ganda yaitu +5 volt dan -5 volt. Untuk menghasilkan pengskalaan penuh 12 *bit* atau 2^{12} (4096) dibutuhkan tegangan *input* maksimal sebesar 4,096 volt. Jika tegangan *input* sebesar 4,096 volt digunakan maka dibutuhkan juga tegangan referensi *input* (*Vref*) sebesar 2,048 volt. untuk memperoleh *Vref* yang maksimal, maka dibuat pembagi tegangan dengan kombinasi R sebagai berikut:

$$\frac{Vout}{Vin} = \frac{R_2}{R_1 + R_2} \qquad (3.1)$$

$$\frac{2V}{5V} = \frac{20}{30 + 20}$$

Dari persamaan diatas diperoleh besarnya nilai R_1 = 30 kΩ dan R_2 = 20 kΩ. Penambahan VR sebesar 10 kΩ dimaksudkan untuk menjaga penurunan tegangan pada R_2 sehingga dapat dilakukan pengesetan.



Gambar 3.3. Rangkaian Pembagi Tegangan

Dengan menggunakan tegangan *input* (V_{in}) sebesar +4 volt, maka ketelitiannya dapat ditentukan dengan persamaan :

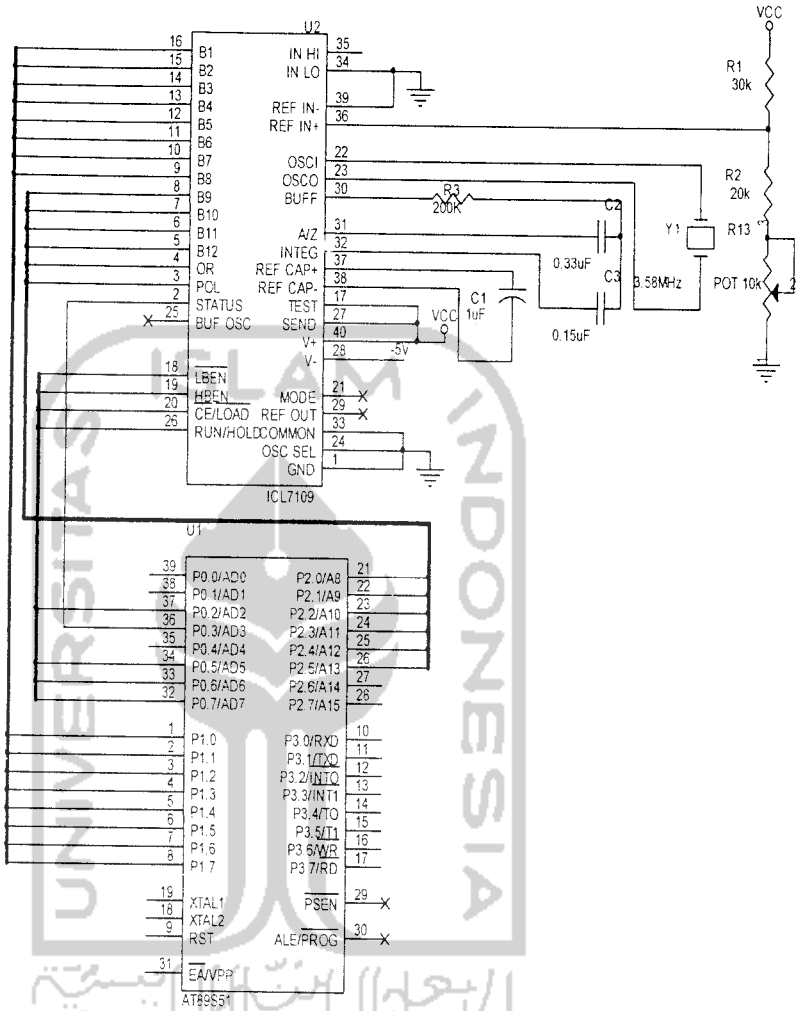
$$ketelitian\ (V) = \frac{V_{input}}{resolusi} \qquad (3.2)$$

$$= \frac{4}{2^{12} - 1}$$

$$= \frac{4}{4095}$$

$$= 0{,}0009768 \text{ V atau } 0.98\ {}^{mV}/_{bit}$$

Rangkaian sistem minimum ADC ICL7109 ditunjukkan pada Gambar 3.4.

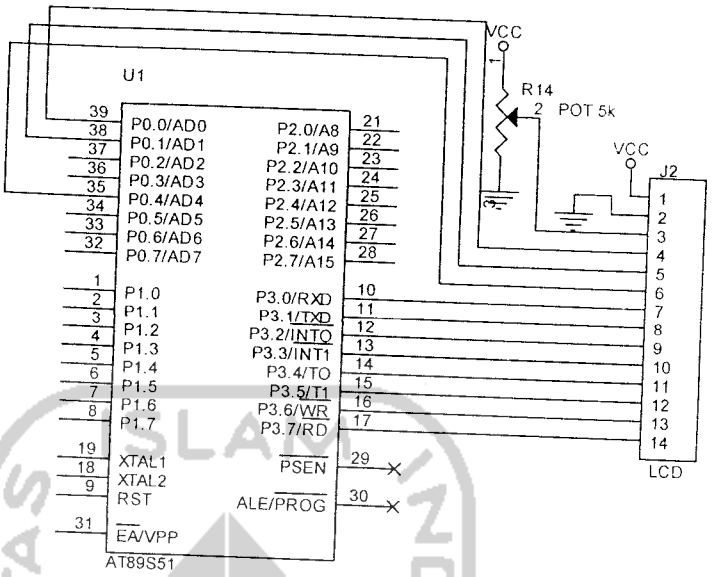


Gambar 3.4 Rangkaian ADC

3.1.3 Rangkaian LCD (*Liquid Crystal Display*)

Untuk mengetahui nilai suhu yang sedang berlangsung dan untuk mengetahui nilai target suhu yang telah dimasukkan, diperlukan sebuah *display* untuk menampilkannya. Pada alat pengatur suhu ini digunakan *display* LCD.

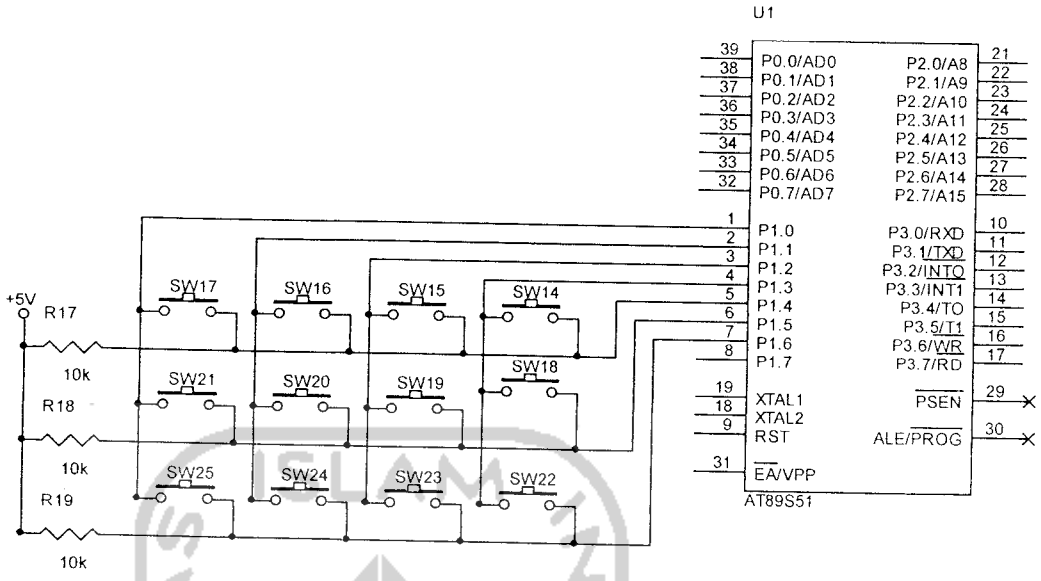
Modul LCD yang digunakan sebagai display adalah LCD seri M1632 produksi *Seiko Instruments Inc.* yang dapat menampilkan 16X2 karakter. LCD M1632 ini terdiri dari 2 bagian, yang pertama merupakan panel display LCD sebagai media penampil informasi dalam bentuk huruf atau, angka (Karakter) dua baris, masing masing baris bisa menampung 16 huruf atau angka (Karakter). Bagian kedua merupakan sebuah sistem, yang dibentuk oleh mikrokontroler yang ditempelkan di balik panel LCD (*Chipset*) yang berfungsi untuk mengatur tampilan informasi serta untuk mengatur komunikasi LCD dengan mikrokontroler yang memakai tampilan LCD itu. Dengan demikian pemakaian LCD M1632 menjadi sederhana karena dengan suatu perintah tertentu kemudian dilanjutkan dengan pengiriman kode-kode ASCII, maka informasi akan ditampilkan seperti umumnya pemakaian sebuah printer. Untuk berhubungan dengan mikrokontroler pemakai LCD M 1632 dibutuhkan 8 jalur data (DB0 .... DB7) yang dipakai untuk menyalurkan kode ASCII maupun perintah pengatur kerja LCD M1632, selain itu dilengkapi pula dengan jalur EN, R/W dan RS seperti layaknya komponen yang kompatibel dengan mikroprosesor, kombinasi sinyal EN dan R/W merupakan sinyal standar pada komponen buatan Motorola, sebaliknya sinyal khas Intel dengan kombinasi sinyal WR dan RD. perbedaan ini membuat teknik penggabungan LCD dengan mikrokontroler AT89S51 memerlukan suatu teknik yang khusus yaitu dengan mensimulasikan sinyal-sinyal tersebut pada port mikrokontroler AT89S51. Pemasangan LCD pada port I/O mikrokontroler AT89S51 diperlihatkan pada gambar 3.5.



Gambar 3.5 Rangkaian Display

3.1.4 Rangkaian *Keypad*

Nilai target suhu yang akan ditentukan dimasukkan dengan menggunakan *keypad*. *Keypad* yang digunakan pada alat kendali suhu ini menggunakan *keypad* 12 tombol, yaitu tombol angka dari 0 – 9, satu tombol *start* untuk memulai proses kendali dan satu tombol *stop* untuk menghentikan proses kendali untuk memasukkan nilai target suhu yang baru. Gambar dari koneksi rangkaian *keypad* dengan Mikrokontroler adalah sebagai berikut :



Gambar 3.6 Koneksi *keypad* ke Mikrokontroler

Keypad yang digunakan disini adalah keypad dengan konfigurasi matriks yang didalamnya sebanyak 12 buah. Untuk menghemat penggunaan pin pada Mikrokontroler, maka hanya digunakan 7 buah pin. Yaitu pin P1.0 – P1.6.

3.1.5 Rangkaian TRIAC Driver

Alat pengatur suhu ini menggunakan rangkaian TRIAC sebagai driver kipas (fan) dan pemanas (heater). Metode penyulutan TRIAC dibagi menjadi 5 kondisi (level). Kondisi ini akan menentukan waktu penyulutan, yang akan mempengaruhi besar tegangan yang dialirkan oleh TRIAC.

Perhitungan waktu penyulutan TRIAC sebagai berikut :

$f = 50$ Hz , sehingga $T = \frac{1}{f}$

maka, T = 0,02 second $\rightarrow$ T = 360 °

0,02 s = 360 °

0.01 s = 180°

Digunakan rumus perbandingan :

$$\frac{t}{t'} = \frac{\theta}{\theta'}$$

$$\mathbf{t1} = \frac{36° \times 0.01s}{180°}$$

$$= 0.002 \text{ s}$$

$$\mathbf{t2} = \frac{72° \times 0.01s}{180°}$$

$$= 0.004 \text{ s}$$

$$\mathbf{t3} = \frac{108° \times 0.01s}{180°}$$

$$= 0.006 \text{ s}$$

$$\mathbf{t4} = \frac{144° \times 0.01s}{180°}$$

$$= 0.008 \text{ s}$$

t5 = TRIAC tidak disulut.

dimana : f = frekuensi

T = amplitude

antara komputer dengan IC Mikrokontroler tersebut. Untuk mengisikan program yang telah jadi kedalam Mikrokontroler AT89S51, penulis menggunakan ATMEL *Universal Microcontroller Writer* yang merupakan produk dari BERIN Elektronik

IC Mikrokontroler AT 89S51 yang digunakan pada alat pengatur suhu ini *Flash Memory* nya dapat dihapus tulis hingga sampai 1000 kali, sehingga bila terjadi kesalahan dari instruksi yang telah dimasukkan kedalam IC tersebut sehingga menimbulkan kesalahan kinerja pada alat yang dibuat, maka program yang sudah masuk kedalam IC tersebut dapat dihapus dan diisikan kembali dengan file program yang sudah diedit.

3.2.1 Penerapan logika *fuzzy* pada sistem

Logika *fuzzy* yang diterapkan pada alat kendali suhu ini, implementasi langsungnya adalah pada perangkat lunak yang akan di *download* kedalam Mikrokontroler, sedangkan metode yang digunakan dalam menerapkan logika *fuzzy* ini didalam Mikrokontroler ini adalah dengan menggunakan rumus-rumus matematis yang diterapkan kedalam bahasa *assembler*.

Seperti alat kendali suhu pada umumnya, biasanya hasil dari pembacaan sensor bila dibandingkan dengan masukan *set point*, akan memiliki tiga kemungkinan *error*, yaitu *error* negatif, *zero* dan positif. *Error* negatif adalah bila masukan *setpoint* melebihi masukan pembacaan sensor (suhu *current*), *zero* adalah bila sama dan positif bila masukan *setpoint* ada dibawah suhu *current*.

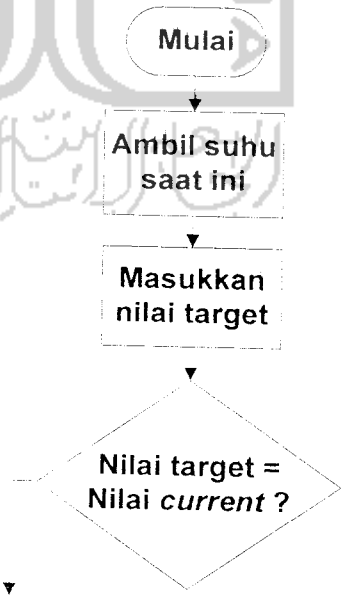
3.2.1.1. Fuzzifikasi

Proses fuzzifikasi antara pemanas dan kipas dalam alat yang dibuat ini memiliki perbedaan ketentuan, dimana perbedaan hanya terletak pada range suhunya, hal ini dilakukan adalah karena proses pemanasan dapat berlangsung lebih cepat bila dibandingkan proses pendinginan. Pada pemanas range pemetaan suhunya adalah pada skala 5 ^{0}C sedangkan pada kipas adalah 2 ^{0}C.

Proses matematis yang dilakukan oleh Mikrokontroler untuk menentukan fuzzifikas*i* ini adalah sebagai berikut :

Nilai target – Nilai *Current* = Nilai *error*..... (3.3)

Bila nilai *error* nya negatif berarti masukan target lebih besar dari nilai *current* dengan demikin proses fuzzifikasi akan menunjuk ke pemanas, seperti yang telah disebutkan diatas tadi bahwa untuk pemanas dipakai skala 5^{0}C. maka hasil *error* tadi akan dibagi dengan angka 5. Di bawah ini adalah diagram alir fuzzifikasi :





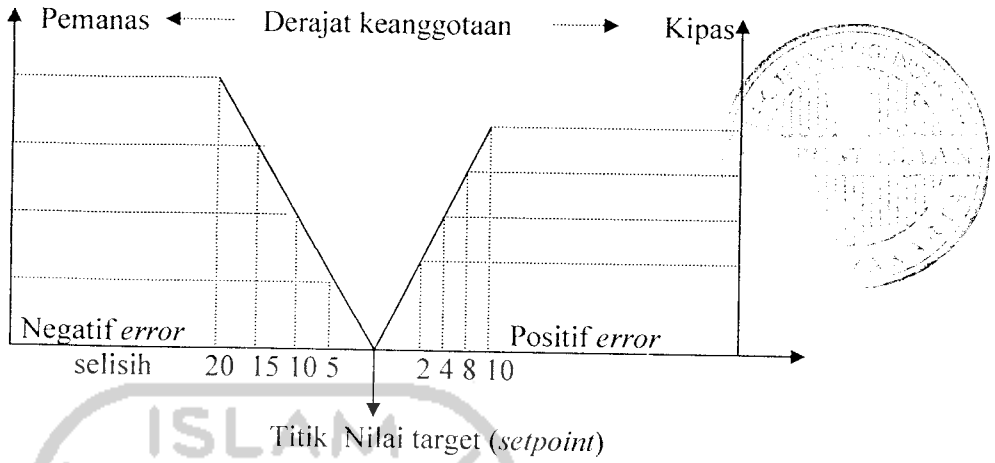
Gambar. 3.8 Diagram alir fuzzifikasi

Bila hasil *error* positif maka Mikrokontroler akan membalik posisi pengurangan persamaan 3.3. tadi menjadi :

Nilai *Current* – Nilai target = Nilai *error* ..... (3.4)

Dan proses fuzzifikasi akan menunjuk ke kipas, dan Nilai *error* ini akan dibagi 2. Bila hasil pengurangan dari persamaan 3.3. dan 3.4 sama dengan nol, maka proses pembagian dengan 5 dan 2 tadi ditiadakan dan antara kipas dan pemanas memiliki kondisi yang sama yaitu *off*.

Adapun grafik dari proses fuzzifikasi adalah sebagai berikut :



Gambar 3.9 Grafik derajat keanggotaan *error* hasil proses fuzzifikasi

3.2.1.2. Inferensi

Dari hasil fuzzifikasi nilai *error* yang didapat, maka dibuatlah aturan-aturan yang akan menentukan proses kendali. Aturan-aturan logika *fuzzy* yang akan dipergunakan sangat bergantung pada sistem yang akan dikendalikan, disamping itu tujuan pengendalian juga sangat mempengaruhi penentuan aturan-aturan kendali. Tidak ada rumusan yang pasti dalam menentukan aturan-aturan *fuzzy* dan fungsi keanggotaan masukan dan keluaran. Pengetahuan seorang operator yang berpengalaman dalam pengendalian dapat dijadikan acuan dalam perancangan pengendali dengan menggunakan logika *fuzzy*.

Adapun penentuan aturan kendali dari sistem yang dibuat diberikan sebagai berikut :

1. Semakin besar nilai *error* negatif berarti suhu yang diinginkan (*setpoint*) masih jauh diatas, sehingga penyalaan pemanas dilakukan pada level maksimal.
2. Semakin besar nilai *error* positif berarti suhu yang diinginkan masih jauh dibawah, sehingga penyalaan kipas dilakukan pada level maksimal.
3. Jika nilai *error* sama dengan nol, maka pemanas dan kipas akan di matikan.

3.2.1.3. Defuzzifikasi

Proses defuzzifikasi adalah bagian akhir dari suatu logika fuzzy yang bertujuan untuk memetakan himpunan fuzzy hasil inferensi ke himpunan keluaran yang berupa aksi kendali. Dari beberapa metode defuzzifikasi seperti *Mean of Maximum* (MOM), dan *Center of Area* (COA), tugas akhir ini menggunakan metode COA, karena kemudahan dalam komputasi, sifat kontinuitas dan kemudahan implementasinya. Fungsi keanggotaan keluaran yang digunakan adalah fungsi keanggotaan *singletone* (konstan).

Pemetaan dari proses defuzzifikasi ini adalah sebagai berikut :

Untuk pemanas (*heater*)

Nilai *error* : 5 = X, dimana nilai variabel X memiliki ketentuan :

Bila X = 0, (nilai *error* = 1-4), keluaran = **t4**

Bila X = 1, (nilai *error* = 5-9), keluaran = **t3**

Bila X = 2, (nilai *error* = 10-14), keluaran = **t2**

Bila X = 3, (nilai *error* = 15 keatas), keluaran = **t1**

Apabila aturan tersebut diatas dicek menggunakan grafik derajat keanggotaan *error* hasil proses fuzzifikasi akan didapatkan hasil sebagai berikut:

Tabel 3.1 Hasil pengecekan aturan fuzzy pada pemanas menggunakan grafik derajat keanggotaan *error*.

Suhu *current* (^{0}C)	Suhu target (^{0}C)	Nilai *error* (^{0}C)	Nilai *error* : 5 = X	Keterangan
25	45	20	20 : 5 = 4	**t1**
35	44	9	9 : 5 = 1.8	**t3**
30	44	14	14 : 5 = 2.8	**t2**
47	50	3	3 : 5 = 0.6	**t4**

Untuk Kipas (*fan*) :

Nilai *error* : 2 = X, dimana nilai variabel X memiliki ketentuan :

Bila X = 0, (nilai *error* = 1), keluaran = **t4**

Bila X = 1, (nilai *error* = 2-3), keluaran = **t3**

Bila X = 2, (nilai *error* = 4-5), keluaran = **t2**

Bila X = 3, (nilai *error* = 6 keatas), keluaran = **t1**

Untuk nilai *error* = 0, maka pemanas dan kipas akan di *off* kan.

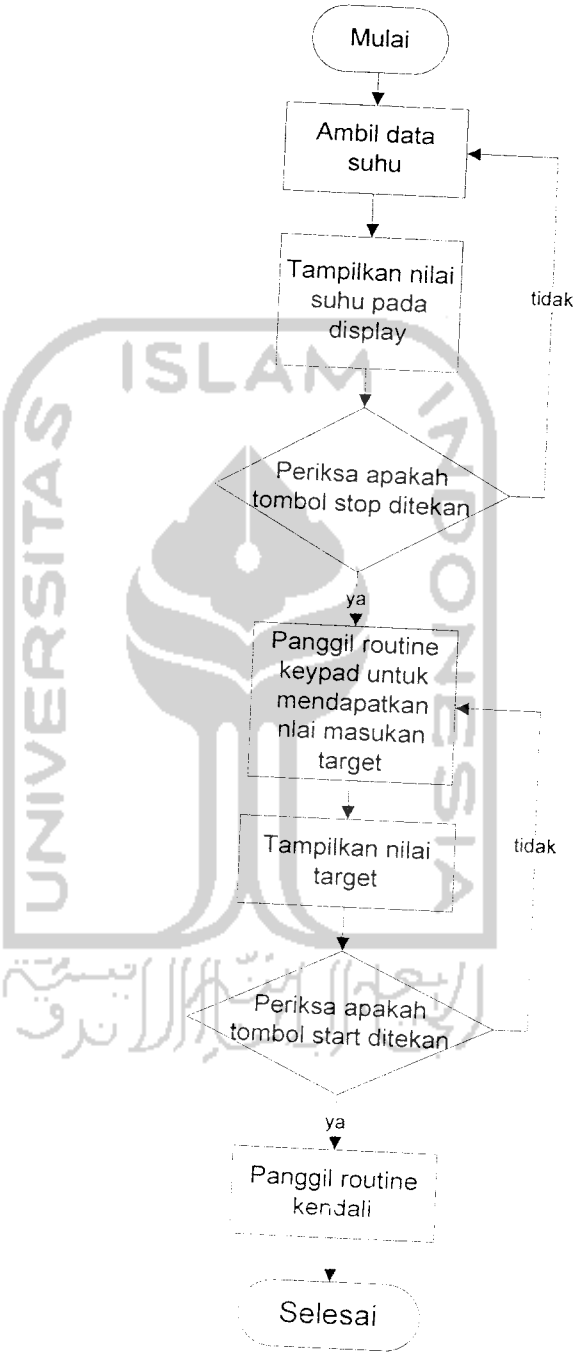
Apabila aturan tersebut diatas dicek menggunakan grafik derajat keanggotaan *error* hasil proses fuzzifikasi akan didapatkan hasil sebagai berikut:

Tabel 3.2 Hasil pengecekan aturan fuzzy pada kipas menggunakan grafik derajat keanggotaan *error*.

Suhu current (^{0}C)	Suhu target (^{0}C)	Nilai *error* (^{0}C)	Nilai *error* : 2 = X	Keterangan
40	34	6	6 : 2 = 3	**t1**
54	52	2	2 : 2 = 1	**t3**
35	30	5	5 : 2 = 2.5	**t2**
51	50	1	1 : 5 = 0.2	**t4**

Dari hasil pengamatan didapatkan bahwa besarnya nilai *error* akan menentukan pemanas atau kipas yang bekerja berdasarkan aturan-aturan *fuzzy* yang dibuat. Sedangkan grafik derajat keanggotaan *fuzzy* hasil proses fuzzifikasi digunakan untuk mengelompokkan keanggotaan himpunan *fuzzy*.

Berikut ini adalah diagram alir dari program kendali suhu dengan menggunakan logika *fuzzy* yang akan dibuat :



Gambar 3.10 Diagram alir program kendali suhu

Saat pertama kali alat kendali suhu ini dihidupkan setelah proses inisialisai bit *input* dan *output* maka Mikrokontroler akan mengambil data suhu yang terbaca pada termokopel kemudian menampilkan pada display. Proses ini akan terus berlangsung sampai ada *user* yang menekan tombol *stop* (#), jika tidak ada, mikrokontroler akan selalu mengambil data suhu dan menampilkan pada display.

Bila suatu saat *user* menekan tombol *stop* maka Mikrokontroler akan melanjutkan langkah dengan memanggil program *keypad* agar *user* dapat memberikan nilai masukan target suhu yang diinginkannya. Setelah proses pengetikan nilai target ini oleh *user* maka kemudian Mikrokontroler akan menunggu apakah *user* menekan tombol *start* (*) atau *stop*. Bila menekan tombol *start*, maka proses kendali suhu akan berlangsung, sedangkan bila tombol *stop* yang ditekan maka berarti proses kendali suhu dibatalkan dan program akan kembali ke label *start* awal.

Berikut ini beberapa potongan-potongan program penting yang dibuat sebagai berikut :

3.2.2. *Routine* program pengambil data suhu

Data suhu yang akan diolah oleh Mikrokontroler diperoleh dari rangkaian ADC ICL 7109, proses kerja dari rangkaian ADC ini dikontrol sepenuhnya oleh Mikrokontroler itu sendiri, mulai dari *start* hingga *end of conversion*, berikut ini adalah potongan program pengambilan data dari ADC ICL 7109 oleh Mikrokontroler

```
disp:  mov  lcd,#89h
```

```
        acall writc
        mov   r0,#11h          ;11h-13h memori seting
disp1:  mov   a,@r0
        add   a,#30h
        mov   lcd,a
        acall writd
        inc   r0
        cjne  r0,#14h,disp1
        mov   lcd,#0c9h        ;14h-16h suhu pembacaan
        acall writc
        mov   r0,#14h
```

3.2.3 *Routine* pemberian instruksi dan penulisan data ke LCD

1. Potongan program perintah penulisan instruksi ke LCD

```
writc:  clr   rs
        mov   p3,lcd
        setb  ena
        acall delay
        clr   ena
        acall delay
        setb  rs
        ret
```

2. Potongan program perintah penulisan data ke LCD

```
writd:    setb  rs
          mov   p3,lcd
          setb  ena
          acall delay
          clr   ena
          acall delay
          clr   rs
          ret
```

Dari dua potongan program diatas, nampak bahwa perbedaan antara penulisan data dan penulisan instruksi ke LCD hanyalah pada baris pertama, dan ketujuh dimana pada penulisan instruksi nilai RS diberi *low* sedangkan pada penulisan data diberi nilai *high*.

3.2.4 *Routine* program *keypad*

Program *keypad* digunakan agar masukan dari *keypad* dapat disandikan menjadi angka-angka desimal dan instruksi lain yang dapat diolah oleh Mikrokontroler. Adapun potongan program dari *routine keypad* ini adalah sebagai berikut

```
input:
          push  0h
          push  1h
          mov   b,#0ffh      ;hasil pada a
```

```
        mov   r0,#0h      ;reg a,b,r0,r1,dptr
        mov   r1,#7fh
atas:   mov   p1,r1
        mov   a,p1
        jb    acc.0,siji
        mov   b,r0
siji:   inc   r0
        jb    acc.1,loro
        mov   b,r0
loro:   inc   r0
        jb    acc.2,telu
        mov   b,r0
telu:   inc   r0
        jb    acc.3,papat
        mov   b,r0
papat:  cjne  r1,#0efh,rty1x
        sjmp  keyjump
```

3.2.5 *Routine* kendali suhu

Setelah nilai target dimasukkan dan kemudian tombol *start* ditekan yang diartikan oleh Mikrokontroler sebagai dimulainya kendali suhu, maka Mikrokontroler akan memanggil *routine* kendali suhu, yang potongan *listing* programnya adalah sebagai berikut :

```
rs      bit   p0.2
ena     bit   p0.0
rw      bit   p0.1
stb     bit   psw.5      ;start stop bit blower
sth     bit   psw.1      ;start stop bit heater
heater bit    p2.6
blower bit    p2.7
lben    bit   p0.7
hben    bit   p0.6
cel     bit   p0.5
run     bit   p0.4
status  bit   p0.3
lcd     equ   8h
hetco   equ   9h
hetco1 equ    0ah
bloco   equ   0bh
bloco1 equ    0ch
dhigh   equ   0dh
dlow    equ   0eh
seth    equ   0fh
setl    equ   10h
temph  equ    17h
templ  equ    18h
```

```
temps       equ   19h
add_data    equ   1ah
add_conv    equ   1bh
byte_data   equ   1ch
byte_conv   equ   1dh
countx      equ   1eh
aux1        equ   1fh
aux2        equ   20h
aux3        equ   21h
```

Langkah pertama dari program ini adalah membandingkan nilai target suhu dengan nilai suhu yang sedang berlangsung, yaitu dengan mengurangkan nilai target dengan nilai suhu *current*. Bila nilai target lebih besar, maka nilai bit *carry* (C) akan bernilai 0 dan ini menandakan bahwa perintah kepada pemanas (*heater*) akan dimulai, akan tetapi sebelum dimulai terlebih dahulu dicek selisih dari nilai target dengan nilai *current*, untuk menentukan kondisi penyulutan yang akan digunakan. Bila pada pengurangan nilai target dengan nilai *current* suhu membuat nilai *carry* (C) melimpah yang berarti nilai current lebih besar dari nilai target, maka program akan dinyalakan kipas, setelah mengecek level penyulutan yang digunakan.

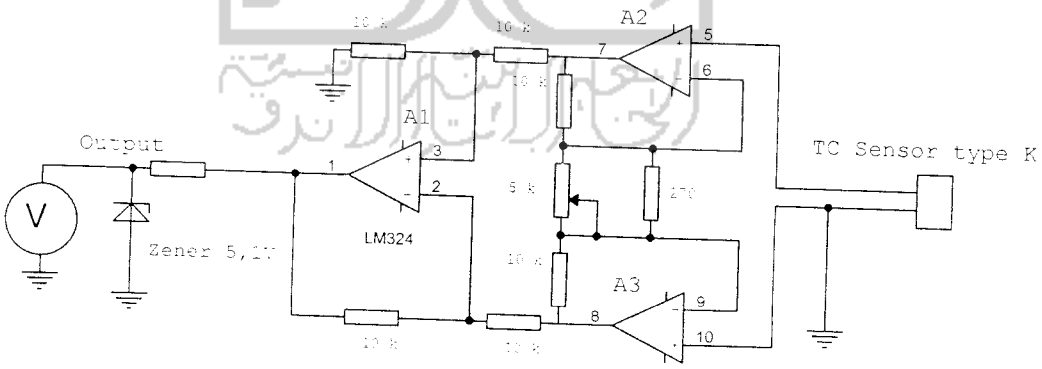
BAB IV

ANALISIS SISTEM ELEKTRONIS

Sebelum alat kendali suhu ini dirangkai utuh dan diuji kinerjanya, penulis melakukan pengujian beberapa rangkaian *hardware* apakah bekerja dengan baik atau tidak. Rangkaian hardware yang diuji antara lain adalah, rangkaian sensor suhu, rangkaian ADC. Sementara untuk *rangkaian fun/heater* dan bagian modul LCD harus diuji setelah semua terpasang karena pengujian bagian tersebut berhubungan dengan Mikrokontroler.

4.1 Pengujian Rangkaian Sensor

Pengujian rangkaian sensor dilakukan adalah dengan mengukur tegangan keluaran dari rangkaian tersebut, sementara pada sensor diberikan perubahan suhu dengan diberikan pemanas didekatnya. Gambar pengukuran dari rangkaian sensor tersebut adalah sebagai berikut :



Gambar 4.1 Pengujian rangkaian sensor suhu

Dari hasil pengujian yang telah dilakukan, didapatkan bahwa terjadi perubahan besaran tegangan pada voltmeter setelah pada sensor didekatkan pemanas, ini membuktikan bahwa rangkaian telah dapat merespon perubahan suhu tersebut. Langkah selanjutnya adalah mengkalibrasi rangkaian ini sehingga didapatkan pembacaan suhu yang benar. Proses kalibrasinya adalah dengan menggunakan termometer kalibrasi, pertama-tama memeriksa suhu yang sedang berlangsung pada termometer kalibrasi, dimisalkan pada termometer kalibrasi tercatat bahwa suhu ada pada 40 ^{0}C. Seperti yang telah disebutkan pada bab tiga bahwa setiap kenaikan level biner dari rangkaian ADC adalah mewakili kenaikan tegangan pada inputnya sebesar 0,97 mV, sedangkan tegangan input dari ADC ini adalah berasal dari rangkaian sensor suhu, dengan demikian dari data-data tersebut diatas, maka pada rangkaian sensor saat itu harus ter*setting* tegangan keluarannya pada tegangan 0,97 dikalikan 40 = 38,8 mV atau 0,038 volt. Penyettingan dapat dilakukan dengan memutar VR 5 KΩ yang ada pada rangkaian sensor suhu, dan harus didapatkan nilai pada voltmeter yang terpasang pada *output* rangkaian sensor suhu sebesar 0,038 volt.

4.2. Pengujian Rangkaian ADC

Pengujian rangkaian ADC meliputi pengukuran catu daya +VCC dan -VCC, pengukuran V*ref* (tegangan referensi) dan *ground*. Hal yang terpenting dari pengukuran ADC adalah pada pengukuran V*ref*, dimana V*ref* sangat berpengaruh besar pada ADC untuk menghasilkan *input* sebesar 4,096 volt. Untuk itu V*ref* yang dibutuhkan adalah 2,04 volt. Pengujian rangkaian ADC dapat ditunjukkan pada Tabel 4.1.

Tabel 4.1 Hasil pengukuran ADC pada rangkaian

Pin	Pengukuran	Tegangan (volt)
40 (+VCC)	1	4,95
	2	4,95
	3	4,95
28 (-VCC)	1	-5,06
	2	-5,06
	3	-5,06
36 (V*ref*)	1	2,04
	2	2,04
	3	2,04
1 (GND)	1	0
	2	0
	3	0

4.3 Analisa Logika *Fuzzy*

Logika *fuzzy* merupakan suatu cara yang tepat untuk memetakan suatu ruang input kedalam suatu ruang output. Pada alat kendali ini diterapkannya logika *fuzzy* dikarenakan konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.

Hasil pembacaan sensor pada alat kendali suhu apabila dibandingkan dengan masukan *setpoint* akan memiliki tiga kemungkinan *error*, yaitu positf, *zero* dan negatif. *Error* negatif terjadi bila masukan *setpoint* melebihi masukan pembacaan sensor (suhu *aktual*), *zero* adalah bila sama dan positif bila masukan *setpoint* ada dibawah suhu *aktual*.

Proses kendali dengan menggunakan logika *fuzzy* dalam alat ini adalah bagaimana memetakan hasil *error* yang diperoleh menjadi level-level yang telah ditentukan, hal ini disebut dengan fuzzifikasi. Proses fuzzifikasi antara pemanas dan

kipas dalam alat yang dibuat ini memiliki perbedaan ketentuan. Dimana perbedaan hanya terletak pada range suhunya karena proses pemanasan dapat berlangsung lebih cepat bila dibandingkan dengan pendinginan.

Setelah proses fuzzifikasi dilakukan didapatkan nilai *error,* maka dibuatlah aturan-aturan yang akan menentukan proses kendali. Aturan-aturan logika *fuzzy* yang akan digunakan sangat bergantung pada sistem yang akan dikendalikan. Dalam logika *fuzzy* tidak terdapat rumusan yang pasti dalam menentukan aturan-aturan *fuzzy* dan fungsi keanggotaan masukan dan keluaran.

Proses defuzzifikasi dimaksudkan untuk memetakan himpunan *fuzzy* hasil inferensi ke himpunan keluaran. Pengendali *fuzzy* harus mengubah variabel keluaran *fuzzy* menjadi nilai-nilai tegas yang dapat digunakan untuk mengendalikan sistem.

4.4 Pengujian Alat Keseluruhan

Setelah semua rangkaian *hardware* dipasang dengan benar dan diberikan catu daya sesuai dengan kebutuhan dari rangkaian masing-masing, langkah selanjutnya adalah menguji kinerja alat kendali suhu yang telah dibuat.

Pada saat pertama kali alat dihidupkan, alat menampilkan suhu *aktual* (suhu yang sedang berlangsung), sementara nilai target masih kosong, kondisi kipas dan pemanas semuanya *off*. Nilai suhu *current* ini ditampilkan oleh LCD pada baris bawah, suhu *aktual* dapat berubah secara alami sesuai dengan kondisi suhu ruang dimana ia ditempatkan. Selanjutnya kemudian ditekan tombol *stop*, yang berarti akan memulai memasukkan nilai target (*setpoint*).

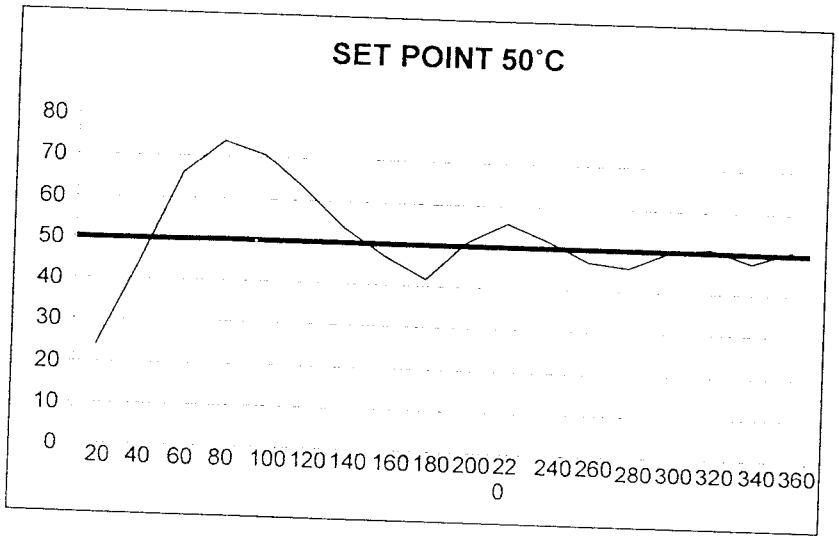
Bila memasukkan nilai target (*setpoint*) diatas nilai suhu *aktual* dan kemudian menekan tombol *start*, maka Mikrokontroler akan memeriksanya pada level mana nilai target (*setpoint*) tersebut menghasilkan selisih dengan nilai suhu *aktual*

(fuzzifikasi) agar dapat ditentukan pada level mana *heater* akan dinyalakan, dan sebaliknya bila dimasukkan nilai target suhu lebih kecil dari pada suhu *aktual*, Mikrokontroler juga akan memeriksanya juga untuk menentukan level *fan* yang akan di nyalakan (defuzzifikasi).

Pada pengamatan pengujian sistem kendali suhu dengan masukan nilai target diatas nilai *aktual* didapat kenyataan bahwa setelah suhu target dan suhu *aktual* bernilai sama, lalu pemanas dimatikan nampak bahwa suhu tetap beranjak naik dari nilai target yang ditentukan, hal ini adalah disebabkan karakteristik dari pemanas, yang tetap memancarkan panas sisa setelah ia dimatikan. Dengan demikian pada alat kendali suhu ini bila masukan target lebih besar dari suhu *aktual*, dapat dipastikan pada akhir nilai sama antara target dan *aktual*, kipas akan bekerja untuk meredam panas sisa dari pemanas.

Sedangkan pada pengamatan pengujian sistem kendali suhu dengan masukan nilai target dibawah nilai *aktual*, setelah nilai target dan *aktual* bernilai sama, kemungkinan selanjutnya yang bekerja ada 2 bisa kipas lagi atau pamanas, hal ini tergantung apakah panas sisa dari pemanas masih ada atau tidak.

Pengujian sistem kendali suhu ini, dilakukan dengan memasukkan nilai setpoint pada 50°C dan 100°C. Pada pengujian sistem ini, terlihat adanya *overshoot* yang terjadi akibat karakteristik pemanas (*heater*). Adapun grafik pengamatan kecepatan pengendalian pada masukan *setpoint* pada 50°C sebagai berikut

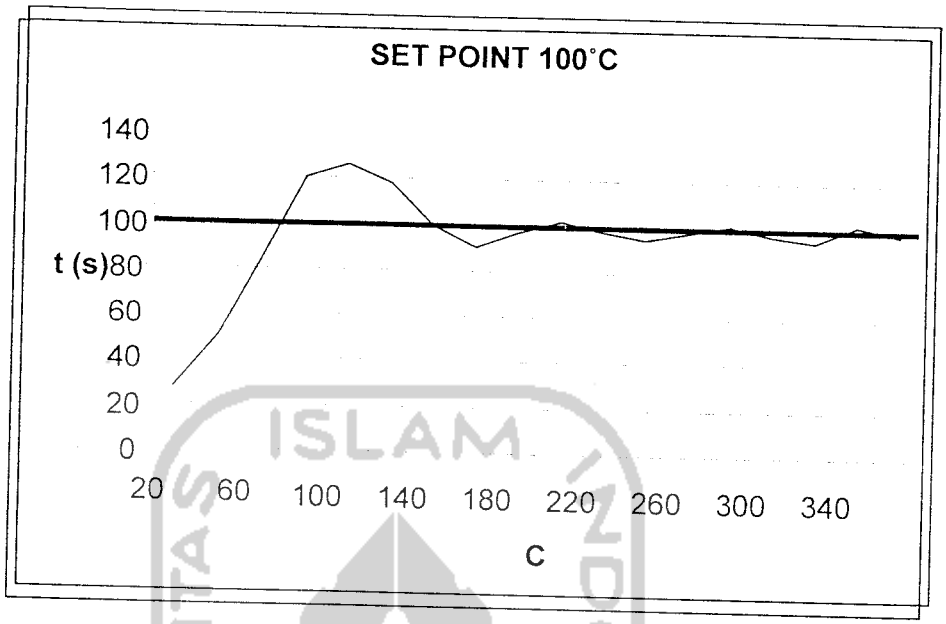


Gambar 4.2 Grafik setpoint pada 50°C

Tabel 4.2 Hasil pengukuran pada setpoint 50°C

°C	t (s)
24	20
44	40
66	60
74	80
71	100
63	120
54	140
47	160
42	180
51	200
56	220
52	240
47	260
46	280
50	300
51	320
48	340
51	360

Sedangkan pada setpoint 100°C. grafik pengamatan pengendalian sistem sebagai berikut :



Gambar 4.3 Grafik setpoint 100°C

Tabel 4.3 Hasil pengukuran pada setpoint 100°C

°C	t (s)
27	20
50	40
84	60
120	80
126	100
118	120
100	140
91	160
97	180
102	200
98	220
95	240
98	260
101	280
97	300
95	320
102	340
98	360

Dari Gambar 4.2 dan Gambar 4.3 dapat dilihat bahwa setelah suhu target yang diinginkan tercapai, maka suhu akan mengalami perubahan (naik dan turun) sampai akhirnya berosilasi untuk proses peredaman panas sisanya..

Proses osilasi (bekerjanya pemanas dan kipas secara bergantian) dari alat ini bekerja pada level rendah, dan range suhu kurang atau lebih dari nilai targetpun sangat kecil sekali. Inilah salah satu keuntungan dari sistem kendali dengan menggunakan logika *fuzzy*, dimana tingkat *error* setelah titik target tercapai relatif kecil, karena sistem penyalaan pemanas maupun kipas tidak langsung pada skala maksimal seperti yang terjadi pada sistem kendali *on / off* biasa.

Hasil pengukuran tegangan pada kipas (*fan*) dan pemanas (*heater*) pada saat pengujian pada setpoint 50°C dan 100°Cdengan asumsi perubahan level pada keanggotaan logika fuzzy sebagai berikut :

Kipas (*fan*)

Panas	= ±	220 V
Hangat	= ±	180 V
Sedang	= ±	150 V
Sejuk	= ±	125V
Dingin	= ±	0V (TRIAC tidak disulut)

Pemanas (*heater*)

Panas	= ±	0 V (TRIAC tidak disulut)
Hangat	= ±	125 V
Sedang	= ±	150 V
Sejuk	= ±	180 V
Dingin	= ±	220 V

BAB V

PENUTUP

5.1 Kesimpulan

Dari keseluruhan kerja yang telah dilakukan mulai dari merancang, merakit dan menguji alat kendali suhu ini, penulis mengambil beberapa kesimpulan sebagai berikut :

1. Alat yang dibuat telah dapat bekerja dengan cukup baik, dimana hal ini dapat dilihat dengan dapatnya alat ini mengendalikan suhu sesuai dengan masukan target yang diinginkan.
2. Osilasi pada alat ini tetap terjadi, dikarenakan suhu yang telah dicapai sesuai target juga akan berubah dengan sendirinya secara alami, dengan demikian selama proses kendali berlangsung pemanas atau kipas akan tetap bergantian bekerja.
3. Dengan menggunakan logika *fuzzy*, sistem kendali suhu ini dapat bekerja dengan lebih baik bila dibandingkan dengan sistem *on/off*, dikarenakan penyalaan suatu sistem pemanas atau kipas dapat diatur sesuai dengan kebutuhan
4. Penggunaan TRIAC memperhalus kerja alat, walaupun tetap terjadi osilasi akibat kondisi lingkungan.

5.2 Saran

Dari hasil perancangan Pengendalian Suhu Benda Konduksi Menggunakan Logika *Fuzzy* ini, para peneliti berikutnya dapat melaksanakan pengembangan dan perbaikan alat agar dapat diperoleh alat yang lebih sempurna terutama dalam hal:

1. Memberi tambahan tampilan petunjuk pada LCD agar *user* lain dapat dengan mudah mengerti bagaimana mengoperasikan alat ini, tanpa harus membaca buku petunjuk yang mungkin ada.
2. Menambah range pada masing-masing pengkondisi suhu (pemanas dan pendingin) agar pengaturan menjadi lebih luas lagi
3. Alat ini dapat bekerja lebih baik bila ditambahkan suatu rangkaian *Zero Crossing* agar respon pada pengaturan menjadi lebih akurat.

DAFTAR PUSTAKA

[KUS04] Kusumawardani, Sri dan Hari Purnomo. *Aplikasi Logika Fuzzy Untuk Pendukung Keputusan.* Yogyakarta: Graha Ilmu, 2004.

[MAL03] Malvino, Albert Paul. *Prinsip-prinsip Elektronika.* Terjemahan Alb Joko Sutoso. Jakarta: Salemba Teknik, 2003.

[MIL72] Millman, Halkias. *Integrated Eledtronics.* International Student Edition. McGraw-Hill&Kasugawa LTD, 1972

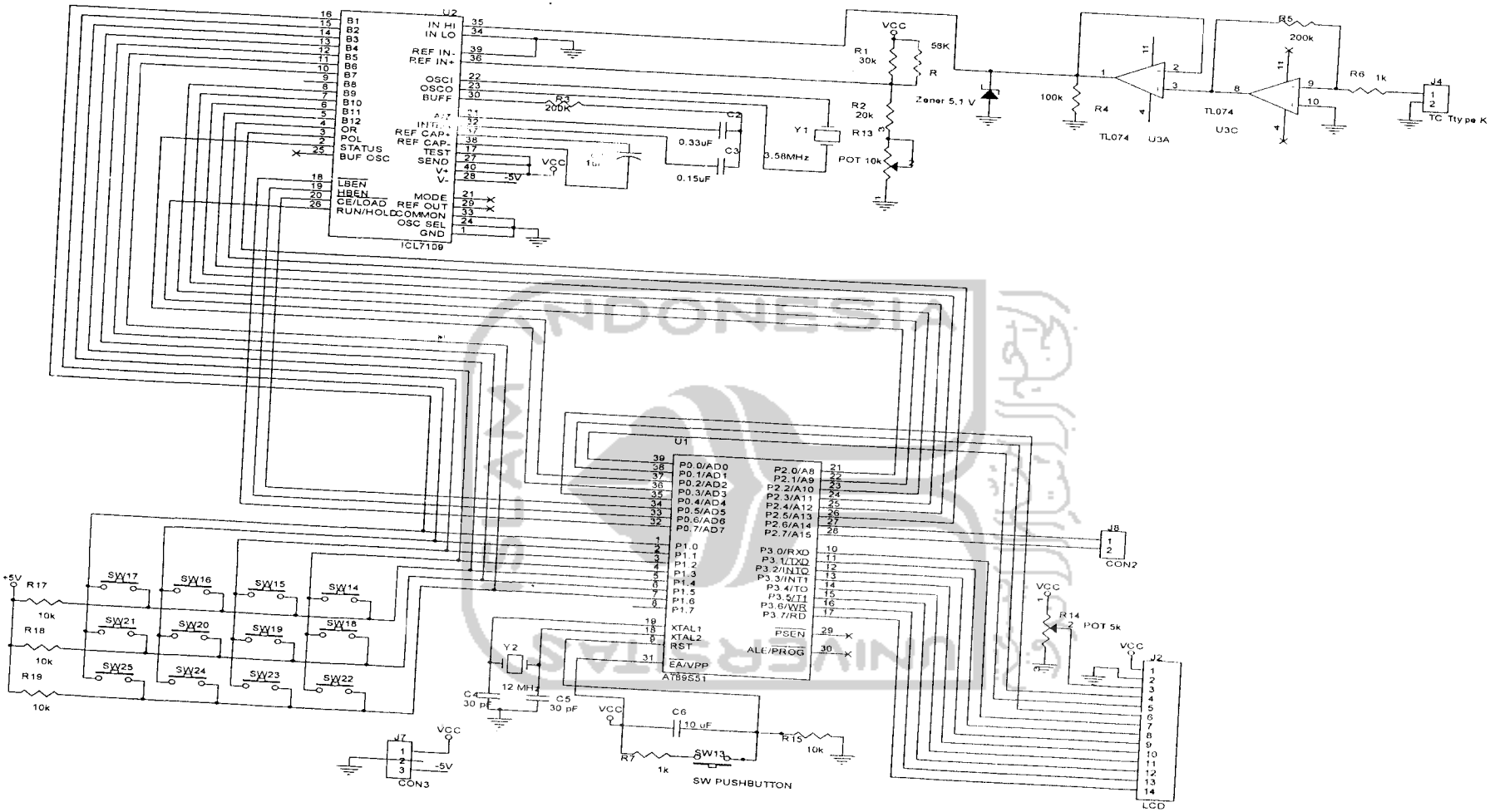
[MIL84] Millman, Jacop. *Mikroelektronika Sistem Digital dan Rangkaian Analog.* Terjemahan Sutanto. Jakarta: Universitas Indonesia, 1984.

[NAL03] Nalwan, Paulus Andi. *Teknik Antarmuka dan Pemrograman Mikrokontroler AT89C51.* Jakarta: Elek Media Komputindo, 2003.

[PUT02] Putra, Agfianto Eko. *Belajar Mikrokontroler AT89C51/52/53.* Jogjakarta: C.V. Gava Media, 2002.

[WAS04] S, Wasito. *Vademekum Elektronika.* Jakarta: Gramedia Pustaka , Utama, 2004

[WOO03] Woollard, Barry G. *Elektronika Praktis.* Terjemahan H. Kristono. Jakarta: Pradnya Paramita, 2003.


U2
ICL7109
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
OR
POL
STATUS
BUF OSC
LBEN
HBEN
CE/LOAD
RUN/HOLD
IN HI
IN LO
REF IN-
REF IN+
OSCI
OSCO
BUFF
INT
REF CAP+
REF CAP-
TEST
SEND
V+
V-
MODE
REF OUT
COMMON
OSC SEL
GND
R3
200K
C2
0.33uF
C3
0.15uF
Y1
3.58MHz
VCC
-5V
R1
30k
58K
R
R2
20k
R13
POT 10k
Zener 5,1 V
100k
R4
TL074
U3A
R5
200k
TL074
U3C
R6
1k
J4
TC Ttype K
U1
AT89S51
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
XTAL1
XTAL2
RST
EA/VPP
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
PSEN
ALE/PROG
Y2
12 MHz
C4
30 pF
C5
30 pF
C6
10 uF
R7
1k
SW13
SW PUSHBUTTON
R15
10k
J8
CON2
R14
POT 5k
J2
LCD
J7
CON3
+5V
R17
R18
R19
10k
SW17
SW16
SW15
SW14
SW21
SW20
SW19
SW18
SW25
SW24
SW23
SW22