



**RANCANG BANGUN MESIN PEMBUAT MINUMAN
KOPI OTOMATIS BERBASIS *MICROCONTROLLER*
DENGAN METODE *FUZZY***

TUGAS AKHIR

**Program Studi
S1 SistemKomputer**

Oleh:

BARRY ADAM MARELLA

09.41020.0063

INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

**RANCANG BANGUN MESIN PEMBUAT MINUMAN
KOPI OTOMATIS BERBASIS *MICROCONTROLLER*
DENGAN METODE *FUZZY***

TUGAS AKHIR



Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer

Oleh :

Nama : Barry Adam Marella

NIM : 09.41020.0063

Program : S1 (Strata Satu)

Jurusan : Sistem Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
INSTITUT BISNIS DAN INFORMATIKA SURABAYA**

2015

Tugas Akhir

RANCANG BANGUN MESIN PEMBUAT MINUMAN KOPI OTOMATIS BERBASIS *MICROCONTROLLER* DENGAN METODE *FUZZY*

dipersiapkan dan disusun oleh

Barry Adam Marella

NIM : 09.41020.0063

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji

pada : 12 Juni 2015

Susunan Dewan Penguji

Pembimbing

I. **Hariato, S.Kom., M.Eng.** _____

II. **Madha Christian Wibowo, S.Kom.** _____

Penguji

I. **Dr. Jusak** _____

II. **Susijanto Tri Resmana, S.Kom., M.T.** _____

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar Sarjana

Dr.Jusak

Dekan Fakultas Teknologi dan Informatika

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

PERNYATAAN

Dengan ini menyatakan dengan benar, bahwa Tugas Akhir ini adalah asli karya saya, bukan plagiat baik sebagian maupun keseluruhan. Karya atau pendapat orang lain yang ada dalam tugas akhir ini adalah semata hanya rujukan yang dicantumkan dalam daftar pustaka saya. Apabila dikemudian hari ditemukan adanya tindakan plagiat pada karya tugas akhir ini, maka saya bersedia untuk dilakukan pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.



Surabaya, 26 Juni 2015
INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

Barry Adam Marella

NIM : 09.41020.0063

ABSTRAK

Mayoritas penduduk Indonesia menyukai minuman kopi khususnya ketika di warung kopi memesan kopi tubruk. Akan tetapi untuk proses pembuatan minuman kopi masih manual dalam proses menakar gula maupun kopi dan proses menyiapkan air untuk menyeduh kopi masih perlunya memasak air yang membutuhkan waktu lama. Selain itu penyimpanan air dalam termos juga yang berpengaruh terhadap suhu air, dimana lama-kelamaan suhu air juga akan semakin turun dan akan berpengaruh saat akan digunakan untuk menyeduh kopi. Dalam penelitian ini dibuatlah mesin pembuat minuman kopi tubruk sehingga dalam proses pembuatannya menjadi lebih praktis. Metode yang digunakan dalam penelitian ini yaitu metode *fuzzy* Sugeno. Metode ini digunakan untuk proses memanaskan dan mengatur kestabilan suhu air sesuai dengan set point yang diinginkan. Dengan metode *fuzzy* ini air mampu dipanaskan hingga mendidih dan suhu air dibuat stabil sesuai dengan set point yang diinginkan. Rata-rata waktu yang dibutuhkan untuk menakar kopi dan gula adalah 17,8 detik/4 gram dan 8,2 detik/4 gram.

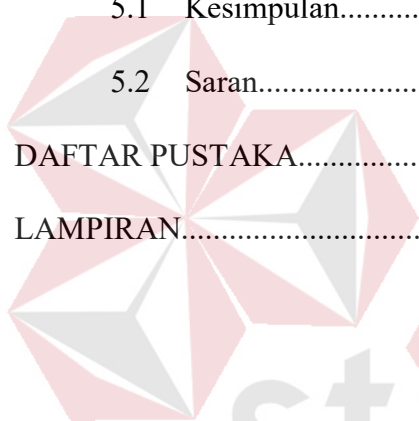
DAFTAR ISI

	Halaman
ABSTRAK.....	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
BAB I.PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan.....	2
1.5. Kontribusi.....	3
1.6. Sistematika Penulisan.....	3
BAB II.LANDASAN TEORI.....	5
2.1. Kopi.....	5
2.1.1. Proses Pembuatan Minuman Kopi.....	5
2.2. Sensor Suhu.....	6
2.3. LCD.....	8
2.4. Relay.....	10
2.4.1. Prinsip Kerja Relay.....	11
2.4.2. Pole dan Throw.....	13
2.4.3. Fungsi Relay.....	15
2.5. Fuzzy.....	16

2.5.1.	Himpunan Fuzzy.....	18
2.5.2.	Fungsi Keanggotaan.....	18
2.5.3.	Operator Dasar.....	18
2.5.4.	Metode Sugeno.....	19
2.6.	Microcontroller ATMega16.....	20
2.6.1.	Konfigurasi Pin AVR ATMega16.....	22
2.6.2.	Peta Memoi AVR Atmega16.....	24
2.6.3.	ADC (Analog to Digital Converter).....	29
2.7.	Motor DC.....	32
2.7.1.	Pengertian Motor DC.....	32
2.7.2.	Prinsip Dasar Cara Kerja.....	33
2.7.3.	Prinsip Arah Putaran Motor.....	37
BAB III.METODE PENELITIAN.....		38
3.1	Perancangan Perangkat Keras.....	38
3.1.1	Rangkaian Minimum sistem.....	40
3.1.2	Rangkaian Relay.....	42
3.1.3	Rangkaian Pull Up Push Button.....	43
3.1.4	Rangkaian LCD.....	43
3.1.5	Rangkaian Sensor Suhu.....	45
3.1.6	Rangkaian Keseluruhan.....	46
3.2	Perancangan Mekanik.....	47
3.3	Perancangan Perangkat Lunak.....	50
3.3.1	Perancangan Perangkat Lunak Untuk Sensor Suhu.....	52
3.3.2	Fuzzy.....	53

BAB IV. PENGUJIAN SISTEM.....	65
4.1 Pengujian Minimum System.....	65
4.1.1 Tujuan.....	65
4.1.2 Alat Yang Digunakan.....	65
4.1.3 Prosedur Pengujian.....	65
4.1.4 Hasil Pengujian.....	66
4.2 Pengujian Sensor Suhu.....	67
4.2.1 Tujuan.....	67
4.2.2 Alat Yang Digunakan.....	67
4.2.3 Prosedur Pengujian.....	67
4.2.4 Hasil Pengujian.....	68
4.3 Pengujian Motor Penuang Gula Dan Kopi.....	70
4.3.1 Tujuan.....	70
4.3.2 Alat Yang Digunakan.....	71
4.3.3 Prosedur Pengujian.....	71
4.3.4 Hasil Pengujian.....	71
4.4 Pengujian Push Button.....	72
4.4.1 Tujuan.....	72
4.4.2 Alat Yang Digunakan.....	72
4.4.3 Prosedur Pengujian.....	72
4.4.4 Hasil Pengujian.....	73
4.5 Pengujian LCD.....	75
4.5.1 Tujuan.....	75
4.5.2 Alat Yang Digunakan.....	75

4.5.3	Prosedur Pengujian.....	76
4.5.4	Hasil Pengujian.....	76
4.6	Pengujian Metode Fuzzy.....	77
4.6.1	Tujuan.....	77
4.6.2	Alat Yang Digunakan.....	77
4.6.3	Prosedur Pengujian.....	78
4.6.4	Hasil Pengujian.....	78
BAB V. PENUTUP.....		84
5.1	Kesimpulan.....	84
5.2	Saran.....	84
DAFTAR PUSTAKA.....		85
LAMPIRAN.....		86



INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

DAFTAR TABEL

	Halaman
Tabel 2.1 Fungsi Khusus Port B.....	23
Tabel 2.2 Fungsi Khusus Port C.....	23
Tabel 2.3 Fungsi Khusus Port D.....	24
Tabel 2.4 Pengalamatan Register I/O.....	27
Tabel 2.5 Konfigurasi Clock ADC.....	30
Tabel 2.6 Pemilihan Sumber Picu ADC.....	32
Tabel 3.1 Penentuan <i>Rule Set</i>	58
Tabel 4.1 Hasil Pengukuran Sensor Suhu.....	69
Tabel 4.2 Hasil Pengujian Kopi.....	71
Tabel 4.3 Hasil Pengujian Gula.....	72
Tabel 4.4 Hasil Pengujian Tombol Menu Kopi.....	74
Tabel 4.5 Hasil Pengujian Tombol Menu Gula.....	75
Tabel 4.6 Konfigurasi LCD 16x2 dengan Minimum System ATmega16.....	76
Tabel 4.7 Tabel Pengujian <i>Fuzzy</i> dengan Setpoint 85°C.....	80
Tabel 4.8 Tabel Pengujian <i>Fuzzy</i> dengan Setpoint 90°C.....	81

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Sensor Suhu LM35.....	7
Gambar 2.2 Pin LCD.....	9
Gambar 2.3 Bentuk Relay.....	11
Gambar 2.4 Simbol Relay.....	11
Gambar 2.5 Struktur Sederhana Relay.....	12
Gambar 2.6 <i>Single Pole Single Throw</i>	14
Gambar 2.7 <i>Single Pole Double Throw</i>	14
Gambar 2.8 <i>Double Pole Single Throw</i>	14
Gambar 2.9 <i>Double Pole Double Throw</i>	14
Gambar 2.10 Simbol Push Button.....	15
Gambar 2.11 Push Button.....	16
Gambar 2.12 Contoh Pemetaan Input-Output.....	17
Gambar 2.13 Konfigurasi Pin ATmega16.....	22
Gambar 2.14 Peta Memori Program AVR ATmega16.....	25
Gambar 2.15 Peta Memori Data AVR ATmega16.....	26
Gambar 2.16 Status Register ATmega16.....	28
Gambar 2.17 Adc Control Dan Status Register A – ADCSRA.....	29
Gambar 2.18 Adc Multiplexer.....	30
Gambar 2.19 ADLAR.....	31
Gambar 2.20 register SFIOR.....	31
Gambar 2.21 motor ADC sederhana.....	33
Gambar 2.22 medan magnet yang membawa arus mengelilingi konduktor.....	33

Gambar 2.23 Konduktor Berbentuk U.....	34
Gambar 2.24 Reaksi Garis Fluks.....	34
Gambar 2.25 Prinsip Kerja Motor DC.....	36
Gambar 3.1 Diagram Blok Sistem.....	39
Gambar 3.2 Rangkaian Minimum Sistem.....	40
Gambar 3.3 Rangkaian Reset.....	41
Gambar 3.4 Rangkaian Crystal.....	41
Gambar 3.5 Rangkaian Driver Relay Motor Penuang Dan Washer Pump.....	42
Gambar 3.6 Rangkaian Driver Relay Heater.....	42
Gambar 3.7 Rangkaian Pull Up Push Button.....	43
Gambar 3.8 Rangkaian Lcd.....	44
Gambar 3.9 Rangkaian Sensor Suhu.....	45
Gambar 3.10 Rangkaian Keseluruhan Sistem.....	46
Gambar 3.11 Mekanik Tampak Depan.....	47
Gambar 3.12 Mekanik Tampak Atas.....	47
Gambar 3.13 Penempatan Heater.....	48
Gambar 3.14 Penempatan Sensor Suhu.....	48
Gambar 3.15 Penempatan Washer Pump.....	48
Gambar 3.16 Mekanik Penuang Gula Dan Kopi.....	49
Gambar 3.17 Posisi Tempat Gula Dan Kopi.....	49
Gambar 3.18 Diagram Alir Program Secara Umum Pada Microcontroller.....	51
Gambar 3.19 Diagram Alir Pembacaan Sensor Suhu.....	52
Gambar 3.20 Pemetaan Derajat Keanggotaan Error.....	53
Gambar 3.21 Pemetaan Derajat Keanggotaan Delta Error.....	53

Gambar 3.22 Flowchart Proses Fuzzifikasi Dalam Memanaskan Dan Menstabilkan Suhu Air.....	54
Gambar 3.23 Flowchart Proses Rule Set.....	60
Gambar 3.24 Fungsi Keanggotaan Singleton.....	61
Gambar 3.25 Flowchart Proses Defuzzifikasi.....	62
Gambar 4.1 Tampilan <i>Chip Signature</i>	66
Gambar 4.2 Tampilan Download Program.....	67
Gambar 4.3 Grafik Hasil Pengujian Sensor Suhu.....	70
Gambar 4.4 Tampilan LCD.....	77
Gambar 4.5 Hasil Pengujian Saat Suhu 29°C.....	78
Gambar 4.6 Grafik Hasil Pengujian <i>Fuzzy</i> Dengan Setpoint 85°C.....	79
Gambar 4.7 Grafik Hasil Pengujian <i>Fuzzy</i> Dengan Setpoint 90°C.....	82
Gambar 4.8 Tampilan Alat Sebenarnya.....	83
Gambar 4.9 Tampilan Alat Tampak Depan.....	83

BAB I

PENDAHULUAN

1.1 Latar Belakang

Mayoritas penduduk Indonesia menyukai minuman kopi khususnya ketika di warung kopi memesan kopi tubruk. Akan tetapi untuk proses pembuatan minuman kopi masih manual dalam proses menakar gula maupun kopi dan proses menyiapkan air untuk menyeduh kopi masih perlunya memasak air yang membutuhkan waktu lama. Selain itu penyimpanan air dalam termos juga yang berpengaruh terhadap suhu air, dimana lama-kelamaan suhu air juga akan semakin turun dan akan berpengaruh saat akan digunakan untuk menyeduh kopi.

Oleh karena itu dibuatlah sebuah mesin khususnya untuk membuat minuman kopi tubruk. Selain itu yang dapat selalu menjaga air agar tetap mendidih untuk membuat minuman kopi secara otomatis berbasis *microcontroller*. Dimana alat ini dapat membuat suhu air menjadi stabil sesuai dengan dengan suhu air mendidih sehingga membuat kopi menjadi lebih mudah dan cepat tanpa perlu memasak air terlebih dahulu. Selain itu rasa dan aroma yang didapat lebih enak dibandingkan dengan air yang kurang mendidih.

Mesin pembuat kopi otomatis berbasis *microcontroller* ini selain dapat membuat suhu air menjadi lebih stabil, mesin ini juga dilengkapi dengan tombol untuk memudahkan memilih takaran kopi dan gula sesuai dengan keinginan *user*, sehingga *user* dapat memilih banyaknya takaran dari kopi dan gula yang diinginkan dalam satu gelas kopi yang dibuat.

Pengaturan pemanas air dilakukan dengan proses fuzzifikasi, penentuan *rule set*, dan defuzzifikasi. Berdasarkan pada nilai *error* dan delta *error* yang terdeteksi oleh sensor suhu lm35, jika *error* dan delta *error* lebih menuju arah negatif maka jumlah pemanas yang digunakan semakin berkurang. Sebaliknya jika nilai *error* dan delta *error* lebih menuju arah positif maka jumlah pemanas yang digunakan semakin bertambah.

1.2 Perumusan Masalah

Dari masalah yang telah diuraikan di atas, maka dapat dirumuskan permasalahan yang dihadapi adalah:

1. Bagaimana sistem mampu mengatur suhu air agar dapat stabil sesuai dengan set point yang diinginkan.
2. Bagaimana sistem mampu menuangkan kopi dan gula sesuai dengan takaran yang diminta *user*.

1.3 Pembatasan Masalah

Dalam perancangan dan pembuatan alat ini, terdapat beberapa pembatasan masalah, antara lain:

1. Semua bahan kopi dan gula dalam bentuk serbuk.
2. Pengadukan setelah minuman kopi selesai dibuat oleh mesin harus dilakukan secara manual.

1.4 Tujuan

Adapun tujuan dari perancangan dan pembuatan perangkat ini diantaranya yaitu:

1. Alat yang dibuat dapat mengatur suhu air agar dapat stabil sesuai dengan set point yang diinginkan.

2. Alat yang dibuat dapat menuangkan kopi dan gula sesuai dengan takaran yang diminta *user*.

1.5 Kontribusi

Dengan adanya Tugas Akhir ini diharapkan dapat mengatur pemanasan air sehingga didapatkan suhu air yang stabil dan menghasilkan takaran gula dan kopi yang sesuai dengan takaran yang diinginkan sehingga dapat memudahkan dalam membuat minuman kopi.

1.6 Sistematika Penulisan

Penulisan tugas akhir ini secara sistematis diatur dan disusun dalam lima bab yang didalamnya terdapat beberapa sub bab. Secara ringkas uraian materi dari bab pertama hingga bab terakhir adalah sebagai berikut:

BAB I : Pendahuluan

Pada bab pendahuluan ini dibahas mengenai latar belakang masalah, perumusan masalah, batasan masalah, tujuan, kontribusi serta sistematika dari penulisan tugas akhir.

BAB II : Landasan Teori

Pada bab landasan teori ini dijelaskan tentang Sistem Fuzzy, microcontroller ATmega16, sensor suhu, relay, motor DC 12V dan lcd.

BAB III : Metode Penelitian

Pada bab ini dibahas mengenai perangkat keras (*hardware*) dan perangkat lunak (*software*). *Hardware* meliputi minimum system, sensor suhu, relay, dan lcd. Sedangkan untuk perangkat lunaknya (*software*) menggunakan bahasa C dan metode *fuzzy* yang digunakan adalah metode Sugeno.

BAB IV : Pengujian Sistem

Pada bab ini berisi tentang pengujian jalannya sistem fuzzy dalam proses memanaskan air dan menstabilkan suhu air sesuai dengan set point yang diinginkan.

BAB V : Kesimpulan dan Saran

Pada bab ini diberikan kesimpulan dari hasil pengujian sistem secara keseluruhan dan saran-saran yang diharapkan dalam pengembangan lebih lanjut dari tugas akhir ini.



BAB II

LANDASAN TEORI

2.1 Kopi

Kopi mulai dikenal sejak abad ke-7 dan telah tumbuh di dekat Laut Merah. Pada abad ke 16 kopi ditemukan di daratan Afrika, yaitu di Yaman dan Ethiopia. Tahun 1669, kopi mulai dikenalkan ke bangsa Eropa dan beberapa tahun kemudian orang-orang Belanda memperkenalkan kopi ke pulau Jawa. Kopi arabika dikenal sejak abad ke-13, sedang kopi robusta dikenal baru akhir abad ke-19.

Secara taksonomi, kopi termasuk famili *Rubiaceae*, genus *coffea*. Spesiesnya ada 2 yaitu *coffea arabica* (arabika) dan *coffea canephora* (robusta). Kopi robusta tumbuh di tempat yang berbeda dengan kopi arabika, yaitu didaerah tropis dimana kopi arabika tidak bisa tumbuh. Harga kopi robusta lebih murah daripada kopi arabika namun rasanya kurang enak/ lebih pahit bila dibanding kopi arabika.

Menurut beberapa hasil penelitian, mengkonsumsi kopi dapat menurunkan insidens dari berbagai macam penyakit diantaranya diabetes mellitus tipe 2, kardiovaskuler, kanker, serta menurunkan kadar asam urat (Lelyana, 2008).

2.1.1 Proses Pembuatan Minuman Kopi

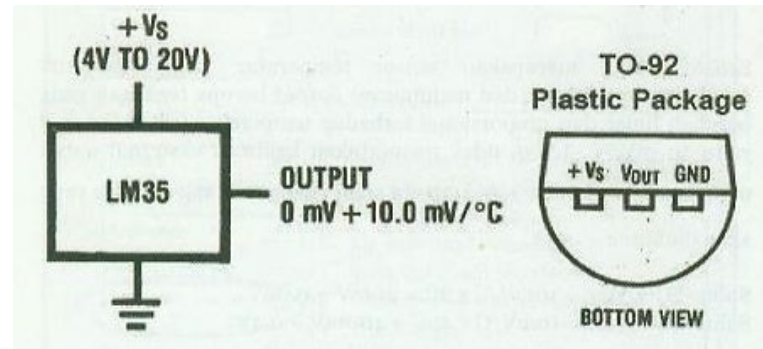
Dalam proses penyeduhan kopi agar kopi yang diminum terasa harum dan nikmat terdapat beberapa tahapan proses untuk membuat kopi dimana proses yang dilakukan ini adalah proses dalam membuat minuman kopi tubruk. Untuk tahap pertama tuangkan bubuk kopi kedalam gelas sesuai takaran yang akan diminum.

Untuk satu cangkir gunakan dua sendok makan bubuk kopi. Untuk proses selanjutnya ini adalah hal yang perlu diperhatikan yaitu memasak air hingga mendidih. Setelah air mendidih kemudian diamkan air selama satu menit. Hal ini dilakukan untuk menyesuaikan temperatur air yang pas untuk bubuk kopi. Dan temperatur ideal menurut Dimas sebagai barista yaitu berada di suhu 85 Derajat Celcius. Setelah satu menit, tuangkan air panas secara perlahan-lahan ke cangkir kopi hingga penuh. Tunggu sekitar 4 menit, hal ini agar bubuk kopi mengendap dan turun kebawah. Setelah 4 menit bubuk kopi akan mengendap dibawah setelah itu kopi sudah siap untuk diminum. (Dimas,2013).

2.2 Sensor Suhu

Sensor suhu LM35 adalah komponen elektronika yang memiliki fungsi untuk mengubah besaran suhu menjadi besaran listrik dalam bentuk tegangan (Suyadhi, 2010). Sensor LM35 memiliki keakuratan tinggi dan kemudahan perancangan jika dibandingkan dengan sensor suhu yang lain. LM35 juga mempunyai keluaran impedans yang rendah dan linearitas yang tinggi sehingga dapat dengan mudah dihubungkan dengan rangkaian kendali khusus serta tidak memerlukan penyetelan lanjutan.

Meskipun tegangan sensor ini dapat mencapai 30 volt, akan tetapi tegangan yang diberikan pada sensor hanya sebesar 5 volt sehingga sensor ini dapat digunakan dengan catu daya tunggal dengan ketentuan bahwa LM35 hanya membutuhkan arus sebesar 60 μ A. Hal ini berarti LM35 mempunyai kemampuan menghasilkan panas (self-heating) dari sensor yang dapat menyebabkan kesalahan pembacaan yang rendah, yaitu kurang dari 0,5°C pada suhu 25°C.



Gambar 2.1 Sensor suhu LM35 (Andrianto, 2013).

LM35 hanya memiliki tiga kaki yang berfungsi dalam penggunaannya, yaitu (1) pin V_s atau tegangan kerja LM35, (2) pin V_{out} atau tegangan keluaran dengan jangkauan kerja dari 0 volt sampai 1,5 volt pada tegangan operasi sensor LM35 antara 4 volt sampai 30 volt, dan (3) pin GND atau ground untuk tegangan bumi/negatif.

Keluaran sensor ini akan naik sebesar 10 mV setiap derajat *Celsius* sehingga persamaan berikut bisa diperoleh.

$$V_{LM35} = \text{Suhu} \times 10\text{mV} \dots \dots \dots (2.1)$$

Secara prinsip, sensor akan melakukan pengindraan pada saat perubahan suhu setiap 1°C akan menunjukkan tegangan sebesar 10 mV. Pada penempatannya, LM35 dapat ditempelkan dengan perekat atau dapat pula disemen pada permukaan, akan tetapi suhunya akan sedikit berkurang sekitar $0,01^\circ\text{C}$ karena terserap suhu permukaan. Dengan cara seperti ini diharapkan selisih antara suhu udara dan suhu permukaan yang dapat dideteksi oleh sensor LM35 sama dengan suhu di sekitarnya. Jika suhu udara di sekitarnya jauh lebih tinggi atau jauh lebih rendah dari suhu permukaan, maka LM35 berada pada suhu

permukaan dan suhu udara di sekitarnya. Berikut adalah karakteristik sensor LM35.

1. Memiliki sensitivitas suhu, dengan faktor skala linear antara tegangan dan suhu $10 \text{ mV}/^\circ\text{C}$ sehingga dapat dikalibrasi langsung dalam *Celcius*.
2. Memiliki ketepatan atau akurasi kalibrasi, yaitu $0,5^\circ\text{C}$ pada suhu 5°C .
3. Memiliki jangkauan maksimal operasi! Suhu antara -55°C sampai $+150^\circ\text{C}$.
4. Bekerja pada tegangan 4 sampai 30 volt.
5. Memiliki arus rendah, yaitu kurang dari $60 \mu\text{A}$.
6. Memiliki pemanasan sendiri yang rendah (*low-heating*), yaitu kurang dari $0,1^\circ\text{C}$ pada udara diam.
7. Memiliki impedansi keluaran yang rendah, yaitu $0,1 \text{ W}$ untuk beban 1 mA .
8. Memiliki ketidaklinearan. Hanya sekitar $\pm 1/4^\circ\text{C}$.

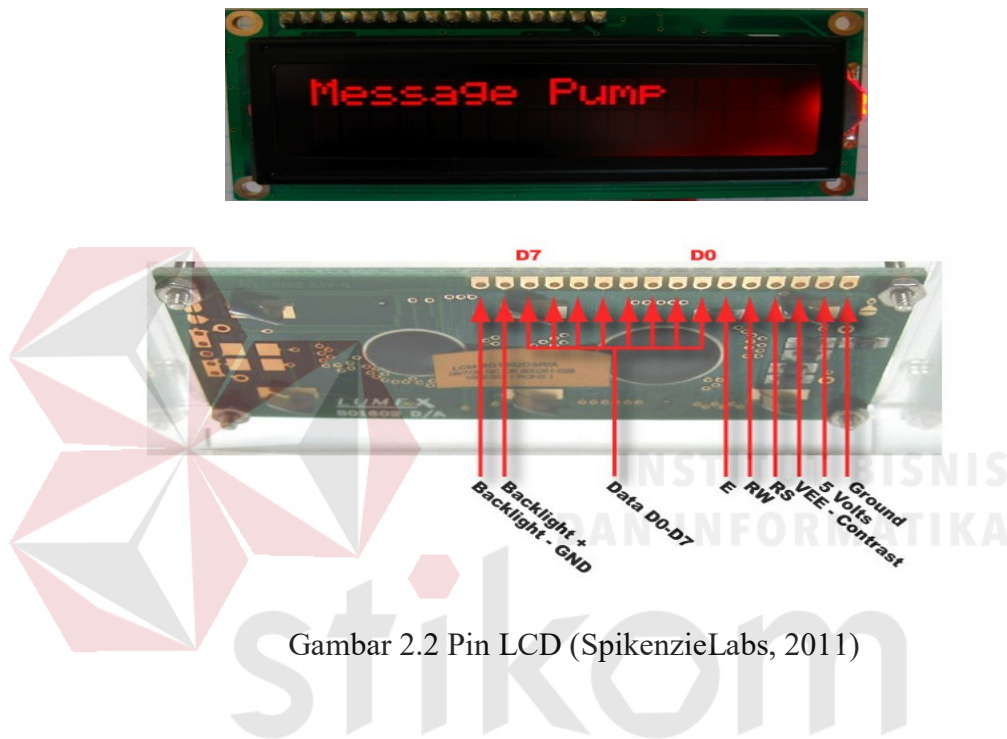
2.3 LCD

LCD (*Liquid Cristal Display*) berfungsi untuk menampilkan karakter angka, huruf ataupun simbol dengan lebih baik dan dengan konsumsi arus yang rendah. Berikut ini adalah karakteristik dari modul LCD yaitu sebagai berikut :

- Terdapat 16×2 karakter huruf yang 80ca ditampilkan.
- Setiap huruf terdiri dari 5×7 dot-matrix cursor.
- Terdapat 192 macam karakter.
- Terdapat 80×8 bit display RAM (maksimal 80 karakter).

- Memiliki kemampuan penulisan dengan 8 bit maupun dengan 4 bit.
- Dibangun dengan osilator 90kHz.
- Satu sumber tegangan 5 volt.
- Otomatis reset saat tegangan dihidupkan.

Gambar di bawah ini merupakan Konfigurasi pin dari LCD yaitu :



Gambar 2.2 Pin LCD (SpikenzieLabs, 2011)

Adapun fungsi Pin LCD yang ditunjukkan oleh Gambar 2.2 yaitu:

- DB0 – DB7 adalah jalur data (data bus) yang berfungsi sebagai jalur komunikasi untuk mengirimkan dan menerima data atau instruksi dari mikrokontroler ke modul LCD
- RS adalah pin yang berfungsi sebagai selektor register (*register select*) yaitu dengan memberikan logika *low* (0) sebagai register perintah dan logika *high* (1) sebagai register data.

- R/W adalah pin yang berfungsi untuk menentukan mode baca atau tulis dari data yang terdapat pada DB0 – DB7. Yaitu dengan cara memberikan logika *low* (0) untuk fungsi *read* dan logika *high* (1) untuk mode *write*.
- Enable (E), berfungsi sebagai *enable Clock* LCD, logika 1 setiap kali pengiriman atau pembacaan data.

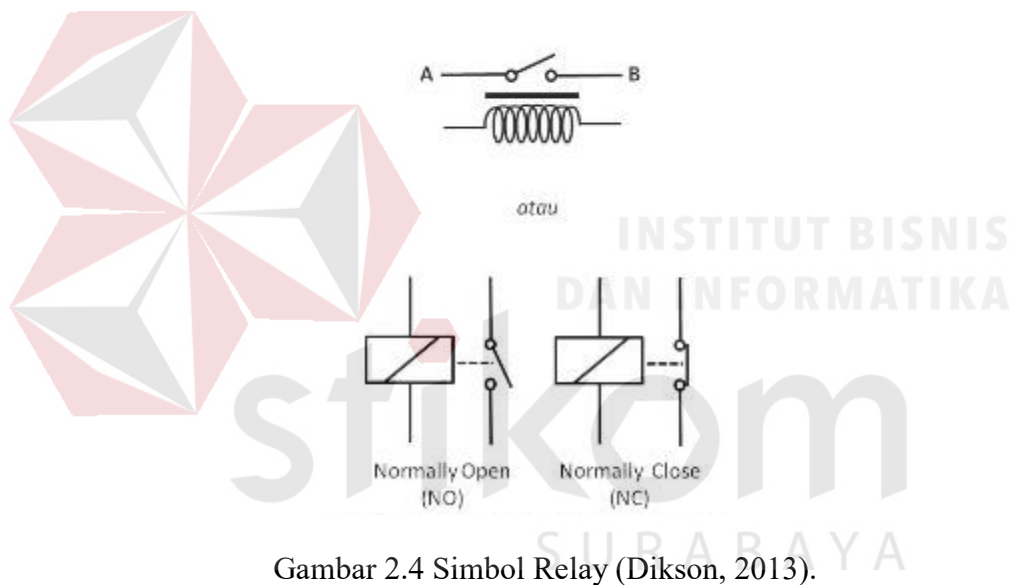
Modul LCD memiliki 3 jalur kontrol yang bernama RS, R/W, dan E. RS digunakan untuk memberitahukan kepada LCD apakah data yang diberikan adalah kata intruksi (*intruction words*) atau kata-kata data (*data word*). Jika akan mengirim intruksi, RS harus di buat 0, sedangkan untuk mengirimkan data RS harus berlogika 1. Jalur R/W digunakan untuk memilih operasi *Read* atau *Write*. *Read* artinya membaca data dari LCR, sedangkan *Write* menuliskan data ke LCD. Dalam kasus ini, kita hanya menuliskan data ke LCD sehingga jalur ini harus selalu dibuat rendah (Logika 0). Terakhir adalah jalur E (*Enable*). Jika jalur ini berlogika tinggi (1), proses penulisan ke LCD akan diaktifkan (Malik dan Juwana, 2009).

2.4 Relay

Relay adalah saklar (*switch*) yang dioperasikan secara listrik dan merupakan komponen elektromekanikal yang terdiri dari 2 bagian utama yakni Elektromagnet (Coil) dan Mekanikal (seperangkat kontak Saklar/switch). Relay menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang beregangan lebih tinggi.



Gambar 2.3 Bentuk Relay (Dikson, 2013).



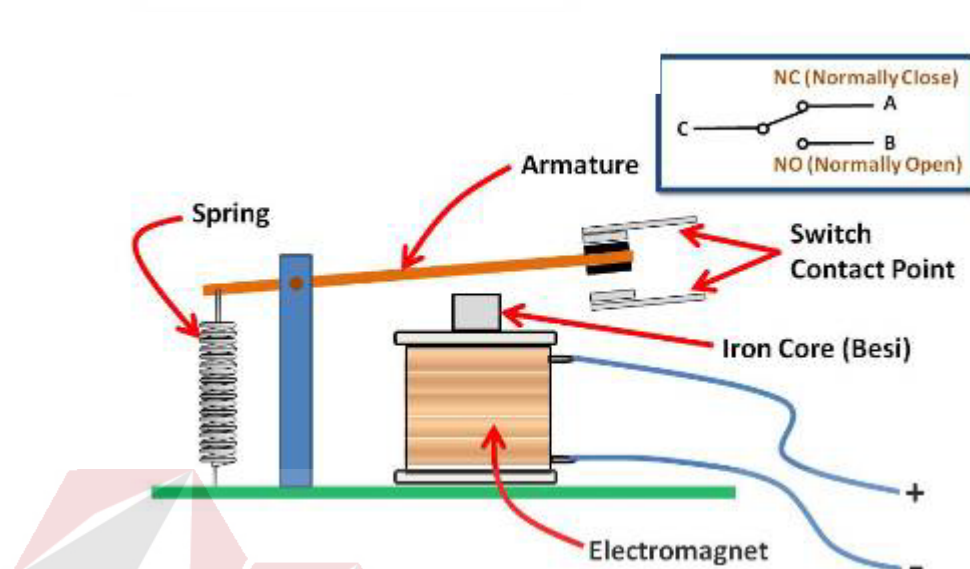
Gambar 2.4 Simbol Relay (Dikson, 2013).

2.4.1 Prinsip kerja Relay

Pada dasarnya dalam sebuah relay sederhana terdiri dari 4 komponen dasar yaitu:

1. *Electromagnet (Coil)*
2. *Armature.*
3. *Switch Contact Point (Saklar).*
4. *Spring.*

Berikut ini merupakan gambar dari bagian-bagian relay (struktur sederhana sebuah relay):



Gambar 2.5 Struktur sederhana relay (Dikson, 2013).

Kontak Point (*Contact Point*) relay terdiri dari 2 jenis yaitu:

- *Normally Close* (NC) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi CLOSE (tertutup).
- *Normally Open* (NO) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi OPEN (terbuka).

Berdasarkan Gambar 2.5, sebuah besi (*Iron Core*) yang dililit oleh kumparan *coil* yang berfungsi untuk mengendalikan besi tersebut. Apabila kumparan *coil* diberikan arus listrik, maka akan timbul gaya elektromagnet yang kemudian menarik *armature* untuk berpindah dari posisi sebelumnya (NC) ke posisi baru (NO) sehingga menjadi saklar yang dapat menghantarkan arus listrik di posisi barunya (NO). Posisi dimana *armature* tersebut berada sebelumnya (NC) akan menjadi OPEN atau tidak terhubung. Pada saat tidak dialiri arus listrik,

armature akan kembali lagi ke posisi Awal (NC). *Coil* membutuhkan arus listrik yang relatif untuk mengaktifkan electromagnet dan menarik *Contact Poin* ke posisi Close.

2.4.2 Pole dan Throw

Karena relay merupakan salah satu jenis saklar, maka istilah *pole* dan *throw* yang dipakai dalam saklar juga berlaku pada *relay*. Berikut ini adalah penjelasan singkat mengenai istilah Pole dan Throw:

Pole : Banyaknya kontak (*Contact*) yang dimiliki oleh sebuah *relay*.

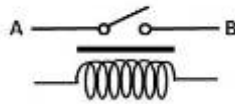
Throw : Banyaknya kondisi yang dimiliki oleh sebuah Kontak (*Contact*).

Berdasarkan penggolongan jumlah *pole* and *throw*-nya sebuah relay, maka relay dapat digolongkan menjadi:

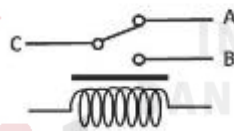
1. Single Pole Single Throw (SPST): *Relay* golongan ini memiliki 4 Terminal, diantaranya 2 Terminal untuk saklar dan 2 Terminalnya lagi untuk *Coil*.
2. Single Pole Double Throw (SPDT): *Relay* golongan ini memiliki 5 Terminal, diantaranya 3 Terminal untuk saklar dan 2 Terminalnya lagi untuk *Coil*.
3. Double Pole Single Throw (DPST): *Relay* golongan ini memiliki 6 terminal, diantaranya 4 Terminal yang terdiri dari 2 pasang Terminal Saklar sedangkan 2 Terminal lainnya untuk *Coil*. *Relay* DPST dapat dijadikan 2 Saklar yang dikendalikan oleh 1 *Coil*.
4. Double Pole Double Throw (DPDT): *Relay* golongan ini memiliki Terminal sebanyak 8, diantaranya 6 Terminal yang merupakan 2 pasang

Relay SPDT yang dikendalikan oleh 1 (single) *Coil*. Sedangkan 2 Terminalnya lainnya untuk *Coil*.

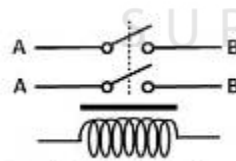
Selain Golongan *Relay* diatas, terdapat juga *Relay-Relay* yang Pole dan Thrownya melebihi dari 2 (dua). Misalnya 3PDT (Triple Pole Double Throw) ataupun 4PDT (Four Pole Double Throw) dan lain sebagainya (Dikson, 2013).



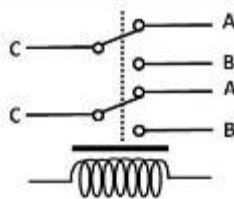
Gambar 2.6 Single Pole Single Throw (SPST) (Dikson, 2013).



Gambar 2.7 Single Pole Double Throw (SPDT) (Dikson, 2013).



Gambar 2.8 Double Pole Single Throw (DPST) (Dikson, 2013).



Gambar 2.9 Double Pole Double Throw (DPDT) (Dikson, 2013).

2.4.3 Fungsi Relay

Beberapa fungsi Relay yang telah umum diaplikasikan ke dalam peralatan Elektronika diantaranya adalah

1. *Relay* digunakan untuk menjalankan fungsi logika (*logic function*).
2. *Relay* digunakan untuk memberikan fungsi penundaan waktu (*time delay function*).
3. *Relay* digunakan untuk mengendalikan sirkuit tegangan tinggi dengan bantuan dari signal tegangan rendah.
4. Ada juga relay yang berfungsi untuk melindungi motor atau komponen lainnya dari kelebihan tegangan ataupun hubung singkat (*short*).

2.5 Saklar

Pada umumnya saklar *push button* adalah tipe saklar yang hanya kontak sesaat saja saat ditekan dan setelah dilepas maka akan kembali lagi menjadi NO, biasanya saklar tipe NO ini memiliki rangkaian penguncinya yang dihubungkan dengan kontaktor dan tipe NO digunakan untuk tombol on. *Push button* ada juga yang bertipe NC, biasanya digunakan untuk tombol off. Terdapat 4 konfigurasi saklar *push button* yaitu tanpa-pengunci (*no-guard*), pengunci-penuh (*full guard*), *entended guard*, dan *mushroom button* (Widodo, 2011). Bentuk fisik jenis *push button* dapat dilihat pada Gambar 2.11.



Gambar 2.10 Simbol *Push button* (Widodo, 2011).



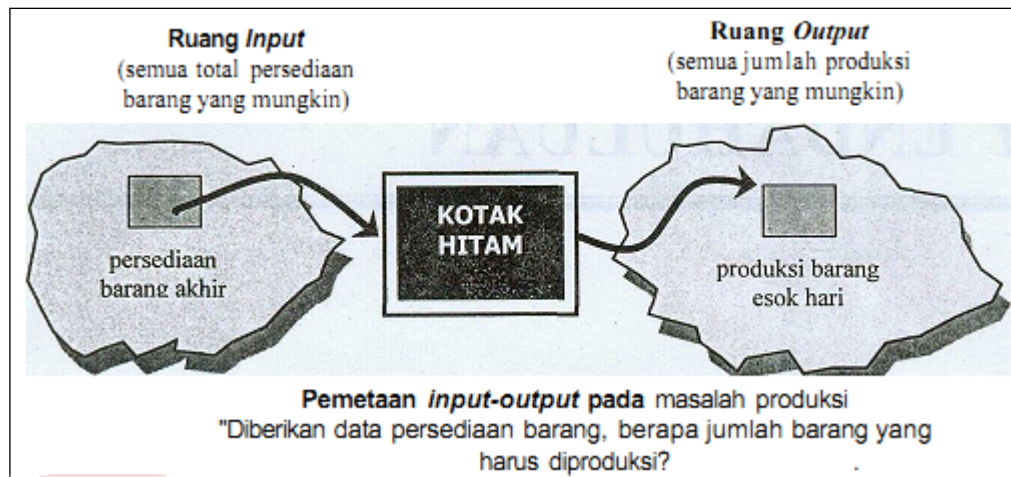
Gambar 2.11 *Push button* (Widodo, 2011).

2.6 Fuzzy

Logika fuzzy merupakan salah satu komponen pembentuk soft computing. Logika fuzzy pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965. Dasar logika fuzzy adalah teori himpunan fuzzy. Pada teori himpunan fuzzy, peranan derajat keanggotaan sebagai penentu. Keberadaan elemen dalam suatu himpunan sangatlah penting. Nilai keanggotaan atau derajat keanggotaan atau membership function menjadi ciri utama dari penalaran dengan logika fuzzy tersebut. Dalam banyak hal, logika fuzzy digunakan sebagai suatu cara untuk memetakan permasalahan dari input menuju ke output yang diharapkan. Beberapa contoh yang dapat diambil antara lain:

1. Manajer pergudangan mengatakan pada manajer produksi seberapa banyak persediaan barang pada akhir minggu ini, kemudian manajer produksi akan menetapkan jumlah barang yang harus diproduksi esok hari.
2. Seorang pegawai melakukan tugasnya dengan kinerja yang sangat baik, kemudian atasan akan memberikan reward yang sesuai dengan kinerja pegawai tersebut.

Salah satu contoh pemetaan suatu input-output dalam bentuk grafis seperti pada gambar 2.12



Gambar 2.12 Contoh Pemetaan Input-Output (Kusumadewi & Purnomo, 2004)

Ada beberapa alasan mengapa orang menggunakan logika fuzzy, antara lain:

1. Konsep logika fuzzy mudah dimengerti. Karena logika fuzzy menggunakan dasar teori himpunan, maka konsep matematis yang mendasari penalaran fuzzy tersebut cukup mudah untuk dimengerti.
2. Logika fuzzy sangat fleksibel, artinya mampu beradaptasi dengan perubahan-perubahan, dan ketidakpastian yang menyertai permasalahan.
3. Logika fuzzy memiliki toleransi terhadap data yang tidak tepat.
4. Logika fuzzy mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks.

5. Logika fuzzy dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika fuzzy dapat bekerjasama dengan teknik-teknik kendali secara konvensional.

2.6.1 Himpunan Fuzzy

Pada himpunan tegas (crisp), nilai keanggotaan suatu item x dalam suatu himpunan A , yang sering ditulis dengan $\mu_A(x)$, memiliki dua kemungkinan, yaitu:

- satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
- nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

2.6.2 Fungsi Keanggotaan

Fungsi Keanggotaan (membership function) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data ke dalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan nilai keanggotaan adalah dengan melalui pendekatan fungsi.

2.6.3 Operator Dasar

a. Operator AND

Operator ini berhubungan dengan operasi interseksi pada himpunan. α -predikat sebagai hasil operasi dengan operator AND diperoleh dengan

mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cap B} = \min(\mu_A(x), \mu_B(y))$$

b. Operator OR

Operator ini berhubungan dengan operasi union pada himpunan. α -predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cup B} = \max(\mu_A(x), \mu_B(y))$$

c. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan. α -predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangi nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

$$\mu_{A^c} = 1 - \mu_A(x)$$

2.6.4 METODE SUGENO

Penalaran dengan metode SUGENO hampir sama dengan penalaran MAMDANI, hanya saja output (konsekuen) sistem tidak berupa himpunan fuzzy, melainkan berupa konstanta atau persamaan linear. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985, sehingga metode ini sering juga dinamakan dengan Metode TSK. Menurut Cox (1994), Metode TSK terdiri-dari 2 jenis, yaitu:

a. Model Fuzzy Sugeno Orde-Nol

Secara umum bentuk model fuzzy SUGENO Orde-Nol adalah:

IF $(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ (x_3 \text{ is } A_3) \circ \dots \circ (x_N \text{ is } A_N)$ THEN $z=k$. Dengan A_i adalah himpunan fuzzy ke-i sebagai anteseden, dan k adalah suatu konstanta (tegas) sebagai konsekuen.

b. Model Fuzzy Sugeno Orde-Satu

Secara umum bentuk model fuzzy SUGENO Orde-Satu adalah:

IF $(x_1 \text{ is } A_1) \circ \dots \circ (x_N \text{ is } A_N)$ THEN $z = p_1 * x_1 + \dots + p_N * x_N + q$. Dengan A_i adalah himpunan fuzzy ke-i sebagai anteseden, dan p_i adalah suatu konstanta (tegas) ke-i dan q juga merupakan konstanta dalam konsekuen.

Apabila komposisi aturan menggunakan metode SUGENO, maka defuzzifikasi dilakukan dengan cara mencari nilai rata-ratanya.

2.7 Microcontroller ATMega16

Fitur-fitur yang dimiliki ATMega16 sebagai berikut:

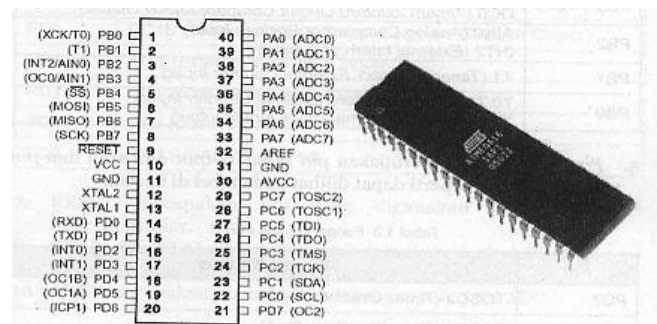
1. Mikrokontroler AVR 8 bit yang memiliki kemampuan tinggi dengan daya rendah.
2. Arsitektur RISC dengan *throughput* mencapai 16 MIPS pada frekuensi 16MHz.
3. Memiliki kapasitas *Flash* memori 16 Kbyte, EEPROM 512 Byte, dan SRAM 1 Kbyte.
4. Saluran I/O sebanyak 32 buah, yaitu *PORT A*, *PORT B*, *PORT C* dan *PORT D*.
5. CPU yang terdiri atas 32 buah *register*.
6. Unit interupsi internal dan eksternal.
7. Port USART untuk komunikasi serial.

8. Fitur *Peripheral*

- Tiga buah *Timer/Counter* dengan kemampuan perbandingan.
 - 2 buah *Timer/Counter* 8 bit dengan Prescaler terpisah dan Mode Compare.
 - 1 buah *Timer/Counter* 16 bit dengan Prescaler terpisah, Mode Compare dan Mode Capture.
- Real Timer Counter dengan oscillator tersendiri.
- 4 channel PWM.
- 8 channel, 10-bit ADC
 - 8 single-ended channel.
 - 7 Differential Channel hanya pada kemasan TQFP.
 - 2 Differential Channel dengan Programmable Gain 1x, 10x, atau 200x.
- Byte-oriented Two-wire Serial Interface.
- Programmable Serial USART.
- Antarmuka SPI
- Watchdog Timer dengan oscillator internal.
- On-chip Analog Comparator.

2.7.1. Konfigurasi Pin AVR ATmega16

Konfigurasi pin ATmega16 dengan kemasan 40 pin PDIP (Dual In-line Package) dapat dilihat pada Gambar 2.13.



Gambar 2.13 Konfigurasi pin ATmega16 (Andrianto, 2013).

Dari gambar 2.13 dapat dijelaskan fungsi dari masing-masing pin ATmega16 sebagai berikut:

1. VCC merupakan pin yang berfungsi sebagai masukan catu daya.
2. GND merupakan pin Ground.
3. Port A(PA0...PA7) merupakan pin input/output dua arah dan pin masukan ADC.
4. Port B(PB0...PB7) merupakan pin input/output dua arah dan pin fungsi khusus, seperti dapat dilihat pada Tabel 2.1.

Tabel 2.1 Fungsi khusus Port B

Pin	Fungsi khusus
PB7	SCK (SPI Bus serial Clock)
PB6	MISO (SPI Bus master output/slave output)
PB5	MOSI (SPI Bus master output/slave input)
PB4	SS (SPI slave select input)
PB3	AIN 1 (Analaog Comparator Negative input) OC0 (timer/Counter0 Output Compare Match output)
PB2	AIN 0 (Analaog Comparator Positive input) INT2 (External Interrupt 2 Input)
PB1	T1 (Timer/counter1 External counter input)
PB0	T0 T1 (Timer/counter0 External counter input) XCK (usart external clock input/output)

5. Port C(PC0..PC7) merupakan pin input/output dua arah dan pin fungsi

khusus, seperti dapat dilihat pada Tabel 2.2.

Tabel 2.2 Fungsi khusus Port C

Pin	Fungsi khusus
PC7	TOSC2 (Timer Oscillator Pin2)
PC6	TOSC1 (Timer Oscillator Pin1)
PC5	TDI (JTAG Test Data In)
PC4	TDO (JTAG Test Data Out)
PC3	TMS (JTAG Test Mode Select)
PC2	TCK (JTAG Test Clock)
PC1	SDA (Two-wire Serial Bus Data Input/Output Line)
PC0	SCL (Two-wire Serial Bus Clock Line)

6. Port D(PD0..PD7) merupakan pin input/output dua arah dan pin fungsi

khusus, seperti dapat dilihat pada Tabel 2.3.

Tabel 2.3 Fungsi khusus Port D

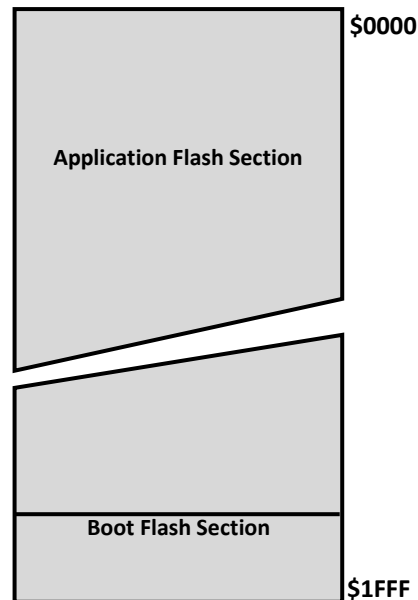
Pin	Fungsi khusus
PD7	OC2 (timer/Counter2 Output Compare Match output)
PD6	ICP (Timer/Counter1 Input Capture Pin)
PD5	OC1A (timer/Counter1 Output Compare A Match output)
PD4	OC1B (timer/Counter1 Output Compare B Match output)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output pin)
PD0	RXD (USART Input Pin)

7. RESET merupakan pin yang digunakan untuk me-reset mikrokontroller.
8. XTAL1 dan XTAL2 merupakan pin masukan clock eksternal.
9. AVCC merupakan pin masukan tegangan untuk ADC.
10. AREF merupakan pin masukan tegangan referensi ADC.

2.7.2 Peta Memori AVR ATmega16

1. Memori Program

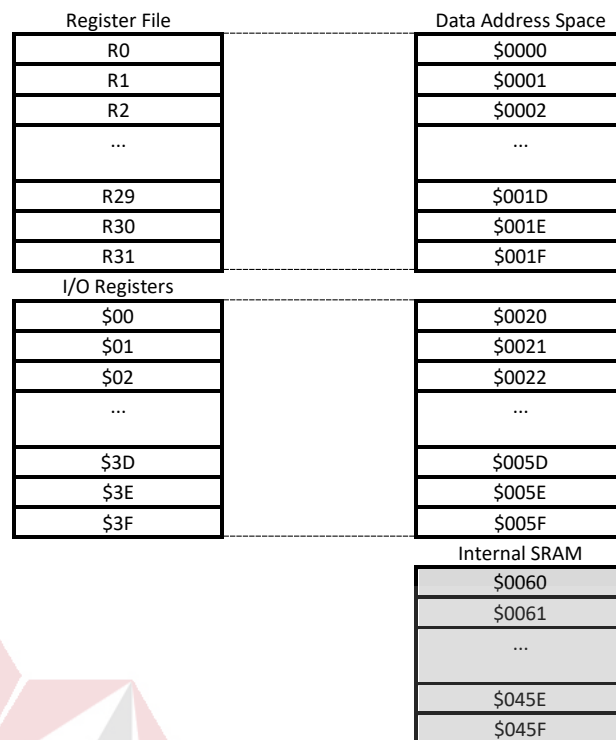
Arsitektur AVR mempunyai dua memori utama, yaitu memori data dan memori program. Selain itu, ATmega16 memiliki memori EEPROM untuk menyimpan data. ATmega16 memiliki 16Kbyte On-Chip In-system Reprogrammable Flash Memory untuk menyimpan program. Karena semua instruksi AVR memiliki format 16 atau 32 bit, Flash diatur dalam 8K x 16 bit. Untuk keamanan program, memori program, flash dibagi ke dalam dua bagian, yaitu bagian program Boot dan aplikasi. Bootloader adalah program kecil yang bekerja pada saat start up time yang dapat memasukkan seluruh program aplikasi ke dalam memori prosesor.



Gambar 2.14 Peta Memori Program AVR ATmega16 (Andrianto, 2013).

2. Memori Data (SRAM)

Memori data AVR ATmega16 terbagi menjadi 3 bagian, yaitu 32 buah register umum, 64 buah register I/O dan 1 Kbyte SRAM internal. General purpose register menempati alamat data terbawah, yaitu \$00 sampai \$1F. Sedangkan memori I/O menempati 64 alamat berikutnya mulai dari \$20 hingga \$5F. Memori I/O merupakan register yang khusus digunakan untuk mengatur fungsi terhadap berbagai peripheral mikrokontroler seperti control register, timer/counter, fungsi-fungsi I/O dan sebagainya. 1024 alamat memori berikutnya mulai alamat \$60 hingga \$45F digunakan untuk SRAM internal.



Gambar 2.15 Peta memori Data AVR ATmega16 (Andrianto, 2013).

3. Memori Data EEPROM

ATmega16 terdiri dari 512 byte memori data EEPROM 8 bit, data dapat tulis/baca dari memori ini, ketika catu daya dimatikan, data terakhir yang ditulis pada memori EEPROM masih tersimpan pada memori ini, atau dengan kata lain memori EEPROM bersifat nonvolatile. Alamat EEPROM mulai \$000 sampai \$1FF.

Tabel 2.4 Pengalamatan Register I/O

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
\$3F (\$5F)	SREG	I	T	H	S	V	N	Z	C
\$3E (\$5E)	SPH	-	-	-	-	-	SP10	SP9	SP8
\$3D (\$5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
\$3C (\$5C)	OCR0	Timer/Counter0 Output Compare Register							
\$3B (\$5B)	GICR	INT1	INT1	INT1	-	-	-	IVSEL	IVCE
\$3A (\$5A)	GIFR	INTF1	INT1	INT1	-	-	-	-	-
\$39 (\$59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0
\$38 (\$58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
\$37 (\$57)	SPMCR	SPMIE	RWWSB	-	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN
\$36 (\$56)	TWCR	TWNT	TWEA	TWSTA	TWST0	TWWC	TWEN	-	TWIE
\$35 (\$55)	MCUCR	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00
\$34 (\$54)	MCUCSR	JTD	ISC2	-	JTRF	WDRF	BORF	EXTRF	PORF
\$33 (\$53)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
\$32 (\$52)	TCNT0	Timer/Counter0 (8 Bits)							
\$31 ⁽¹⁾ (\$51) ⁽¹⁾	OSCCAL	Oscillator Calibration Register							
	OCDR	On-Chip Debug Register							
\$30 (\$50)	SFIOR	ADTS2	ADTS1	ADTS0	-	ACME	PUD	PSR2	PSR10
\$2F (\$4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10
\$2E (\$4E)	TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
\$2D (\$4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte							
\$2C (\$4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte							
\$2B (\$4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High Byte							
\$2A (\$4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low Byte							
\$29 (\$49)	OCR1BH	Timer/Counter1 – Output Compare Register B High Byte							
\$28 (\$48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low Byte							
\$27 (\$47)	ICR1H	Timer/Counter1 – Input Capture Register High Byte							
\$26 (\$46)	ICR1L	Timer/Counter1 – Input Capture Register Low Byte							
\$25 (\$45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20
\$24 (\$44)	TCNT2	Timer/Counter2 (8Bits)							
\$23 (\$43)	OCR2	Timer/Counter2 Output Compare Register							
\$22 (\$42)	ASSR	-	-	-	-	AS2	TCN2UB	OCR2UB	TCR2UB
\$21 (\$41)	WDTCR	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0
\$20 ⁽²⁾ (\$40) ⁽²⁾	UBRRH	URSEL	-	-	-	-	UBRR(11:8)		
	UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL
\$1F (\$3F)	EEARH	-	-	-	-	-	-	-	EEAR8
\$1E (\$3E)	EEARL	EEPROM Address Register Low Byte							
\$1D (\$3D)	EEDR	EEPROM Data Register							
\$1C (\$3C)	EEDR	-	-	-	-	EERIE	EEMWE	EWE	EERE
\$1B (\$3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0
\$1A (\$3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0
\$19 (\$39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0
\$18 (\$38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
\$17 (\$37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
\$16 (\$36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
\$15 (\$35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0
\$14 (\$34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
\$13 (\$33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
\$12 (\$32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
\$11 (\$31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
\$10 (\$30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
\$0F (\$2F)	SPDR	SPI Data Register							
\$0E (\$2E)	SPSR	SPIF	WCOL	-	-	-	-	-	SPI2X
\$0D (\$2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
\$0C (\$2C)	UDR	USART I/O Data Register							
\$0B (\$2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM
\$0A (\$2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
\$09 (\$29)	UBRRL	USART Baud Rate Register Low Byte							
\$08 (\$28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0
\$07 (\$27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
\$06 (\$26)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
\$05 (\$25)	ADCH	ADC Data Register High Byte							
\$04 (\$24)	ADCL	ADC Data Register Low Byte							
\$03 (\$23)	TWDR	Two-wire Serial Interface Data Register							
\$02 (\$22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Page
\$01 (\$21)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	-	TWPS1	TWPS0	182
\$00 (\$20)	TWBR	Two-wire Serial Interface Bit Rate Register								181

4. Status Register (SREG)

Status Register adalah register berisi status yang dihasilkan pada setiap operasi yang dilakukan ketika suatu instruksi dieksekusi. SREG merupakan bagian dari inti CPU mikrokontroler.

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.16 Status register atmega16 (Andrianto, 2013).

Bit7 : I (global Interrupt Enable).

Bit untuk mengaktifkan interupsi, jika terjadi interupsi yang dipicu oleh hardware bit I akan di-clear, dan akan di-set kembali menggunakan instruksi RETI.

Bit 6 : T (Bit Copy Storage).

Bit 5 : H (Half Carry Flag).

Bit 4 : S (Sign Bit).

Bit S merupakan hasil operasi EOR antara flag-N (negative) dan flag-V (komplemen dua overflow).

Bit 3 : V (Two's Complement Overflow Flag).

Bit berguna untuk mendukung operasi aritmatika.

Bit 2 : N (Negative Flag).

Apabila suatu operasi menghasilkan bilangan negative, maka flag-N akan di-set.

Bit 1 : Z (Zero Flag).

Bit akan di-set bila hasil operasi yang diperoleh adalah nol.

Bit 0 : C (Carry Flag).

Bit akan di-set bila hasil operasi menghasilkan carry.

2.7.3 ADC (Analog to Digital Converter)

ADC adalah suatu alat yang mampu untuk mengubah sinyal atau tegangan analog menjadi informasi digital. Resolusi ADC selalu dinyatakan sebagai jumlah bit-bit dalam kode keluaran digitalnya. Misalnya, ADC dengan resolusi n-bit memiliki 2ⁿ kode digital yang mungkin dan berarti juga memiliki 2ⁿ step level. Jika resolusi ADC semakin tinggi, maka semakin banyak kemungkinan nilai-nilai analog yang bisa disajikan. Misalnya ADC dengan resolusi 8 bit menghasilkan bilangan 0 sampai dengan 255 (256 bilangan dan 255 step), dengan demikian tidak mungkin menyajikan semua kemungkinan nilai-nilai analog. Jika sekarang resolusinya menjadi 10 bit maka akan menghasilkan bilangan 0 sampai dengan 1023 (1024 bilangan dan 1023 step).

$$V_{ADC} = \left(\frac{\text{Nilai Digital}}{256} \right) \times V_{REF} \dots\dots\dots (2.1)$$

$$\text{Nilai Digital} = \left(\frac{V_{ADC}}{V_{REF}} \right) \times 256 \dots\dots\dots (2.2)$$

AVR ATMegai6 memiliki fitur ADC sebanyak 8 channel dengan resolusi 10-bit Register yang digunakan untuk setting ADC ini adalah

- **ADC Control and Status Register A - ADCSRA**

Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.17 ADC Control and Status Register A – ADCSRA (Andrianto, 2013).

ADEN : 1 = adc enable, 0 = adc disable

ADSC : 1 = mulai konversi, 0 = konversi belum terjadi

ADATE : 1 = auto trigger diaktifkan, trigger berasal dari sinyal yang dipilih (set pada register SFIOR bit ADTS). ADC akan start konversi pada edge positif sinyal trigger.

ADIF : diset ke 1, jika konversi ADC selesai dan data register ter-Update. Namun, ADC Conversion Complete Interrupt dieksekusi jika bit ADIE dan bit-1 dalam register SREG di set.

ADIE : di set 1, jika bit-1 dalam register SREG di-set.

ADPS[0..2] : Bit pengatur clock ADC, faktor pembagi 0...7 = 2, 4, 8, 16, 32, 64, 128.

Tabel 2.5 Konfigurasi *Clock* ADC

ADPS[2..0]	Besar clock ADC
000 – 001	$F_{0s0}/2$
010	$F_{0s0}/4$
011	$F_{0s0}/8$
100	$F_{0s0}/16$
101	$F_{0s0}/32$
110	$F_{0s0}/64$
111	$F_{0s0}/128$

- ADC Multiplexer - ADMUX

Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.18 ADC Multiplexer (Andrianto, 2013).

REPS_{0,1} : Pemilihan tegangan referensi ADC

00 : $v_{ref} = AREF$

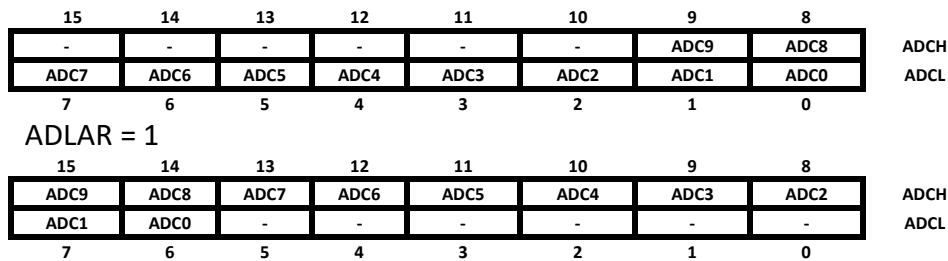
01 : $v_{ref} = AVCC$ dengan eksternal kapasitor pada AREF

10 : $v_{ref} = Reserved$

11 : vref = internal 2.56 volt dengan eksternal kapasitor pada AREF

ADLAR : Untuk setting format data hasil konversi ADC, default = 0.

ADLAR = 0

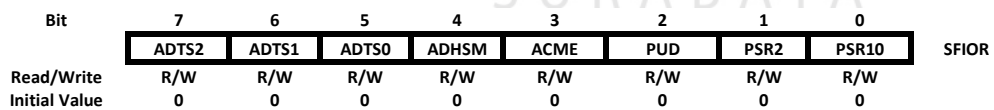


Gambar 2.19 ADLAR (Andrianto, 2013).

MUX_{0.4} : Pemilihan channel ADC yang digunakan, 0 = channel₁, 1 = channel₂, dst.

- **Special Function 10 Register -SFIOR**

SFIOR merupakan register 8 bit pengatur sumber picu konversi ADC, apakah dari picu eksternal atau dari picu internal, susunannya sebagai berikut:



Gambar 2.20 Register SFIOR (Andrianto, 2013).

ADTS[0...2] : Pemilihan *trigger* (pengatur picu) untuk konversi ADC, *bit-bit* ini akan berfungsi jika *bit* ADATE pada *register* ADCSRA bernilai 1. Konfigurasi bit ADTS[0...2] sebagai berikut:

Tabel 2.6 Pemilihan sumber picu ADC

ADTS[2...0]	Sumber Triger (picu)
000	Mode Free Runing
001	Komparator analog
010	Interupsi Eksternal
011	Timer/Counter 0 Compare Match
100	Timer/Counter 0 Overflow
101	Timer/Counter 1 Compare Match
110	Timer/Counter 1 Overflow
111	Timer/Counter 1 Capture Event

ADHSM : 1, ADC high speed mode enabled.

Untuk operasi ADC, bit ACME, PUD, PSR₂, dan PSR₁₀ tidak diaktifkan.

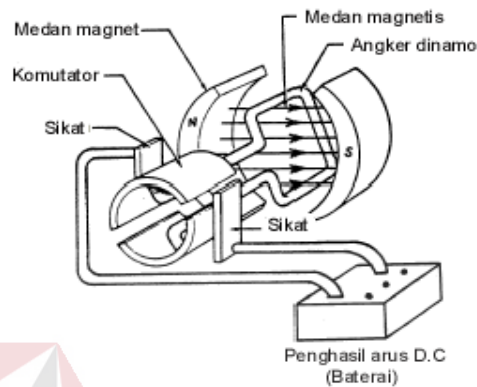
2.8 Motor DC

2.8.1 Pengertian Motor DC

Motor listrik merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk, misalnya memutar *impeller* pompa, *fan* atau *blower*, menggerakkan kompresor, mengangkat bahan, dll. Motor listrik digunakan juga di rumah (*mixer*, bor listrik, *fan* angin) dan di industri. Motor listrik kadangkala disebut “kuda kerja” nya industri sebab diperkirakan bahwa motor-motor menggunakan sekitar 70% beban listrik total di industri.

Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor dc disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet, maka akan timbul tegangan (GGL) yang berubah-ubah arah pada setiap setengah putaran, sehingga merupakan tegangan bolak-balik. Prinsip kerja

dari arus searah adalah membalik fasa tegangan dari gelombang yang mempunyai nilai positif dengan menggunakan komutator, dengan demikian arus yang berbalik arah dengan kumparan jangkar yang berputar dalam medan magnet. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen.

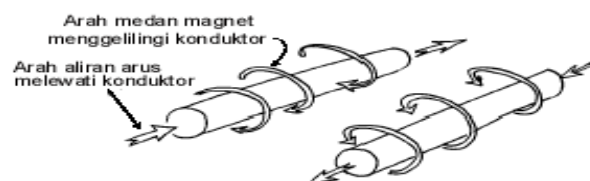


Gambar 2.21 Motor DC Sederhana (Sumanto, 1994).

Catu tegangan DC dari baterai menuju ke lilitan melalui sikat yang menyentuh komutator, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan satu lilitan pada gambar di atas disebut angker dinamo. Angker dinamo adalah sebutan untuk komponen yang berputar di antara medan magnet.

2.8.2 Prinsip Dasar Cara Kerja

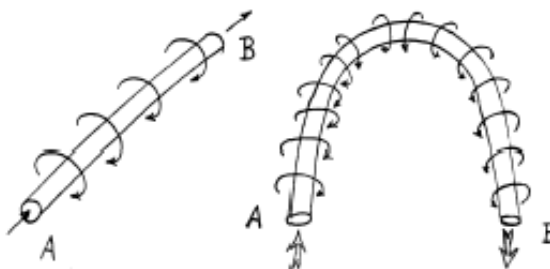
Jika arus lewat pada suatu konduktor, timbul medan magnet di sekitar konduktor. Arah medan magnet ditentukan oleh arah aliran arus pada konduktor.



Gambar 2.22 Medan magnet yang membawa arus mengelilingi konduktor

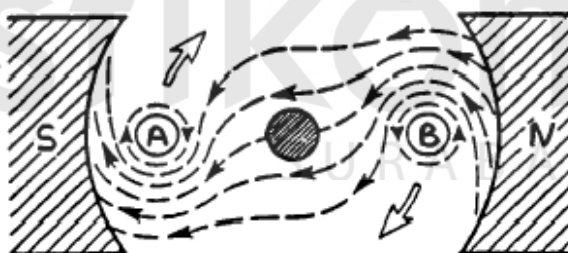
(Djati, 2010).

Aturan Genggaman Tangan Kanan bisa dipakai untuk menentukan arah garis fluks di sekitar konduktor. Genggam konduktor dengan tangan kanan dengan jempol mengarah pada arah aliran arus, maka jari-jari anda akan menunjukkan arah garis fluks. Gambar 3 menunjukkan medan magnet yang terbentuk di sekitar konduktor berubah arah karena bentuk U.



Gambar 2.23 Konduktor Berbentuk U (Djati, 2010).

Jika konduktor berbentuk U (angker dinamo) diletakkan di antara kutub utara dan selatan yang kuat medan magnet konduktor akan berinteraksi dengan medan magnet kutub. Lihat gambar 2.24



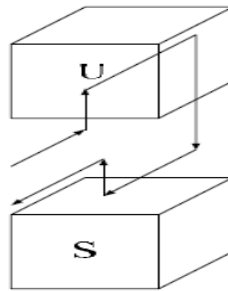
Gambar 2.24 Reaksi garis Fluks (Djati, 2010).

Lingkaran bertanda A dan B merupakan ujung konduktor yang dilengkungkan (*looped conductor*). Arus mengalir masuk melalui ujung A dan keluar melalui ujung B. Medan konduktor A yang searah jarum jam akan menambah medan pada kutub dan menimbulkan medan yang kuat di bawah konduktor. Konduktor akan berusaha bergerak ke atas untuk keluar dari medan kuat ini. Medan konduktor B yang berlawanan arah jarum jam akan menambah

medan pada kutub dan menimbulkan medan yang kuat di atas konduktor. Konduktor akan berusaha untuk bergerak turun agar keluar dari medan yang kuat tersebut. Gaya-gaya tersebut akan membuat angker dinamo berputar searah jarum jam. Mekanisme kerja untuk seluruh jenis motor secara umum :

- Arus listrik dalam medan magnet akan memberikan gaya.
- Jika kawat yang membawa arus dibengkokkan menjadi sebuah lingkaran / *loop*, maka kedua sisi *loop*, yaitu pada sudut kanan medan magnet, akan mendapatkan gaya pada arah yang berlawanan.
- Pasangan gaya menghasilkan tenaga putar / *torque* untuk memutar kumparan.
- Motor-motor memiliki beberapa *loop* pada dinamonya untuk memberikan tenaga putaran yang lebih seragam dan medan magnetnya dihasilkan oleh susunan elektromagnetik yang disebut kumparan medan.

Pada motor dc, daerah kumparan medan yang dialiri arus listrik akan menghasilkan medan magnet yang melingkupi kumparan jangkar dengan arah tertentu. Konversi dari energi listrik menjadi energi mekanik (motor) maupun sebaliknya berlangsung melalui medan magnet, dengan demikian medan magnet disini selain berfungsi sebagai tempat untuk menyimpan energi, sekaligus sebagai tempat berlangsungnya proses perubahan energi, daerah tersebut dapat dilihat pada gambar 2.25.



Gambar 2.25 Prinsip Kerja Motor Dc (Djati, 2010).

Agar proses perubahan energi mekanik dapat berlangsung secara sempurna, maka tegangan sumber harus lebih besar daripada tegangan gerak yang disebabkan reaksi lawan. Dengan memberi arus pada kumparan jangkar yang dilindungi oleh medan maka menimbulkan perputaran pada motor.

Dalam memahami sebuah motor, penting untuk mengerti apa yang dimaksud dengan beban motor. Beban dalam hal ini mengacu kepada keluaran tenaga putar / *torque* sesuai dengan kecepatan yang diperlukan. Beban umumnya dapat dikategorikan ke dalam tiga kelompok :

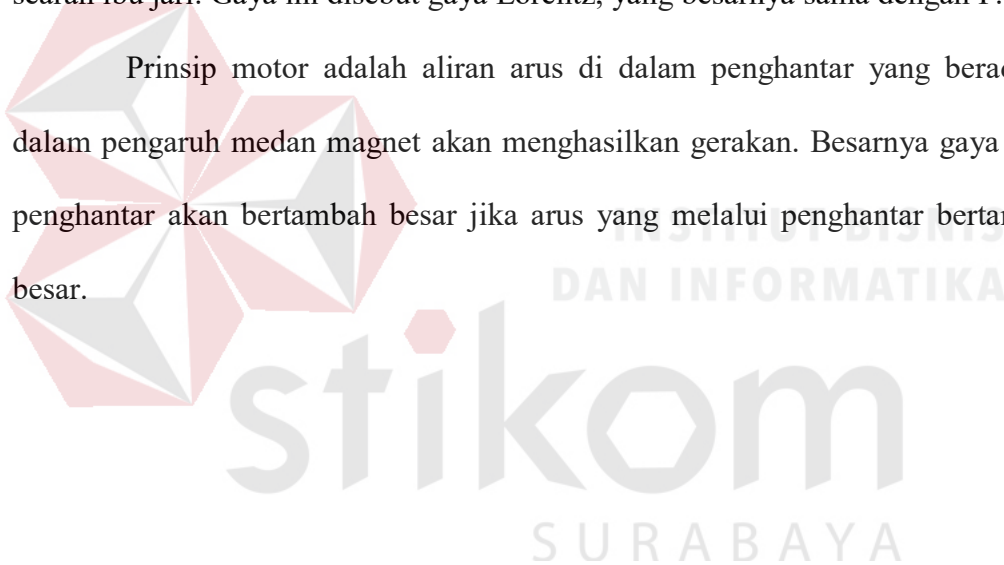
1. Beban *torque* konstan adalah beban dimana permintaan keluaran energinya bervariasi dengan kecepatan operasinya namun *torquencya* tidak bervariasi. Contoh beban dengan *torque* konstan adalah *corveyors*, *rotary kilns*, dan pompa *displacement* konstan.
2. Beban dengan variabel *torque* adalah beban dengan *torque* yang bervariasi dengan kecepatan operasi. Contoh beban dengan variabel *torque* adalah pompa sentrifugal dan *fan* (*torque* bervariasi sebagai kuadrat kecepatan).
3. Peralatan Energi Listrik : Motor Listrik.

4. Beban dengan energi konstan adalah beban dengan permintaan *torque* yang berubah dan berbanding terbalik dengan kecepatan. Contoh untuk beban dengan daya konstan adalah peralatan-peralatan mesin.

2.8.3 Prinsip Arah Putaran Motor

Untuk menentukan arah putaran motor digunakan kaedah Flamming tangan kiri. Kutub-kutub magnet akan menghasilkan medan magnet dengan arah dari kutub utara ke kutub selatan. Jika medan magnet memotong sebuah kawat penghantar yang dialiri arus searah dengan empat jari, maka akan timbul gerak searah ibu jari. Gaya ini disebut gaya Lorentz, yang besarnya sama dengan F .

Prinsip motor adalah aliran arus di dalam penghantar yang berada di dalam pengaruh medan magnet akan menghasilkan gerakan. Besarnya gaya pada penghantar akan bertambah besar jika arus yang melalui penghantar bertambah besar.



BAB III

METODE PENELITIAN

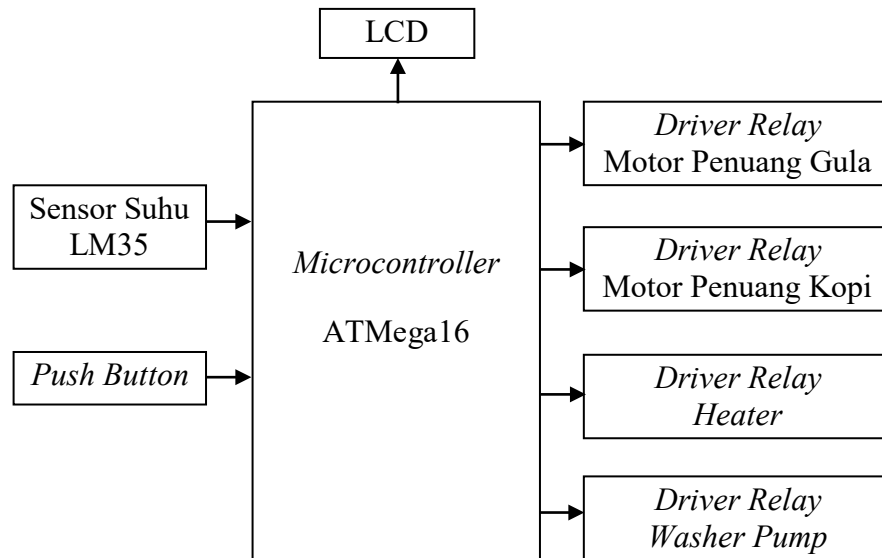
Perancangan perangkat keras dilakukan dengan metode penelitian yang didasarkan pada studi kepustakaan berupa data-data literatur dari masing-masing komponen, informasi dari *internet*, dan konsep-konsep teoritis dari buku-buku penunjang. Setelah literatur terkumpul barulah dilakukan perancangan perangkat keras yang dilanjutkan dengan perancangan dan pembuatan perangkat lunak yaitu program sederhana untuk melakukan pengujian pada tiap bagian perangkat keras yang telah dibuat dan program untuk menjalankan perangkat keras secara keseluruhan.

Pada bab ini akan dibahas mengenai masalah yang timbul dalam perencanaan dan pembuatan perangkat keras (*hardware*) maupun perangkat lunak (*software*). Dari kedua bagian tersebut akan dipadukan agar dapat bekerja sama untuk menjalankan sistem yang baik.

Perencanaan ini diperlukan sebelum proses pembuatan sistem tersebut. Perancangan ini berguna agar pengerjaan tahapan selanjutnya berjalan dengan lancar. Tahapan-tahapannya meliputi tahap pembuatan perangkat keras, perangkat lunak dan menggabungkan keduanya.

3.1 Perancangan Perangkat Keras

Perancangan perangkat keras pada sistem dapat dilakukan berdasarkan blok diagram yang terdapat pada Gambar 3.1.



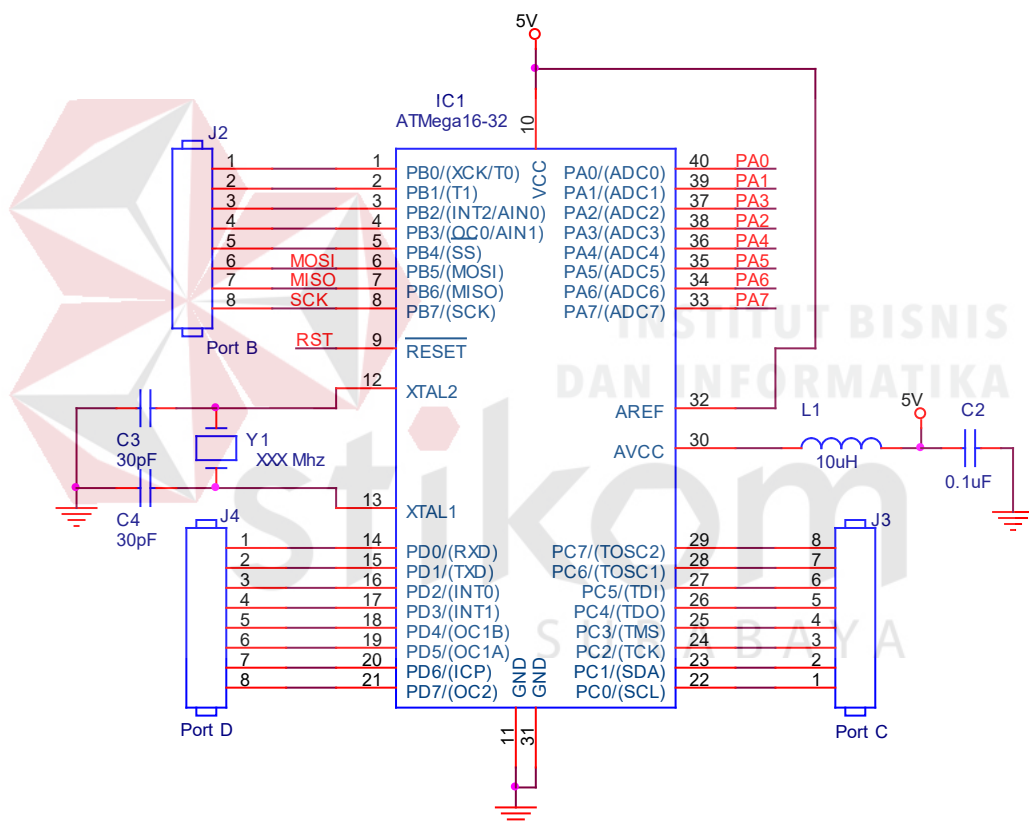
Gambar 3.1 Diagram Blok Sistem

Sensor suhu LM35 berfungsi untuk mengukur suhu air ketika dipanaskan sehingga nanti akan didapatkan suhu yang diinginkan. Dalam hal ini sensor suhu LM35 mengkonversi masukan suhu dalam skala °C menjadi keluaran tegangan dalam skala mV. Keluaran dari LM35 dimasukkan ke ADC dari *microcontroller* ATmega16. *Microcontroller* ATmega16 digunakan untuk mengolah data dari sensor LM35 yang dimasukkan ke ADC dari *microcontroller* ATmega16, mengontrol *driver relay* dan tampilan ke LCD. *Driver relay* terdiri dari *driver relay* motor penuang kopi, *driver relay* motor penuang gula, *driver relay heater*, *driver relay washer pump*. *Driver relay* motor penuang kopi dan *driver relay* motor penuang gula berfungsi untuk menuangkan kopi dan gula dari tempatnya sesuai dengan takaran yang diinginkan oleh *user*. *Driver relay heater* berfungsi untuk mengontrol pemanasan air sehingga didapat suhu air yang diinginkan. *Driver relay washer pump* digunakan untuk menjalankan *washer pump* yang berfungsi untuk mengeluarkan air yang sudah dipanaskan. *Driver* ini bekerja

apabila suhu air yang telah dipanaskan sesuai dengan suhu mendidih air yaitu 100°C .

3.1.1 Rangkaian Minimum Sistem

Untuk menjalankan *microcontroller* dibutuhkan sebuah rangkaian agar *microcontroller* tersebut dapat bekerja dengan baik. Rangkaian minimum sistem terbagi menjadi 3 rangkaian utama yaitu rangkaian IC, rangkaian *reset* dan rangkaian *crystal*.

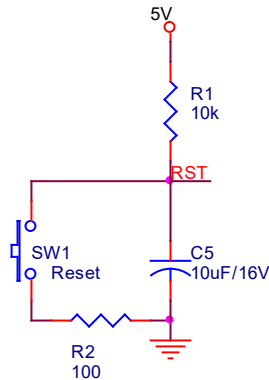


Gambar 3.2 Rangkaian Minimum Sistem

Cara Kerja Rangkaian :

Pada *Microcontroller* tipe ATmega16 terdapat 40 pin. IC ini memiliki 4 *Port*. *Port A* digunakan untuk *input* ADC dari sensor suhu. *Port B* digunakan

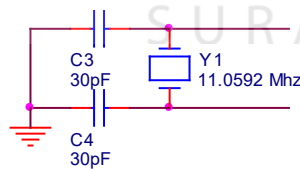
untuk *input* dari *push button* dan *output* untuk *washer pump*. *Port C* digunakan untuk *output* LCD. *Port D* digunakan untuk *output driver relay pemanas heater*.



Gambar 3.3 Rangkaian Reset

Cara Kerja Rangkaian :

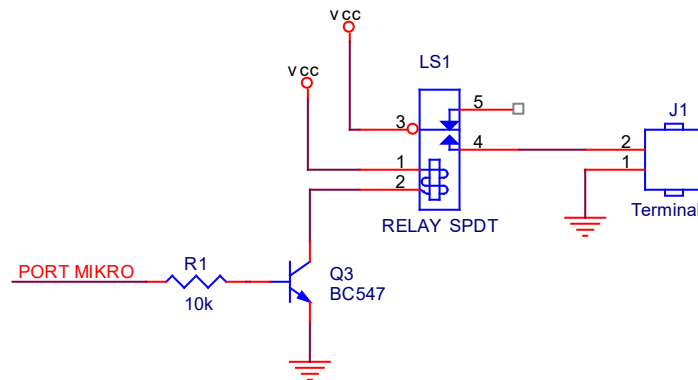
Rangkaian Reset pada tombol *reset* (SW1) digunakan untuk mengurangi *noise* serta memiliki fungsi terpenting yaitu untuk melakukan *reset* saat pertama kali catu daya dinyalakan. *Reset* untuk pertama kali merupakan hal yang terpenting sehingga dapat memastikan bahwa program telah berada pada posisi awal.



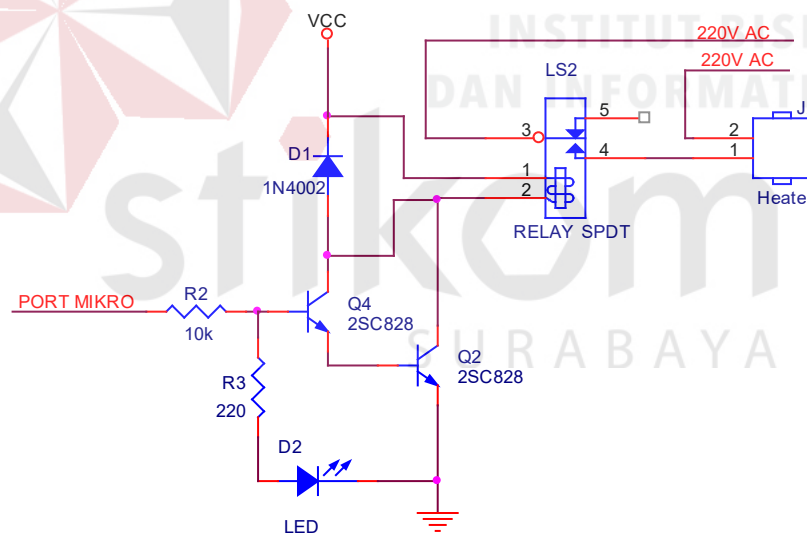
Gambar 3.4 Rangkaian *Crystal*

3.1.2 Rangkaian Relay

Rangkaian relay dibutuhkan untuk mengontrol motor penuang gula, motor penuang kopi, *washer pump* dan *heater*.



Gambar 3.5 Rangkaian *Driver Relay* Motor Penuang dan *Washer Pump*



Gambar 3.6 Rangkaian *Driver Relay Heater*

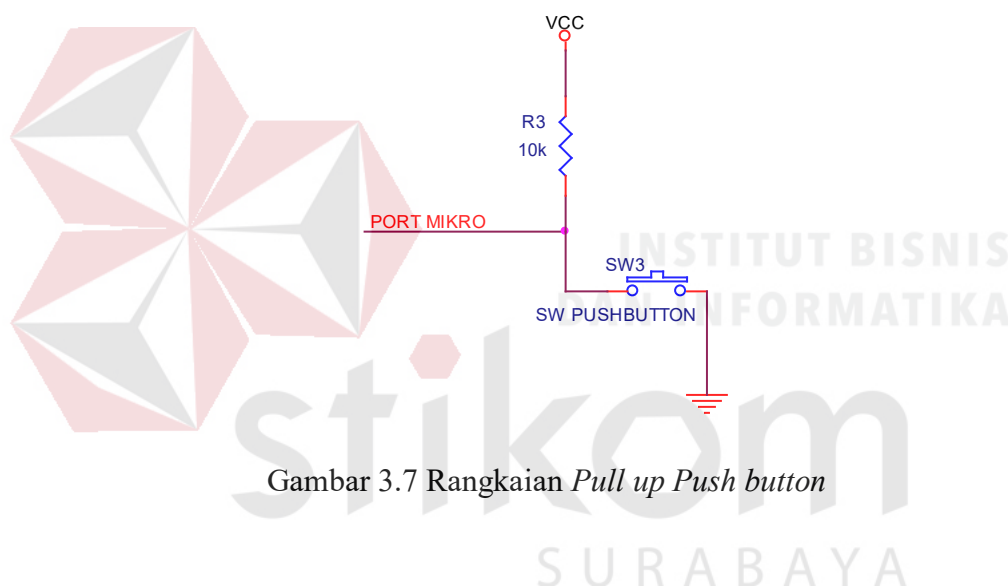
Cara Kerja Rangkaian :

Transistor pada gambar 3.5 dan gambar 3.6 berfungsi untuk menguatkan arus yang berasal dari *microcontroller*. Saat *Coil* dialiri arus maka *COM* yang

semula posisinya berada di *NC* akan berpindah ke *NO* sehingga menghubungkan tegangan 12V yang nantinya akan digunakan untuk menggerakkan motor penuang gula, motor penuang kopi dan *washer pump*. Begitu pula rangkaian *Driver Relay Heater* pada gambar 3.6 mempunyai prinsip kerja yang sama, namun *COM* menghubungkan tegangan 220V AC yang digunakan untuk menyalakan *heater*.

3.1.3 Rangkaian *Pull up PushButton*

Rangkaian ini merupakan rangkaian tombol yang digunakan untuk pemilihan menu takaran untuk gula dan kopi yang diinginkan user.



Gambar 3.7 Rangkaian *Pull up Push button*

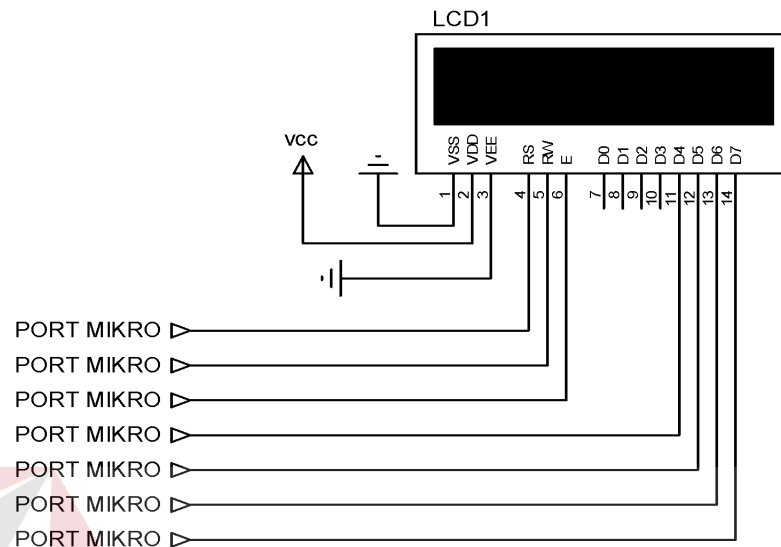
Cara kerja rangkaian :

Resistor 10k digunakan untuk *pull up*. Ketika kondisi *push button* terbuka (tidak ditekan), maka tidak ada arus yang mengalir ke resistor. Sehingga tegangan V_{out} nilai tegangan sama dengan V_{in} yaitu +5V. Dan ketika kondisi *push button* tertutup (ditekan) dan V_{out} terhubung ke *ground*, maka arus mengalir ke resistor.

3.1.4 Rangkaian LCD

LCD merupakan suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama. LCD digunakan untuk menampilkan menu takaran gula

dan kopi yang akan dipilih oleh *user*. LCD dihubungkan ke *microcontroller*. Gambar 3.8 menunjukkan rangkaian LCD.



Gambar 3.8 Rangkaian LCD

Penjelasan Komponen Rangkaian :

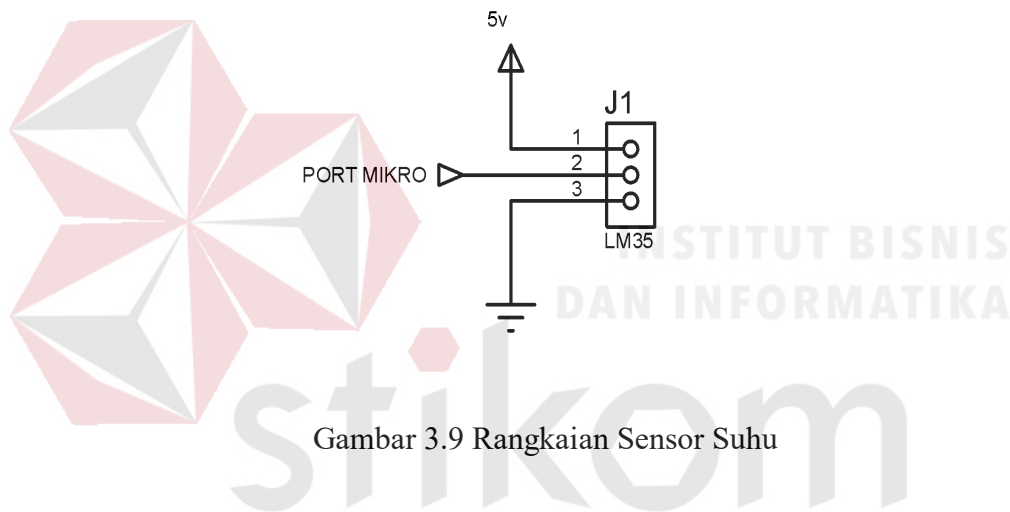
Tampilan karakter pada LCD diatur oleh Pin E, RS dan RW. Pin E pada LCD terhubung dengan Port C Pin 2 pada *microcontroller*. Jalur ini dinamakan E atau *Enable*. Jalur ini digunakan untuk memberitahu LCD bahwa sedang mengirimkan sebuah data. Untuk mengirimkan data ke LCD, maka melalui program E harus dibuat logika *low* “0” dan set pada dua jalur kontrol yang lain RS dan RW. Ketika dua jalur yang lain telah siap, set E dengan logika “1” dan tunggu untuk sejumlah waktu tertentu (sesuai dengan *datasheet* dari LCD tersebut) dan berikutnya set E ke logika *low* “0” lagi.

Pin RS pada LCD terhubung dengan Port C Pin 0 pada *microcontroller*. Jalur ini dinamakan RS atau jalur *Register Select*. Ketika RS berlogika *low* “0”, data akan dianggap sebagai sebuah perintah atau instruksi khusus (seperti clear screen, posisi kursor dll). Ketika RS berlogika *high* “1”, data yang dikirim adalah

data text yang akan ditampilkan pada *display* LCD. Pin RW pada LCD terhubung dengan Port C Pin 1 pada *microcontroller*.

3.1.5 Rangkaian Sensor Suhu

Sensor suhu berfungsi untuk mendeteksi perubahan suhu menjadi sinyal listrik dalam bentuk tegangan. Sensor suhu yang digunakan dalam perencanaan alat ini adalah IC LM35 dengan pertimbangan mempunyai rangkain yang sederhana, keluarannya linier terhadap suhu, terkalibrasi dalam derajat celcius serta mudah didapat.



Gambar 3.9 Rangkaian Sensor Suhu

Dalam perancangan alat ini menggunakan 1 buah sensor suhu LM35. Cara kerja sensor LM35 ini adalah data dari sensor ini akan berubah sesuai dengan perubahan suhu air yang diukur, data ini kemudian akan diterima oleh ADC dari *microcontroller* yang kemudian akan digunakan sebagai inputan fuzzy dan ditampilkan ke LCD. Resolusi ADC dari *microcontroller* yang digunakan adalah 8 bit.

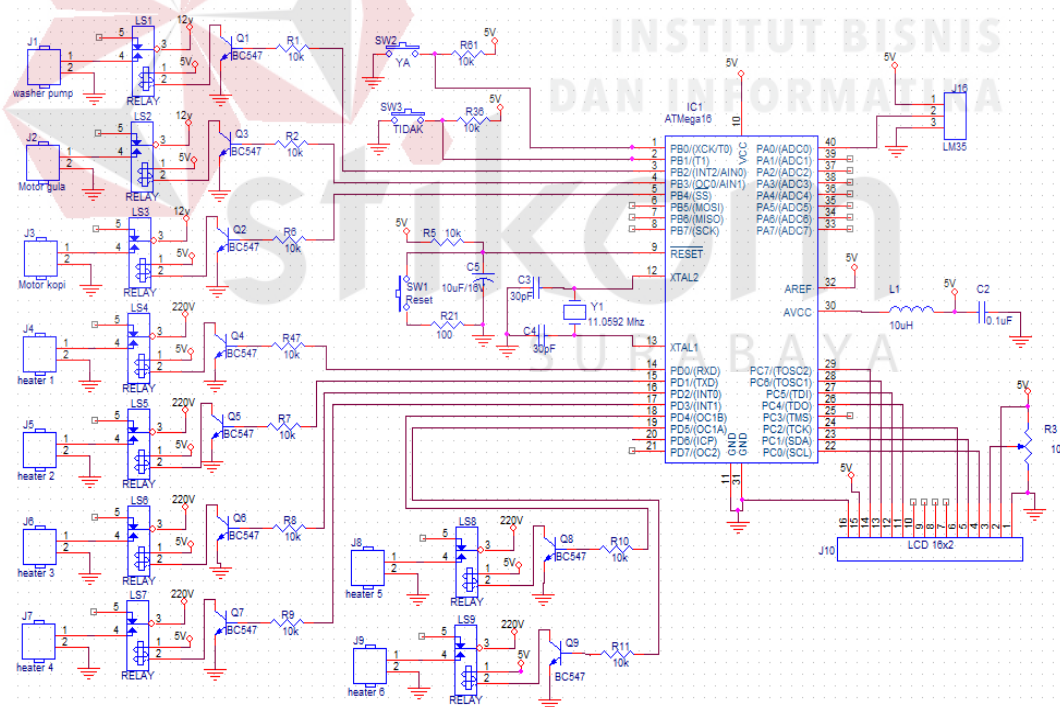
$$V_{ADC} = \left(\frac{\text{Nilai Digital}}{256} \right) \times V_{REF} \dots\dots\dots(3.1)$$

$$\text{Nilai Digital} = \left(\frac{V_{ADC}}{V_{REF}} \right) \times 256 \dots\dots\dots(3.2)$$

Dari perhitungan (3.1) maka akan didapat tegangan dari keluaran LM35 yang kemudian dari perhitungan tersebut dikalikan dengan 100, sehingga didapatkan suhu yang terukur.

3.16 Rangkaian Keseluruhan

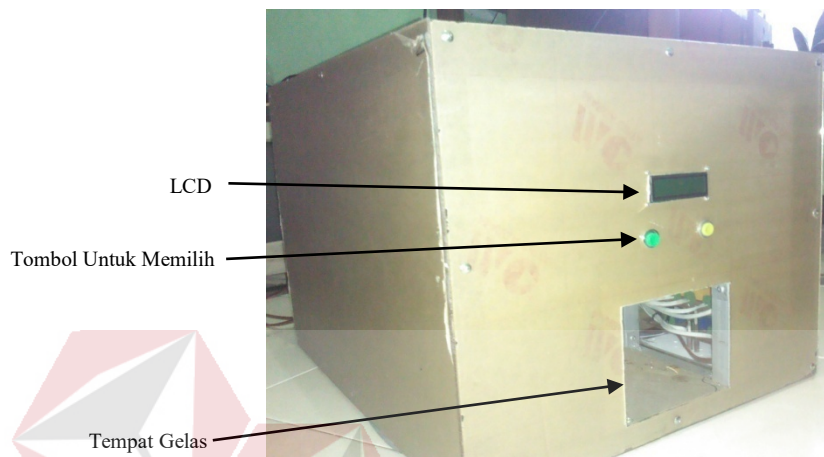
Pada Gambar 3.10 untuk penggunaan pin pada mikrokontroler sebagai input yaitu sensor suhu pada PORTA.0 yang berfungsi sebagai ADC mikrokontroler. Untuk inputan lainnya yaitu tombol untuk pemilihan menu dan jumlah takaran berada pada PORTB.0 dan PORTB.1. Sedangkan untuk output yaitu LCD pada PORTC.0 - PORTC.7, *driver relay heater* pada PORTD.0 - PORTD.7, *driver relay motor penguang gula* dan *motor penguang kopi* pada PORTB.3 dan PORTB.4, *driver relay washer pump* pada PORTB.2.



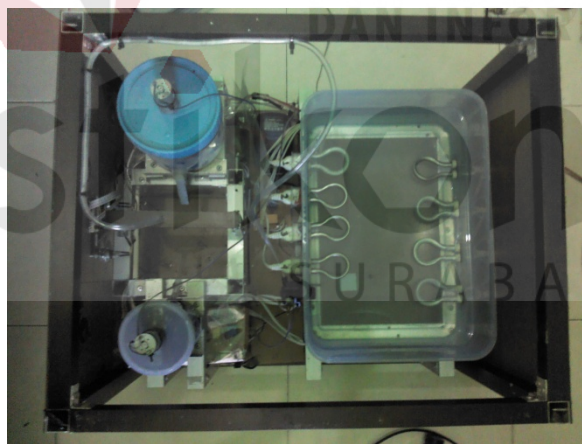
Gambar 3.10 Rangkaian Keseluruhan Sistem

3.2 Perancangan Mekanik

Desain mekanik terdiri dua motor, *washer pump*, *heater* dan kerangka yang berbahan aluminium. Adapun bentuk keseluruhan dari mekanik sebagai berikut :



Gambar 3.11 Mekanik tampak depan



Gambar 3.12 Mekanik Tampak Atas

Untuk *heater* dan *washer pump* langsung ditempatkan pada tempat air seperti pada Gambar 3.13 dan Gambar 3.15, sedangkan untuk penempatan sensor suhu diletakkan pada tutup tempat air tersebut seperti pada dan Gambar 3.14.



Gambar 3.13 Penempatan *Heater*



Gambar 3.14 Penempatan Sensor Suhu

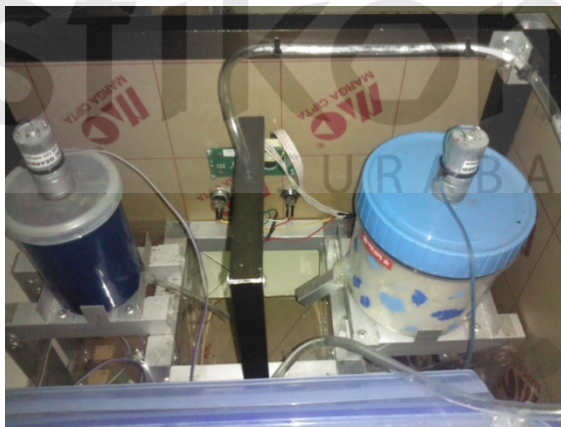


Gambar 3.15 Penempatan *WasherPump*

Mekanik motor penuang gula dan penuang kopi diletakkan pada tutup tempat gula dan kopi itu sendiri, seperti pada Gambar 3.16. Sedangkan untuk posisi tempat gula dan kopi pada Gambar 3.17.



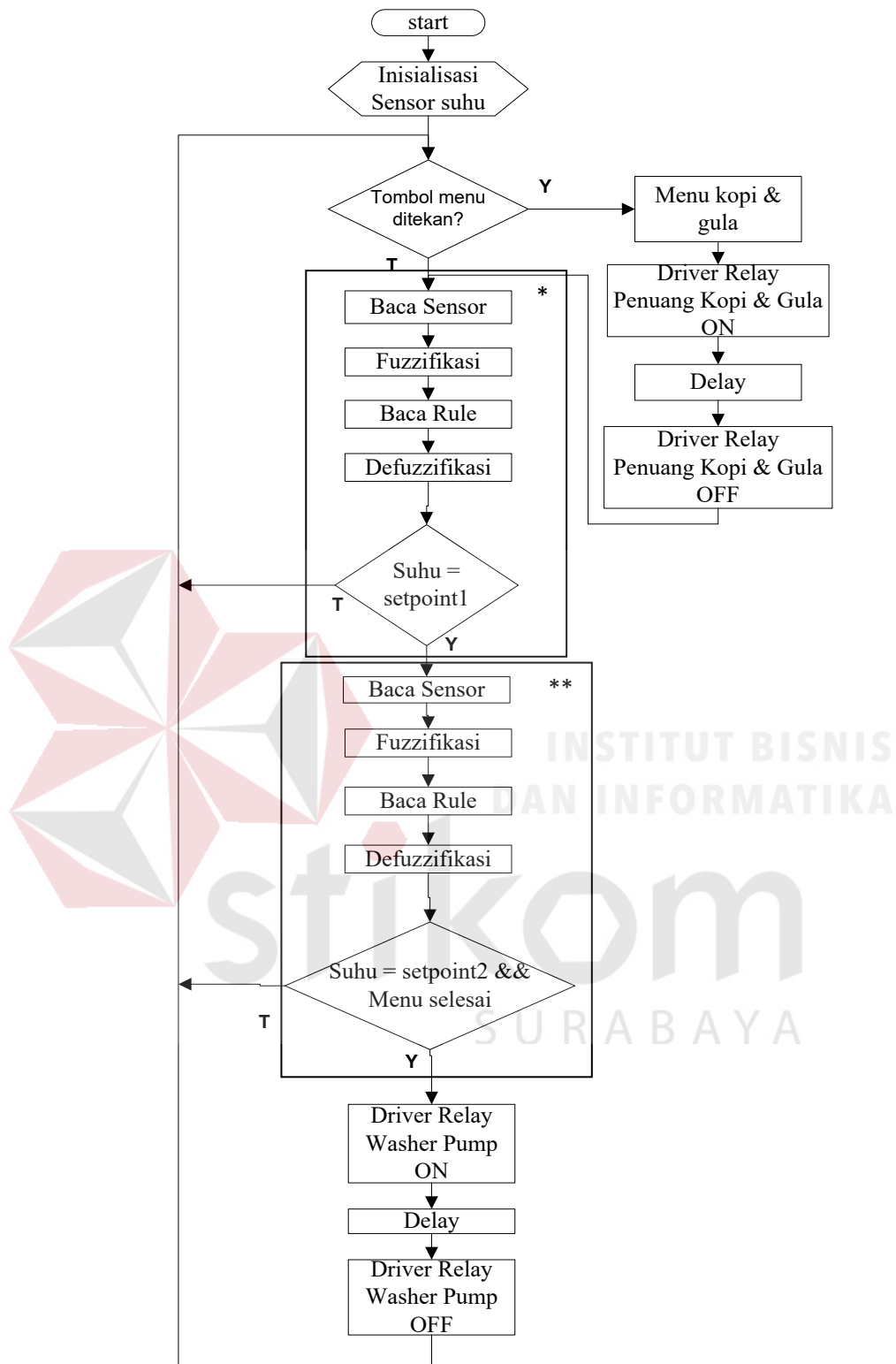
Gambar 3.16 Mekanik Penuang Gula dan Kopi



Gambar 3.17 Posisi Tempat Gula dan Kopi

3.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak bertujuan agar sistem yang telah dibuat dapat berjalan sesuai dengan yang diinginkan. Minimum sistem memperoleh data dari *push button* berupa banyaknya takaran gula dan kopi yang diinginkan, selain itu minimum sistem juga memperoleh data dari sensor suhu berupa suhu air saat normal hingga air ketika dipanaskan. Minimum sistem juga mengatur banyaknya heater yang digunakan, dan juga mengatur lama putaran motor penuang gula dan kopi. Perancangan perangkat lunak terbagi menjadi beberapa bagian program yaitu program pembacaan sensor, program algoritma fuzzy. Diagram alir perangkat lunak secara umum dapat dilihat pada Gambar 3.18. Berdasarkan pada diagram alir terdapat dua set point yaitu untuk set point pertama digunakan untuk proses memanaskan air hingga mendidih kemudian setelah setelah air mendidih melakukan proses menstabilkan suhu air sesuai dengan set point yang kedua. Dalam proses memanaskan dan menstabilkan suhu air akan selalu melakukan pendeteksian tombol menu, apabila tombol menu ditekan maka akan melakukan proses menampilkan menu kopi dan gula hingga proses menakar. Sedangkan apabila selama proses memanaskan dan menstabilkan suhu air tidak melakukan penekanan tombol menu maka proses memanaskan dan menstabilkan suhu air dilakukan berulang-ulang. Jika suhu sudah sesuai dengan set point yang kedua maka air telah siap digunakan untuk menyeduh kopi.



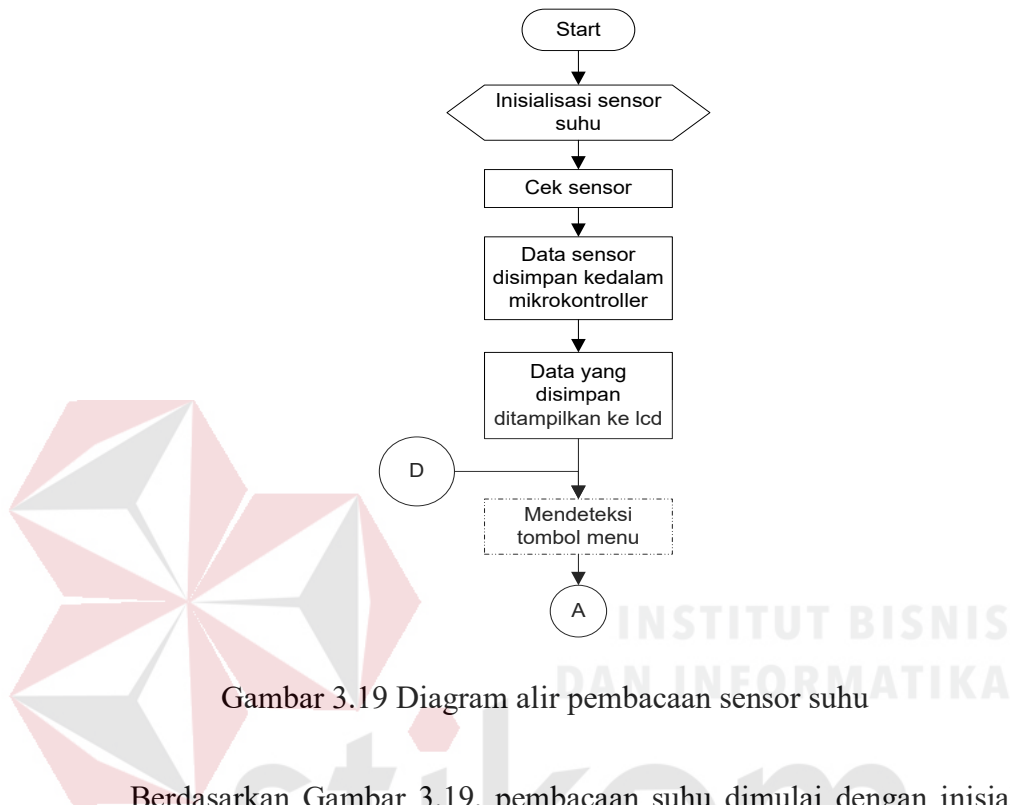
Gambar 3.18 Diagram alir program secara umum pada *microcontroller*

*Proses memanaskan air

**Proses menstabilkan suhu air

3.3.1 Perancangan Perangkat Lunak Untuk Sensor Suhu

Diagram alir untuk mengetahui suhu yang terbaca melalui sensor suhu dan akan ditampilkan ke LCD terdapat pada Gambar 3.19.



Gambar 3.19 Diagram alir pembacaan sensor suhu

Berdasarkan Gambar 3.19, pembacaan suhu dimulai dengan inisialisasi sensor suhu. Kemudian data akan disimpan ke dalam *microcontroller* dan ditampilkan ke LCD. Berikut adalah potongan program pembacaan sensor suhu:

```

unsigned int lm()
{
  unsigned int temp;
  float vin;
  int suhu;
  unsigned char lcd_buffer[10],adc[10];
  temp=read_adc(0);
  vin=((float)temp*0.01960);
  suhu=vin*100;
  lcd_clear();
  ftoa(vin,4,lcd_buffer);
  lcd_gotoxy(0,0);
  lcd_puts(lcd_buffer);
  lcd_gotoxy(8,0);
  lcd_puts("Suhu:");
  itoa(suhu,adc);
  lcd_gotoxy(13,0);
  lcd_puts(adc);
  return suhu;
}

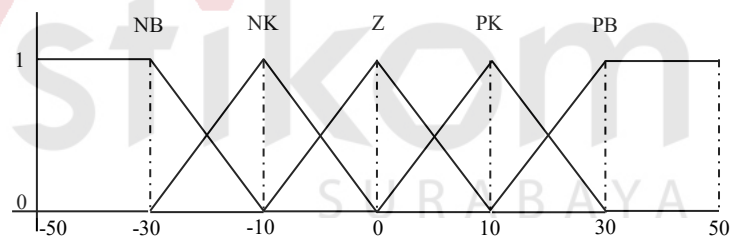
```

3.3.2 Fuzzy

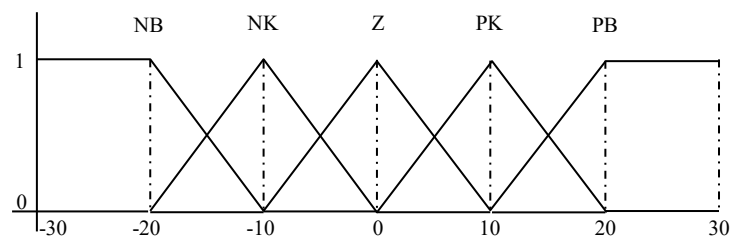
Untuk menentukan jumlah heater yang digunakan untuk memanaskan air dengan metode *fuzzy* terdapat tiga proses, yaitu :

1. Fuzzifikasi

Pada proses fuzzifikasi ini terjadi pengambilan keputusan dengan cara memetakan error dan delta error ke dalam data *fuzzy* sesuai dengan tipe dan bentuk fungsi keanggotaan. Digunakannya error sebagai inputan adalah untuk mengetahui seberapa besar jarak antara suhu yang terukur oleh sensor suhu dengan suhu pada *set point*, sedangkan delta error digunakan untuk mengetahui berapa besar dan cepatnya suhu mengalami kenaikan atau mengalami penurunan setiap kali terukur oleh sensor suhu hingga mencapai *set point*. Penentuan dari fungsi keanggotaan error dan delta error dapat dilihat pada Gambar 3.20 dan Gambar 3.21.



Gambar 3.20 Pemetaan Derajat Keanggotaan Error



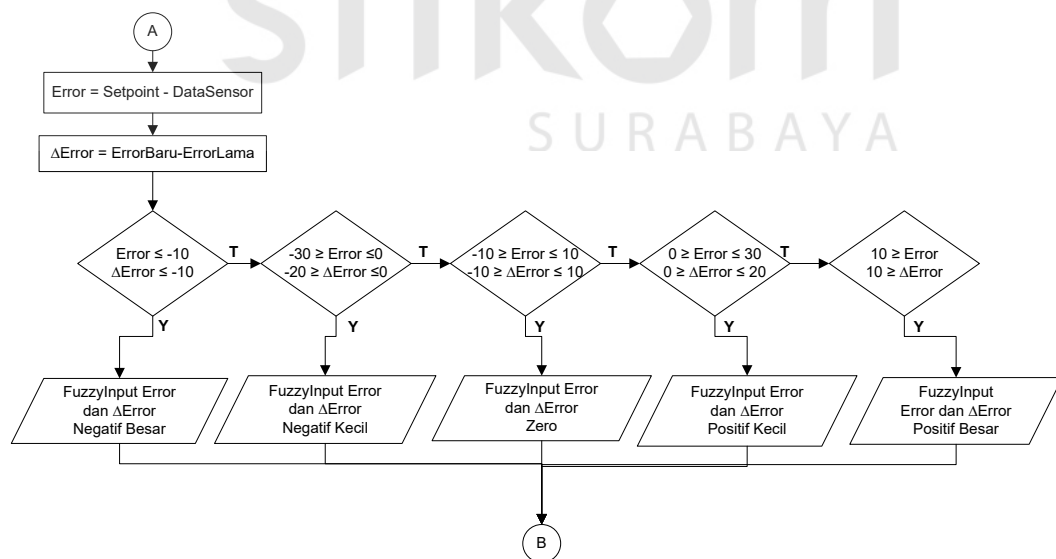
Gambar 3.21 Pemetaan Derajat Keanggotaan Delta Error

Pada saat proses fuzzifikasi, pendeteksian suhu melalui sensor lm35 dilakuakn terlebih dahulu, selanjutnya data hasil pendeteksian tersebut akan dipetakan sesuai fungsi keanggotaan *fuzzy* yang ada. Rumus yang ditentukan adalah sebagai berikut :

$$Error = Setpoint - sensor \dots \dots \dots (3.3)$$

$$\Delta Error = ErrorBaru - ErrorLama \dots \dots \dots (3.4)$$

Rumus (3.3) diperoleh setelah mendapatkan inputan dari sensor suhu berupa suhu yang terukur. Rumus (3.4) diperoleh setelah perhitungan *error* dilakukan kemudian dicari selisihnya dengan *error* yang baru. Pada Gambar 3.22 merupakan flowchart proses fuzzikasi yang dilakukan pada sistem. Dimana setelah dilakukan perhitungan sesuai dengan rumus (3.3) dan (3.4) maka tahapan selanjutnya dilakukan pengelompokan sesuai dengan fungsi keanggotaan dari *error* dan *delta error* yang nantinya akan didapatkan berupa fuzzy input berdasarkan fungsi anggota masing-masing.



Gambar 3.22 Flowchart Proses Fuzzifikasi dalam Memanaskan dan Menstabilkan Suhu Air

Realisasi proses memperoleh nilai derajat keanggotaan dari masing-masing fungsi keanggotaan adalah sebagai berikut :

```

error = setpoint - sensor;
derror = error - error_lama;

//-----error
dke[0]=anggotal(-50,-50,-30,-10,error);
dke[1]=anggotal(-30,-10,-10,0,error);
dke[2]=anggotal(-10,0,0,10,error);
dke[3]=anggotal(0,10,10,30,error);
dke[4]=anggotal(10,30,50,50,error);

//-----derror
dkd[0]=anggota2(-30,-30,-20,-10,derror);
dkd[1]=anggota2(-20,-10,-10,0,derror);
dkd[2]=anggota2(-10,0,0,10,derror);
dkd[3]=anggota2(0,10,10,20,derror);
dkd[4]=anggota2(10,20,30,30,derror);

error_lama = error;

float anggotal(float a, float b, float c, float d, float error)
{
    float u1;

    if ((a== -50) && (d== -10)) //nb
    {
        if ((error>=c) && (error<=d))
        {
            u1 = (d - error) / (d - c);
        }
        else if ((error>=a) && (error<=c))
        {
            u1 = 1;
        }
        else if (error<=a)
        {
            u1=1;
        }
        else{u1=0;};
    }
    else if ((a== -30) && (d==0)) //nk
    {
        if ((error>c) && (error<d))
        {
            u1 = (d - error) / (d - c);
        }
        else if ((error < b) && (error > a))
        {
            u1 = (error - a) / (b - a);
        }
        else if ((error==b) && (error == c))
        {
            u1=1;
        }
        else
        {u1=0;};
    }
    else if ((a== -10) && (d==10)) //z
    {
        if ((error>c) && (error<d))
        {
            u1 = (d - error) / (d - c);
        }
    }
}

```

```

    }
    else if((error <b) && (error >a))
    {
        u1 = (error - a) / (b - a);
    }
    else if ((error==b) && (error == c))
    {
        u1=1;
    }
    else
    {u1=0;};
}
else if((a==0)&&(d==30))          //pk
{
    if((error>c)&&(error<d))
    {
        u1 = (d - error) / (d - c);
    }
    else if((error < b) && (error > a))
    {
        u1 = (error - a) / (b - a);
    }
    else if ((error==b) && (error == c))
    {
        u1=1;
    }
    else
    {u1=0;};
}
else if((a==10)&&(d==50))        //pb
{
    if((error>=a)&&(error<=b))
    {
        u1 = (error - a) / (b - a);
    }
    else if((error<=d)&&(error>=b))
    {
        u1 = 1;
    }
    else if(error>=d)
    {
        u1=1;
    }
    else{u1=0;};
}
else{u1=1;};
return u1;
}

float anggota2(float a2, float b2, float c2, float d2, float
derror)
{    float u2;

    if ((a2==--30)&&(d2==--10))    //nb
    {    if((derror>=c2)&&(derror<=d2))
        {
            u2 = (d2 - derror) / (d2 - c2);
        }
        else if((derror>=a2)&&(derror<=c2))
        {
            u2 = 1;
        }
        else if(derror<=a2)
        {

```

```

        u2=1;
    }
    else{u2=0;};
}
else if((a2==-20)&&(d2==0)) //nk
{
    if((derror>c2)&&(derror<d2))
    {
        u2 = (d2 - derror) / (d2 - c2);
    }
    else if((derror <b2) && (derror >a2))
    {
        u2 = (derror - a2) / (b2 - a2);
    }
    else if ((derror==b2) && (derror == c2))
    {
        u2=1;
    }
    else
    {u2=0;};
}
else if((a2==-10)&&(d2==10)) //z
{
    if((derror>c2)&&(derror<d2))
    {
        u2 = (d2 - derror) / (d2 - c2);
    }
    else if((derror <b2) && (derror >a2))
    {
        u2 = (derror - a2) / (b2 - a2);
    }
    else if ((derror==b2) && (derror == c2))
    {
        u2=1;
    }
    else
    {u2=0;};
}
else if((a2==0)&&(d2==20)) //pk
{
    if((derror>c2)&&(derror<d2))
    {
        u2 = (d2 - derror) / (d2 - c2);
    }
    else if((derror <b2) && (derror >a2))
    {
        u2 = (derror - a2) / (b2 - a2);
    }
    else if ((derror==b2) && (derror == c2))
    {
        u2=1;
    }
    else
    {u2=0;};
}
else if((a2==10)&&(d2==30)) //pb
{
    if((derror>=a2)&&(derror<=b2))
    {
        u2 = (derror - a2) / (b2 - a2);
    }
    else if((derror>=b2)&&(derror<=d2))
    {
        u2 = 1;
    }
}

```

```

else if (derror>=d2)
{
    u2=1;
}
else{u2=0;};
}
else{u2=1;};
return u2;
}

```

2. Rule Set

Rule set / evaluasi aturan adalah proses mengevaluasi derajat keanggotaan tiap-tiap fungsi keanggotaan himpunan *fuzzy* masukan ke dalam basis aturan yang telah ditetapkan. Tujuan dari evaluasi aturan ini adalah menentukan derajat keanggotaan dari keluaran *fuzzy*. Sebelum melakukan evaluasi aturan terlebih dahulu ditetapkan basis aturan. Basis aturan merupakan keseluruhan aturan dari kombinasi dua masukan yang mungkin. Secara lengkap, jumlah kombinasi yang mungkin dari dua himpunan *fuzzy* masukan masing-masing lima fungsi keanggotaan sehingga jumlah aturannya adalah dua puluh lima aturan. Dua puluh lima aturan kendali *fuzzy* dapat dilihat pada tabel 3.1.

Tabel 3.1 Penentuan *Rule Set*

		Delta Error				
		NB	NK	Z	PK	PB
Error	NB	OFF	OFF	OFF	OFF	OFF
	NK	OFF	OFF	OFF	OFF	OFF
	Z	L1	L1	OFF	L1	L1
	PK	L3	L2	L4	L1	L1
	PB	L4	L4	L4	L3	L3

Keterangan:

NB : Negatif besar

NK : Negatif kecil

Z : Zero

PK : Positif Kecil

PB : Positif Besar

L1 : Level 1

L2 : Level 2

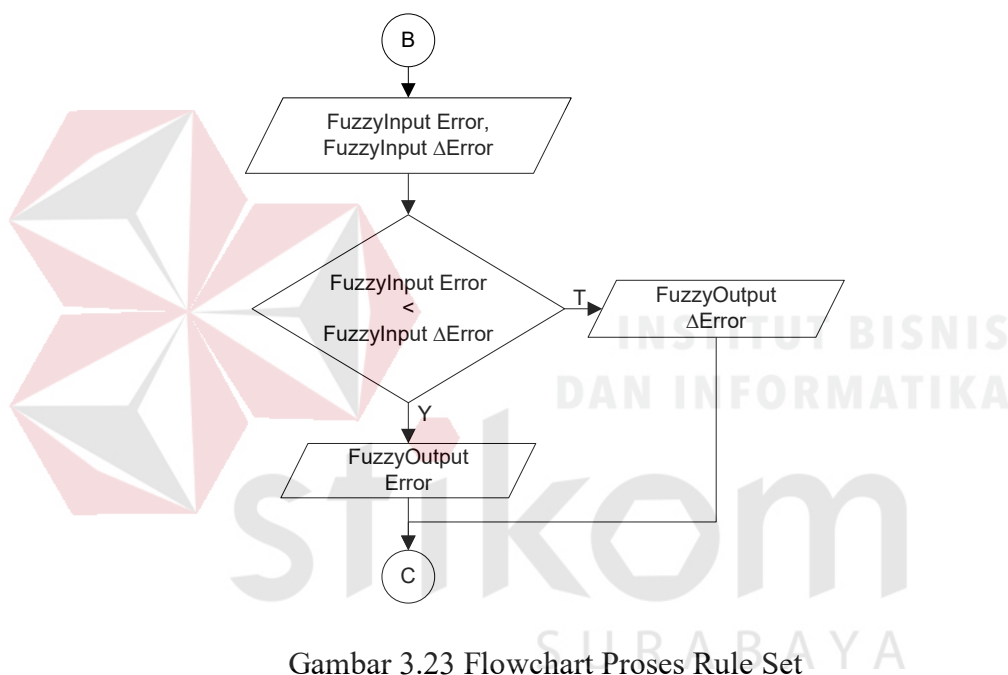
L3 : Level 3

L4 : Level 4

Untuk penentuan *Rule Set* dilakukan dengan cara melihat kondisi *error* terlebih dahulu kemudian melihat kondisi *delta error*. Ketika *error* pada kondisi negatif, baik negatif kecil maupun besar yaitu saat suhu yang terukur oleh sensor sudah melebihi *set point* yang ditentukan maka kondisi apapun *delta error* pemanasan air tidak dilakukan atau di set OFF. Untuk kondisi *error* pada kondisi positif kecil dan *delta error* pada kondisi *zero*, dimana suhu mendekati set point tetapi tidak mengalami perubahan maka perlu adanya tingkat pemanasan air yang lebih tinggi dari kondisi pemanasan air yang awal maka di set L4. Untuk kondisi *error* pada positif besar dan *delta error* pada kondisi positif besar maupun positif kecil dimana suhu masih berada jauh dari set point tetapi suhu berikutnya yang terbaca oleh sensor mengalami penurunan maka pemanas air di set pada level 3 atau L3. Untuk *error* pada kondisi *zero* dan *delta error* saat kondisi negatif besar maupun negatif kecil maka pemanas di set pada level 1 atau

L1. Sedangkan untuk *error* pada kondisi positif kecil lalu *delta error* pada kondisi negatif kecil yang mana suhu sudah mendekati set point dan perubahan suhunya pelan atau kecil maka pemanas di set pada level 2 atau L2.

Setelah proses fuzzifikasi dieksekusi, maka dilakukan proses agregasi dengan mengambil nilai minimum dari masing-masing fungsi keanggotaan variabel keluaran seperti pada Gambar 3.22.



Gambar 3.23 Flowchart Proses Rule Set

Realisasi proses evaluasi aturan dalam bentuk program adalah

sebagai berikut :

```

h=0;
for (i=0; i<5; i++)
{
    for (j=0; j<5; j++)
    {
        min_u[h]=rule(dke[i],dkd[j]);
        h++;
    };
};
float rule(float a, float b)
{ float min;

```

```

if (a<b)
{
  min=a;
}
else
{
  min=b;
};
return min;
}

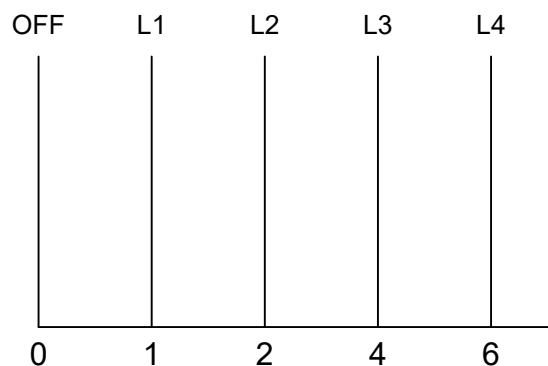
```

3. Defuzzifikasi

Defuzzifikasi adalah kebalikan dari proses fuzzifikasi, yaitu mengubah himpunan *fuzzy* keluaran menjadi keluaran tegas (*crisp*). Pengubahan nilai ini diperlukan karena konstanta kendali *fuzzy* hanya mengenal nilai tegas sebagai variabel sinyal kontrol. Hasil keluaran *crisp output* akan digunakan sebagai penentu jumlah pemanas (*heater*) yang akan digunakan. Realisasi proses pengambilan keputusan metode sugeno menjadi bentuk *crisp out* dalam bentuk rumus sebagai berikut :

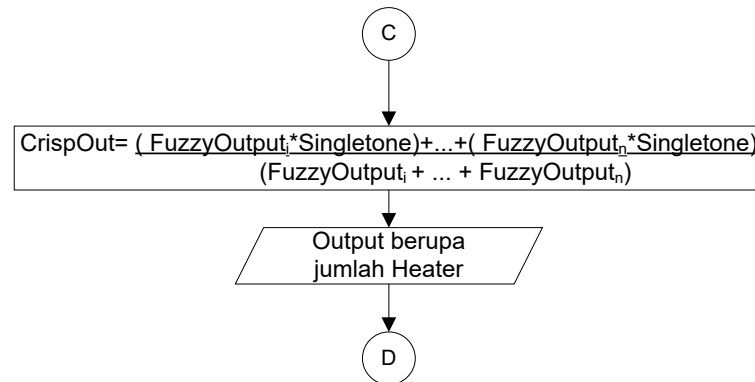
$$Crisp_out = \frac{\sum_{i=0}^n FuzzyOutput_i \times (PosisiSingleton \times axis_i)}{\sum_{i=0}^n FuzzyOutput_i} \dots\dots\dots(3.5)$$

Dibawah ini adalah tampilan fungsi keanggotaan dari proses defuzzifikasi :



Gambar 3.24 Fungsi keanggotaan singleton

Realisasi proses pengambilan keputusan metode sugeno menjadi bentuk *crisp out* dalam bentuk flowchart terdapat pada Gambar 3.24.



Gambar 3.25 Flowchart Proses Defuzzifikasi

Sedangkan untuk proses pengambilan keputusan metode sugeno menjadi bentuk *crisp out* dalam bentuk program adalah sebagai berikut :

```

total_u=0;
for (i=0; i<25; i++)
{
    total_u = total_u + min_u[i];
};

off=0;
L1=1;
L2=2;
L3=4;
L4=6;

max_f1[0]=op(min_u[0],min_u[1]);
max_f1[1]=op(min_u[2],min_u[3]);
max_f1[2]=op(min_u[4],min_u[5]);
max_f1[3]=op(min_u[6],min_u[7]);
max_f1[4]=op(min_u[8],min_u[9]);
max_f1[5]=op(max_f1[0],min_u[12]);

max_f2[0]=op(max_f1[1],max_f1[2]);
max_f2[1]=op(max_f1[3],max_f1[4]);
max_f2[2]=op(max_f1[5],max_f2[0]);
max_f2[3]=op(max_f2[1],max_f2[2]);
max_u[0]=max_f2[3]*off;

//-----L1
l=10;
k=13;
for(i=0; i<2; i++)
{
    maxl1_1[i]=op(min_u[l],min_u[k]);
    l=l+1;
    k=k+1;
};
maxl1_1[2]=op(min_u[18],min_u[19]);
maxl1_1[3]=op(maxl1_1[0],maxl1_1[1]);
if(maxl1_1[2]>maxl1_1[3])
{
    max_u[1]=maxl1_1[2]*L1;
  
```

```

}
else
{
    max_u[1]=maxl1_1[3]*L1;
};

//-----L2
max_u[2]=min_u[16]*L2;

//-----L3
maxl3_1[0]=op(min_u[23],min_u[24]);
if(maxl3_1[0]> min_u[15])
{
    max_u[3]=maxl3_1[0]*L3;
}
else
{
    max_u[3]=min_u[15]*L3;
};

//-----L4
maxl4_1[0]=op(min_u[17],min_u[22]);
maxl4_1[1]=op(min_u[20],min_u[21]);

if (maxl4_1[0] > maxl4_1[1])
{
    max_u[4] = maxl4_1[0]*L4;
}
else
{
    max_u[4] = maxl4_1[1]*L4;
};

pembilang=0;
for(j=0; j<5; j++)
{
    pembilang = pembilang + max_u[j];
};
hasil = pembilang / total_u;

ftoa(pembilang,1,tamp);
lcd_gotoxy(0,1);
lcd_puts(tamp);

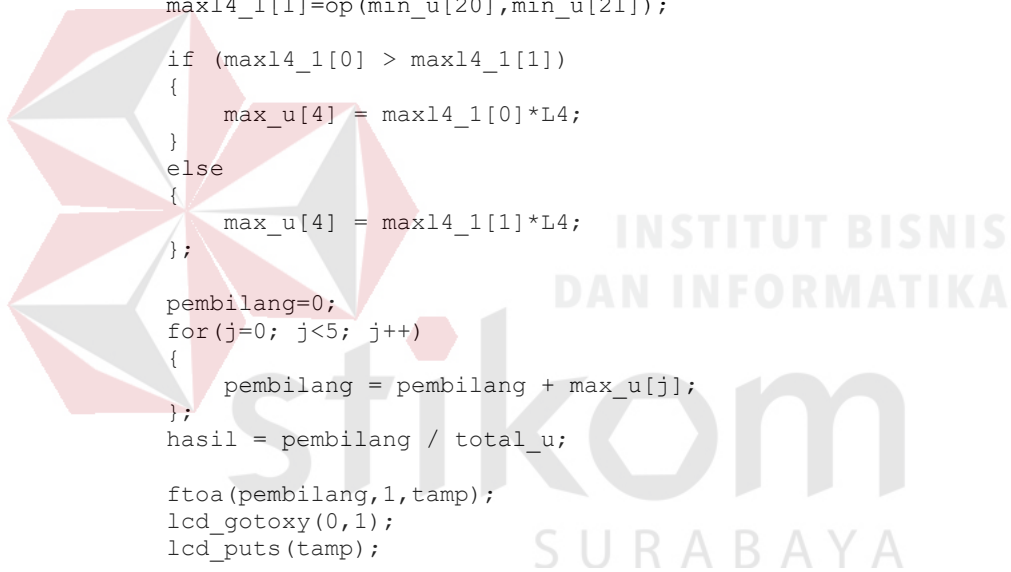
lcd_gotoxy(3,1);
lcd_puts("/");

ftoa(total_u,1,tamp2);
lcd_gotoxy(4,1);
lcd_puts(tamp2);

itoa(hasil,tamp3);
lcd_gotoxy(8,1);
lcd_puts(tamp3);

if((hasil>=0)&&(hasil<1))
{
    a=0;
}
else if((hasil>=1)&&(hasil<2))
{
    a=1;
}
else if((hasil>=2)&&(hasil<3))
{

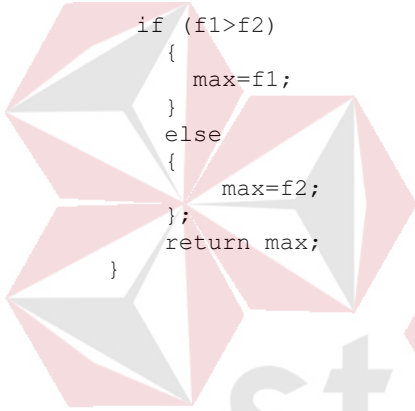
```



```
        a=2;
    }
    else if((hasil>=3)&&(hasil<4))
    {
        a=3;
    }
    else if((hasil>=4)&&(hasil<5))
    {
        a=4;
    }else if((hasil>=5)&&(hasil<6))
    {
        a=5;
    }
    else if((hasil>=6)&&(hasil<7))
    {
        a=6;
    };

    relay();
}
float op(float f1, float f2)
{ float max;
```

```
    if (f1>f2)
    {
        max=f1;
    }
    else
    {
        max=f2;
    };
    return max;
}
```



INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

BAB IV

PENGUJIAN SISTEM

Pengujian sistem yang dilakukan penulis merupakan pengujian terhadap perangkat keras dan perangkat lunak dari sistem secara keseluruhan yang telah selesai dibuat untuk mengetahui komponen-komponen sistem apakah berjalan dengan baik sesuai yang diharapkan. Terdapat beberapa pengujian sistem, antara lain:

4.1 Pengujian *Minimum system*

4.1.1 Tujuan

Pengujian *minimum system* bertujuan untuk mengetahui apakah *minimum system* dapat melakukan proses *signature* dan download program ke mikrokontroller dengan baik.

4.1.2 Alat yang digunakan

1. Rangkaian *minimum system* ATmega16
2. *Downloader*.
3. *Power supply* 12000mA-12V
4. Regulator +5V

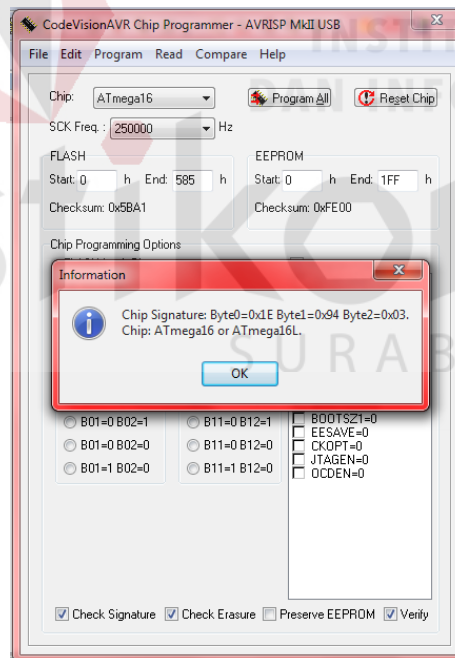
4.1.3 Prosedur Pengujian

1. Aktifkan *power supply* dan hubungkan dengan *minimum system*.
2. Sambungkan *minimum system* dengan kabel *downloader*.
3. Selanjutnya aktifkan PC dan jalankan program *CodeVisionAVR*.
4. Untuk download program yang telah dibuat ke dalam *minimum system* maka yang harus dilakukan adalah menjalankan menu *chip signature programmer* pada *CodeVisionAVR*.

- Setelah proses *signature* selesai maka selanjutnya proses *compile project* dengan menekan F9 pada *keyboard* kemudian proses *download* program ke mikrokontroler masuk ke *menu* → *make project* pada CodeVisionAVR.

4.1.4 Hasil Pengujian

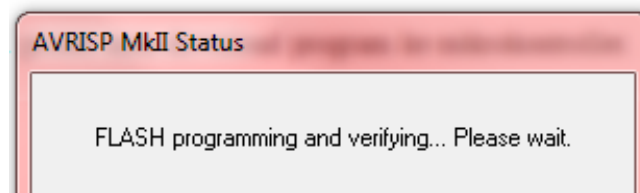
Dari percobaan di atas apabila menu *chip signature programmer*, *download* program dapat berhasil dikerjakan maka *minimum system* dapat dikatakan bekerja dengan baik. Tampilan dari program *chip signature* pada CodeVisionAVR yang akan digunakan untuk menuliskan program dan melakukan percobaan terhadap *minimum system*. Hasil program *chip signature* dapat dilihat pada Gambar 4.1 berikut:



Gambar 4.1 Tampilan *Chip Signature*

Berdasarkan pada Gambar 4.1, setelah dilakukan pengujian dengan cara *chip signature*, *minimum system* dapat membaca mikrokontroler yang digunakan,

yaitu mikrokontroller ATmega16. Pada gambar 4.2 menunjukkan bahwa minimum system sedang melakukan proses men-download program ke mikrokontroller dan program dari komputer ke *minimum system* dapat di download dengan baik.



Gambar 4.2 Tampilan Download Program

4.2 Pengujian Sensor Suhu

4.2.1 Tujuan

Pengujian sensor suhu dilakukan untuk mengetahui suhu air ketika air dipanaskan. Data yang dihasilkan oleh sensor kemudian diolah oleh mikrokontroller lalu ditampilkan ke LCD.

4.2.2 Alat yang digunakan

1. Rangkaian *minimum system* ATmega16.
2. *Power supply* 1200mA - 12V.
3. Regulator +5V.
4. Sensor suhu LM35.
5. Multitester digital.

4.2.3 Prosedur Pengujian

Langkah-langkah untuk melakukan pengujian pembacaan sensor suhu adalah sebagai berikut:

1. Hubungkan sensor suhu dengan *minimum system* ATmega16
2. Aktifkan *power supply* dan hubungkan dengan regulator serta *minimum system* ATmega16.
3. Panaskan air dan masukkan sensor suhu ke dalam air yang dipanaskan.
4. Bandingkan keluaran suhu dengan termometer digital untuk mengetahui akurasi sensor dalam mengukur suhu.

4.2.4 Hasil Pengujian

Hasil pengujian sensor suhu dapat dilihat pada Tabel 4.1 dan Gambar 4.3.

adapun cuplikan program yang digunakan dalam pengujian sensor suhu yaitu:

```

unsigned int lm()
{
    unsigned int temp;
    float vin;
    int suhu;
    unsigned char lcd_buffer[10],adc[10];

    temp=read_adc(0);
    vin=((float)temp*0.01960);
    suhu=vin*100;

    lcd_clear();
    ftoa(vin,4,lcd_buffer);
    lcd_gotoxy(0,0);
    lcd_puts(lcd_buffer);

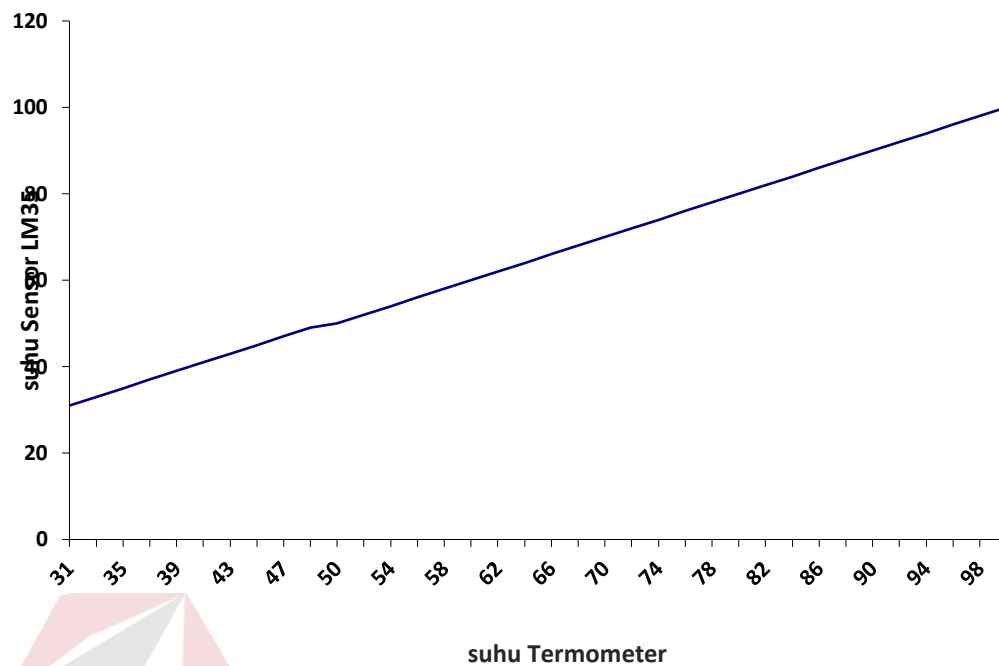
    lcd_gotoxy(8,0);
    lcd_puts("Suhu:");
    itoa(suhu,adc);
    lcd_gotoxy(13,0);
    lcd_puts(adc);

    return suhu;
}

```

Tabel 4.1 Hasil pengukuran sensor suhu

Suhu (°C)	V _{out} sensor suhu LM35	
	°C	mV
31	31	329
33	33	348
35	35	366
37	37	387
39	39	407
41	41	428
43	43	448
45	45	466
47	47	486
49	49	507
50	50	517
52	52	536
54	54	559
56	56	578
58	58	595
60	60	616
62	62	639
64	64	656
66	66	676
68	68	694
70	70	712
72	72	734
74	74	753
76	76	776
78	78	795
80	80	814
82	82	835
84	84	855
86	86	874
88	88	895
90	90	909
92	92	937
94	94	956
96	96	977
98	98	994
100	100	1 Volt



Gambar 4.3 Grafik Hasil Pengujian Sensor Suhu

Berdasarkan pada Tabel 4.1 dan Gambar 4.3 hasil dari pengujian sensor suhu LM35 didapatkan bahwa keluaran sensor suhu LM35 dalam mengukur suhu air menghasilkan suhu yang sesuai dengan suhu yang terukur oleh termometer dan suhu yang terukur oleh sensor suhu LM35 setiap 2°C.

4.3 Pengujian motor penuang gula dan kopi

4.3.1 Tujuan

Pengujian ini dilakukan untuk mengetahui apakah takaran yang dihasilkan oleh motor penuang gula dan motor penuang kopi sudah sesuai dengan takaran sendok yang sebenarnya.

4.3.2 Alat yang Digunakan

1. Rangkaian *minimum system* ATmega16.
2. *Power supply* 1200mA - 12V.
3. Regulator +5V.
4. Motor DC.
5. Stopwatch.

4.3.3 Prosedur Pengujian

1. Aktifkan power supply dan hubungkan dengan regulator serta minimum system ATmega16.
2. Hubungkan driver relay motor penuang gula dan motor penguang kopi ke minimum system ATmega16.
3. Amati dan hitung waktu yang dibutuhkan untuk menakar gula dan kopi dengan menggunakan stopwatch.

4.3.4 Hasil Pengujian

Dari pengujian yang dilakukan, maka didapatkan lama waktu putaran motor untuk menakar gula dan kopi setiap 4 gram. Adapun hasilnya pada tabel 4.2 dan tabel 4.3.

Tabel 4.2 Hasil Pengujian Kopi

Pengujian Kopi ke-	Waktu(detik) / 4 gram
1	18 detik/4gram
2	18 detik/4gram
3	18 detik/4gram
4	18 detik/4gram
5	17 detik/4gram

Tabel 4.3 Hasil Pengujian Gula

Pengujian Kopi ke-	Waktu(detik) / 4 gram
1	9 detik/4gram
2	9 detik/4gram
3	8 detik/4gram
4	8 detik/4gram
5	7 detik/4gram

4.4 Pengujian Tombol Menu

4.4.1 Tujuan

Pengujian ini dilakukan untuk mengetahui tombol dapat bekerja dengan baik, sehingga saat digunakan untuk memilih menu dapat berjalan normal.

4.4.2 Alat yang digunakan

1. Rangkaian *minimum system* ATmega16.
2. *Power supply* 1200mA - 12V.
3. Regulator +5V.
4. Push button.
5. LCD 16x2.

4.4.3 Prosedur pengujian

1. Aktifkan *power supply* dan hubungkan dengan regulator serta *minimum system* ATmega16.
2. Hubungkan pushbutton ke *minimum system* ATmega16.
3. Tekan tombol TIDAK, lalu amati jumlah takaran gula dan kopi yang tampil pada LCD dengan jumlah takaran sebelum penekakan tombol TIDAK.
4. Kemudian tekan tombol YA, lalu amati proses penakaran gula dan kopi sesuai dengan pilihan yang tampil pada LCD.





4.4.4 Hasil pengujian

Untuk pengujian tombol pada menu kopi dapat dilihat pada tabel 4.4, dimana sebelum tombol ditekan muncul menu kopi dengan takaran 1 sendok dan setelah penekanan tombol TIDAK maka jumlah takaran berganti dengan jumlah takaran berikutnya. Tetapi, saat tombol YA ditekan, alat menakar sesuai dengan pilihan takaran user.





Sedangkan untuk pengujian tombol pada menu gula dapat dilihat pada tabel 4.4, dimana sebelum tombol ditekan muncul menu gula dengan takaran 1 sendok dan setelah penekanan tombol TIDAK maka jumlah takaran berganti dengan jumlah takaran berikutnya. Saat tombol YA ditekan, alat menakar sesuai dengan pilihan takaran user kemudian berganti ke proses memanaskan dan menstabilkan suhu air.

Setelah dilakukan pengujian tombol menu, dapat disimpulkan bahwa tombol menu sudah berfungsi sesuai dengan yang diharapkan. Dimana saat tombol TIDAK ditekan, jumlah takaran bisa berganti ke jumlah takaran berikutnya dan ketika tombol YA ditekan, melakukan proses menakar gula dan kopi sesuai dengan pilihan dari user.

Tabel 4.4 Hasil pengujian tombol menu kopi

Tombol	Kopi	
	Sebelum ditekan	Setelah ditekan
TIDAK		
YA		

Tabel 4.5 Hasil pengujian tombol menu gula

Tombol	Gula	
	Sebelum ditekan	Setelah ditekan
TIDAK		
YA		

4.5 Pengujian LCD

4.5.1 Tujuan

Pengujian ini dilakukan untuk mengetahui LCD dapat bekerja dengan baik dan dapat menampilkan *output* sesuai yang berasal dari mikrokontroler ATmega16.

4.5.2 Alat yang digunakan

Peralatan yang diperlukan untuk melakukan pengujian ini sebagai berikut:

1. Rangkaian *minimum system* ATmega16.
2. *Power supply* 1200mA – 12V.
3. Regulator +5V.
4. LCD 16x2.

4.5.3 Porsedur pengujian

1. Aktifkan *power supply* dan hubungkan dengan regulator serta *minimum system* ATmega16.
2. Hubungkan +5V pada *minimum system* ATmega16 pada (+) LCD dan hubungkan 0V pada *minimum system* ATmega16 pada (-) LCD.
3. Hubungkan PORT pada *minimum system* ATmega16 dengan pin LCD sesuai dengan tabel 4.6.

Tabel 4.6 Konfigurasi LCD 16x2 dengan *minimum system* ATmega16

Minimum system ATmega16	LCD 16x2
PORTC.0	RS
PORTC.1	R/W
PORTC.2	E
PORTC.4	DB4
PORTC.5	DB5
PORTC.6	DB6
PORTC.7	DB7

Download program berikut ke *minimum system* ATmega16 untuk menampilkan *output* ke LCD 16x2.

```
while (1)
{
    // Place your code here

    lcd_clear();
    lcd_gotoxy(0,0);
    lcd_puts("silahkan tunggu");
}
}
```

4.5.4 Hasil pengujian

Setelah dilakukannya pengujian pada LCD, maka dapat dilihat pada Gambar 4.3 yang merupakan hasil dari pengujian. Dimana LCD dapat menampilkan tulisan sesuai dengan program yang diinputkan ke *minimum sytem* ATmega16. Maka dengan ini LCD dapat berfungsi dengan benar.



Gambar 4.4 Tampilan LCD

4.6 Pengujian metode Fuzzy

4.6.1 Tujuan

Pengujian ini dilakukan untuk menguji apakah program fuzzy yang dibuat mampu berjalan sesuai dengan keinginan dan kebutuhan tugas akhir ini.

4.6.2 Alat yang digunakan

Peralatan yang diperlukan untuk melakukan pengujian ini adalah sebagai berikut:

1. Rangkaian *minimum system* ATmega16.
2. *Power supply* 1200mA – 12V.
3. Regulator +5V.
4. LCD 16x2.
5. Sensor suhu LM35.
6. Driver relay heater.

4.6.3 Prosedur pengujian

Langkah-langkah untuk melakukan pengujian metode fuzzy adalah sebagai berikut:

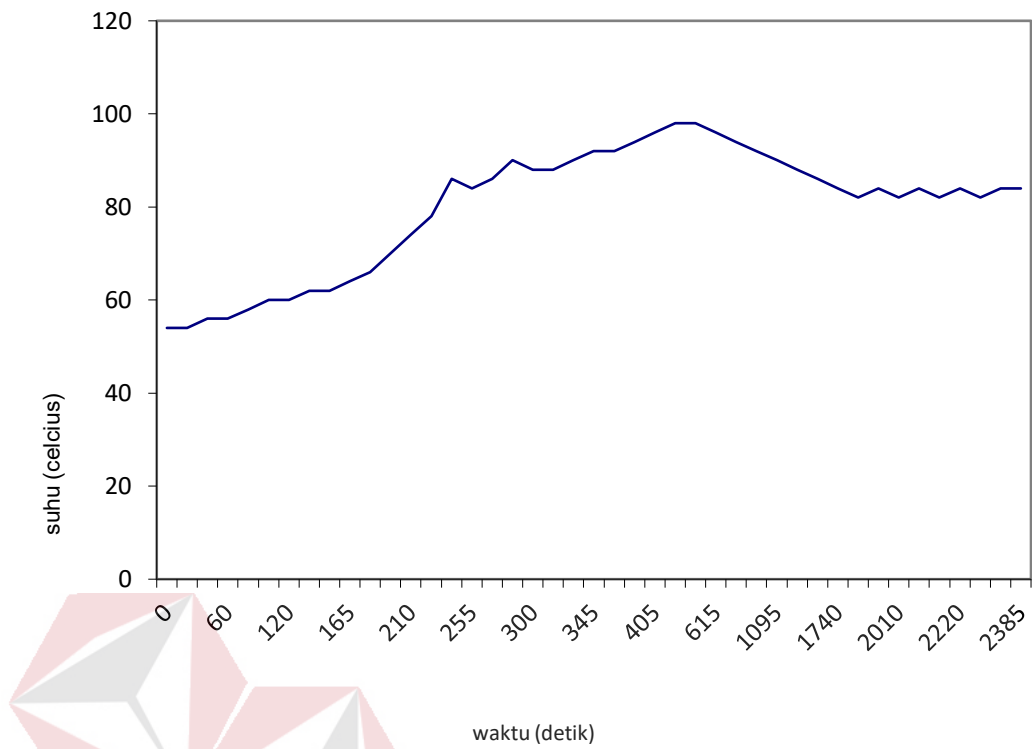
1. Hubungkan sensor suhu LM35, *driver relay heater* dan LCD dengan *minimum system*.
2. Aktifkan *power supply* dan hubungkan dengan regulator serta *minimum system*.
3. Amati data yang tampil pada LCD dan *driver relay heater*.
4. Lakukan beberapa kali untuk mengetahui hasil dari metode *fuzzy*.

4.6.4 Hasil pengujian

Dengan setpoint 85°C sistem mampu memanaskan air dari suhu awal 54°C hingga 98°C selama 10 menit 15 detik. Kemudian setelah air dipanaskan hingga mencapai suhu 98°C, suhu diturunkan dan mampu distabilkan oleh sistem pada suhu 84°C. Sehingga total waktu yang diperlukan oleh sistem dari suhu awal 54°C hingga proses penurunan suhu menuju setpoint 85°C memerlukan waktu 39 menit 45 detik. Hasil pengujian dengan setpoint 85°C dapat dilihat pada Tabel 4.7 dan Gambar 4.6.



Gambar 4.5 Hasil Pengujian saat Suhu 29°C



Gambar 4.6 Grafik Hasil Pengujian *Fuzzy* dengan Setpoint 85°C



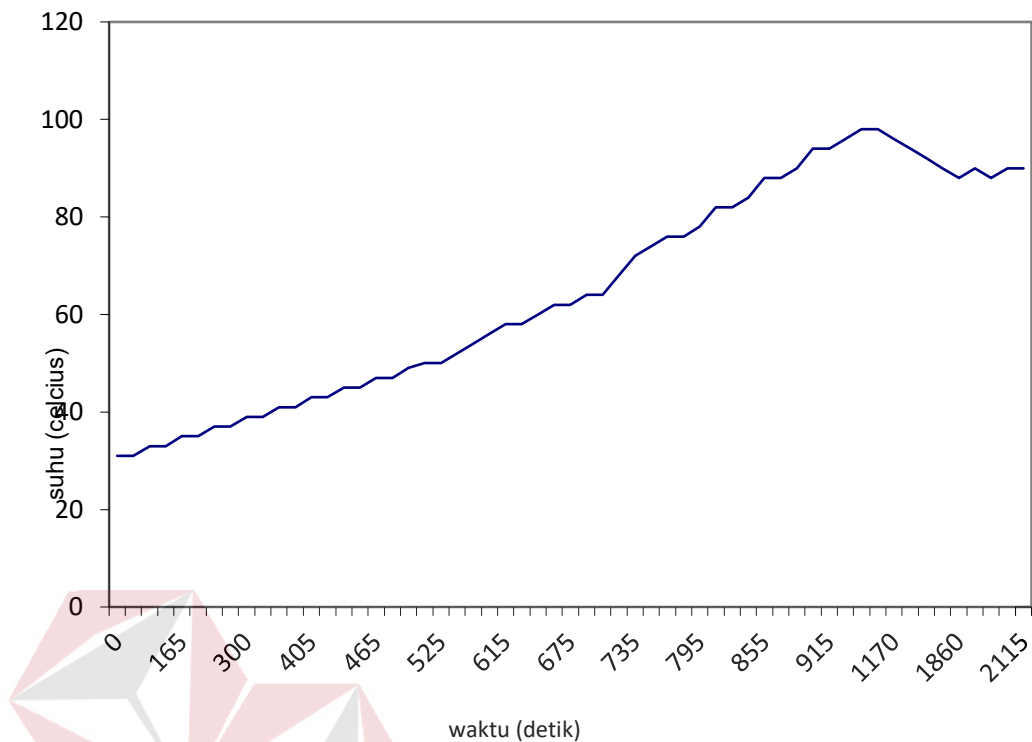
Tabel 4.8 Pengujian Metode Fuzzy dengan Setpoint 90°C

Suhu	Jumlah heater ON
31	4
31	6
33	4
33	6
35	4
35	6
37	4
37	6
39	4
39	6
41	4
41	6
43	4
43	6
45	4
45	6
47	4
47	6
49	4
50	5
50	6
52	4
54	6
56	4
58	4
58	6
60	4
62	4
62	6
64	4
64	6
68	3
72	3
74	3
76	3
76	4
78	2
82	2
82	3
84	3
88	3
88	5
90	5
94	2
94	3
96	2

Suhu	Jumlah heater ON
98	1
98	0
96	0
94	0
92	0
90	0
88	1
90	0
88	1
90	0

INSTITUT BISNIS
DAN INFORMATIKA

kom
SURABAYA



Gambar 4.7 Grafik Hasil Pengujian *Fuzzy* dengan Setpoint 90°C

Berdasarkan Tabel 4.8 dan Gambar 4.7, pengujian fuzzy dengan setpoint 90°C sistem dapat menstabilkan suhu air pada suhu 90°C. Dimana suhu awal air sebelum dipanaskan yaitu 31°C dan dipanaskan hingga mencapai suhu 98°C dengan memerlukan waktu 19 menit. Kemudian setelah suhu dipanaskan hingga mencapai 98°C, suhu mampu di turunkan hingga menuju setpoint 90°C. Ketika suhu berada pada suhu 88°C, sistem mampu menaikkan suhu air kembali hingga berada pada suhu 90°C sesuai dengan setpoint yang diinginkan. Berdasarkan dengan hasil pengujian dengan set point 85°C dan 90°C, metode fuzzy yang digunakan mampu memanaskan air hingga mendidih kemudian menurunkan suhu dan membuat stabil sesuai dengan set point yang diinginkan.



Gambar 4.8 Tampilan Alat Sebenarnya



Gambar 4.9 Tampilan Alat Tampak Depan

BAB V

PENUTUP

Berdasarkan pengujian pada perangkat keras dan perangkat lunak yang dipergunakan dalam tugas akhir ini, maka dapat diambil kesimpulan dan saran-saran dari hasil yang diperoleh.

5.1. Kesimpulan

Setelah melakukan penelitian dan pengujian, penulis mengambil kesimpulan sebagai berikut:

1. Dengan jumlah air sebanyak 6 liter, mesin mampu memanaskan air dan menstabilkan suhu sesuai setpoint yang diinginkan menggunakan metode fuzzy dengan rata-rata *error* 1,29%.
2. Rata-rata waktu yang dibutuhkan untuk menakar gula yaitu 8,2 detik/4gram sedangkan rata-rata waktu yang dibutuhkan untuk menakar kopi yaitu 17,8 detik/4gram.

5.2. Saran

Sebagai pengembangan dari penelitian yang dilakukan, penulis memberikan saran sebagai berikut:

1. Penggantian sistem pemanas air agar daya yang digunakan tidak terlalu besar.
2. Perubahan mekanik penuang gula dan kopi agar lebih presisi.

DAFTAR PUSTAKA

- Anand. 2008. *Push button*, (Online),
(<http://www.ustudy.in/node/3041>, diakses 6 Mei 2013)
- Budiharto, Widodo. 2005. *Perancangan Sistem dan Aplikasi Mikrokontroler*. Jakarta: PT Elex Media Komputindo.
- Budiharto, Widodo. 2008. *10 Proyek Robot Spektakuler*. Jakarta: PT Elex Media Komputindo.
- EngineersGarage. 2012. *LM35 Temperature Sensor*, (Online),
(<http://www.engineersgarage.com/electronic-components/lm35-sensor-datasheet>, diakses 6 Mei 2013).
- Hewes, J. 2013. , *Relays*, (Online),
(<http://electronicsclub.info/relay.htm>, diakses 6 Mei 2013)
- Iswanto. 2008. *Desain dan Implementasi Sistem Embedded Mikrokontroler ATmega8535 Dengan Bahasa Basic*. Jogjakarta: Gava Media.
- Lelyana, R. 2008. Pengaruh Kopi Terhadap Kadar Asam Urat Darah. Tesis tidak diterbitkan. Semarang: Program Pascasarjana Magister Ilmu Biomedik Universitas Diponegoro.
- Malik, M. I. & Juwana, M. Unggul. 2009. *Aneka Proyek Mikrokontroler PIC16F84/A*. Jakarta: PT. Elex media Komputindo.
- Rahim, Saifur. 2010. Alat Pembuat Minuman Kopi Otomatis Berbasis Mikrokontroler AT89C52. Skripsi tidak diterbitkan. Malang: Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim.
- Spikenzielabs. 2011. *A quick introduction to LCD display*, (Online),
(http://spikenzielabs.com/SpikenzieLabs/LCD_How_To.html, diakses 6 Mei 2013)
- Sri Kusumadewi, Hari Purnomo. 2010. *Aplikasi Logika Fuzzy Untuk Pendukung Keputusan, edisi 2*. Jogjakarta: Graha Ilmu.
- Suyadhi, T.D.S. 2010. *Buku Pintar Robotika Bagaimana Merancang dan Membuat Robot Sendiri*. Yogyakarta: ANDI Yogyakarta.
- Widodo. 2011. *Komponen Elektronika*, (Online),
(<http://profil.widodoonline.com/Elektronika/komponen/komponen-pasif/saklar.html>, diakses 3 April 2013).
- Dimas.2013. *Cara Membuat Kopi*, (Online),
(<http://madrecoffe.com/cara-membuat-kopi/>, diakses 2 Juni 2015).