



**UNIVERSITAS INDONESIA**

**PENGENDALIAN KECEPATAN PUTARAN MOTOR DC  
TERHADAP PERUBAHAN TEMPERATUR DENGAN  
SISTEM MODULASI LEBAR PULSA**

**SKRIPSI**

**MOHAMMAD HAMDANI  
0806366094**

**DEPARTEMEN ELEKTRO  
FAKULTAS TEKNIK UNIVERSITAS INDONESIA  
DEPOK  
JULI 2010**



**UNIVERSITAS INDONESIA**

**PENGENDALIAN KECEPATAN PUTARAN MOTOR DC  
TERHADAP PERUBAHAN TEMPERATUR DENGAN  
SISTEM MODULASI LEBAR PULSA**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik**

**MOHAMMAD HAMDANI  
0806366094**

**DEPARTEMEN ELEKTRO  
FAKULTAS TEKNIK UNIVERSITAS INDONESIA  
DEPOK  
JULI 2010**

## **PERNYATAAN KEASLIAN SKRIPSI**

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul :

### **PENGENDALIAN KECEPATAN PUTARAN MOTOR DC**

### **TERHADAP PERUBAHAN TEMPERATUR DENGAN**

### **SISTEM MODULASI LEBAR PULSA**

Yang dibuat untuk melengkapi persyaratan kelulusan pendidikan jenjang strata 1,

Departemen Elektro, Fakultas Teknik, Universitas Indonesia, sejauh yang saya

ketahui bukan merupakan tiruan atau duplikasi dari tugas seminar / skripsi yang

pernah diajukan untuk mendapatkan gelar kesarjanaan, kecuali bagian-bagian

yang sumber informasinya dicantumkan sebagaimana mestinya.

**Nama** : **Mohammad Hamdani**

**NPM** : **0806366094**

**Tanda Tangan** :

**Tanggal** :

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Mohammad Hamdani  
NPM : 0806366094  
Program Studi : Teknik Elektro  
Judul Skripsi : Pengendalian Kecepatan Putaran Motor DC Terhadap  
Perubahan Temperatur Dengan Sistem Modulasi Lebar  
Pulsa

**Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro Fakultas Teknik Universitas Indonesia.**

### DEWAN PENGUJI

Pembimbing : Prof. Dr. Ir. Rudy Setiabudy DEA ( )  
Penguji : Prof. Dr. Ir. Iwa Garniwa M K MT ( )  
Penguji : Budi Sudiarto ST, MT ( )

Ditetapkan di : Depok  
Tanggal : 7 Juli 2010

## KATA PENGANTAR

Segala Puji dan Syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan Rahmat dan Hidayah-Nya, sehingga penulis dapat menyelesaikan Skripsi ini dengan baik. Shalawat serta Salam semoga selalu tercurah kepada Nabi Besar Muhammad SAW, beserta keluarga dan sahabatnya. Penulis menyadari tidak mudah untuk menyelesaikan penulisan skripsi ini tanpa bantuan dan dukungan dari :

1. Prof. Rudi Setiabudy selaku Dosen Pembimbing yang telah banyak meluangkan waktu untuk memberikan saran, bimbingan dan pengarahan dalam penyelesaian Skripsi ini.
2. Kedua orang tua, adik dan keluarga yang telah banyak memberikan dukungan moral dan materil kepada penulis dalam penyelesaian Skripsi ini.
3. Rekan-rekan mahasiswa dan rekan-rekan kerja yang telah banyak memberikan dukungan moril kepada penulis.

Penulis berharap semoga Skripsi ini dapat bermanfaat bagi para pembaca, khususnya mahasiswa Teknik Elektro Universitas Indonesia dan para pembaca pada umumnya.

## ABSTRAK

Nama : Mohammad Hamdani  
Program Studi : Teknik Elektro  
Judul : Pengendalian Kecepatan Putaran Motor DC  
Terhadap Perubahan Temperatur Dengan  
Sistem Modulasi Lebar Pulsa

Kemajuan teknologi telah banyak diterapkan dalam kehidupan sehari-hari dan dunia kerja baik dalam sektor real maupun non-real. Penerapan teknologi ini diharapkan dapat meningkatkan efektifitas dan efisiensi dari suatu proses kerja dan kegiatan serta dapat meningkatkan hasil yang diharapkan baik dari segi kualitas maupun kuantitas. Penerapan teknologi ini dapat dibedakan menjadi dua, yaitu teknologi dengan sistem manual dan teknologi dengan sistem otomatis. Dari kedua sistem teknologi tersebut sistem teknologi otomatis yang paling banyak digunakan dalam perkembangannya karena hasil yang diperoleh dapat menjadi sangat signifikan. Sebagai contoh adalah perlunya suatu kondisi temperatur yang stabil seperti pada rumah kaca tempat perkembangbiakan dan perawatan tanaman dimana untuk mencapai temperatur yang stabil tersebut dilakukan dengan sistem yang dapat bekerja secara otomatis. Mengacu pada hal tersebut, akan dirancang sebuah aplikasi dalam bentuk prototype yang diharapkan dapat menjaga kondisi temperatur dalam kondisi yang cukup stabil, yaitu perancangan pengendalian kecepatan putaran Motor DC terhadap perubahan temperatur. Tujuan dari perancangan alat ini adalah untuk mengendalikan atau mengatur kecepatan putaran motor DC, dimana perubahan kecepatan putaran terjadi dengan adanya perubahan temperatur. Untuk pengendalian kecepatan Motor DC ini digunakan metode Pulse Width Modulation (PWM) untuk mengendalikan kecepatan putarannya, yaitu dengan mengatur durasi waktu tunda dari pulsa yang diumpankan kepada rangkaian pengendali (driver) motor DC 12 Volt yang juga berfungsi sebagai penguat sinyal PWM. Pengaturan durasi waktu tunda tersebut dapat dilakukan pada sisi negatif atau sisi positif dari pulsa dimaksud. Semakin lebar waktu tunda positif yang dihasilkan maka putaran motor akan semakin cepat dan begitu juga sebaliknya. Komponen utama dari alat ini adalah Motor DC 12 Volt, Mikrokontroler AT89S51, LCD, sensor temperatur, sensor putaran dan beberapa komponen pendukung lainnya. Perubahan temperatur lingkungan sekitar akan dibaca oleh sensor temperatur dan diproses menjadi sinyal output berupa tegangan listrik. Perubahan durasi waktu tunda dari pulsa atau sinyal Pulse Width Modulation (PWM), dipicu dengan adanya perubahan sinyal output yang dihasilkan oleh sensor temperatur tersebut. Sinyal output sensor temperatur terlebih dahulu diumpankan ke rangkaian mikrokontroler untuk dapat dihasilkan sinyal Pulse Width Modulation (PWM) yang sesuai dengan perencanaan sistem. Sensor putaran akan mendeteksi kecepatan putaran dari motor untuk ditampilkan pada LCD berikut perubahan nilai dari temperatur.

Kata Kunci :  
pmw, motor dc, temperature, kecepatan, mikrokontroler

## ABSTRACT

Name : Mohammad Hamdani  
Study Program : Electrical Engineering  
Title : DC Motor Speed Control dial on Change of Temperature  
With Pulse Width Modulation System

Advances in technology has been widely applied in daily life and the world of work both in the real sector and non-real. Application of this technology is expected to improve the effectiveness and efficiency of work processes and activities and can improve the expected outcome in terms of both quality and quantity. Application of this technology can be divided into two, namely technology and technology with manual systems with automated systems. From both of these technology systems automated technology systems most widely used in its development because the results obtained can be very significant. An example is the need for a stable temperature conditions as in the greenhouse where the breeding and care of plants in which to achieve a stable temperature was performed with a system that can work automatically. Referring to this, we will design an application in the form of a prototype that is expected to maintain the temperature within a fairly stable condition, namely the design of DC motor speed control to change of temperature with pulse width modulation system. The purpose of this tool design is to control or adjust the rotation speed of DC motor, where the rotation speed changes occur with the change of environmental temperature. DC motor speed control method is used Pulse Width Modulation (PWM) to control the speed of rotation, namely by adjusting the duration of the pulse delay circuit is fed to the controller (driver) 12-volt DC motor that also functions as a PWM signal amplifier. Setting the duration of the delay time can be done on the negative or positive side of the pulse in question. The more broadly positive delay time generated then the motor will spin faster and vice versa. The main components of this tool is 12 Volt DC Motor, AT89S51 Microcontroller, LCD, temperature sensor, rotation sensor and a few other supporting components. Changes in temperature of the surrounding environment will be read by a temperature sensor and processed output signal into an electrical voltage. Change the duration of the time delay of the signal pulse or pulse width modulation (PWM), triggered by a change in the output signal generated by the temperature sensor. Temperature sensor output signal is fed to the first circuit can be generated signal microcontroller for Pulse Width Modulation (PWM) that corresponds to the planning system. Rotation sensor detects rotation speed of the motor to be displayed on the LCD below the value of the temperature changes.

Keywords :  
pmw, dc motor, temperature, speed, microcontroller

## DAFTAR ISI

<b>PERNYATAAN KEASLIAN SKRIPSI .....</b>	<b>iii</b>
<b>HALAMAN PENGESAHAN .....</b>	<b>iv</b>
<b>KATA PENGANTAR .....</b>	<b>v</b>
<b>ABSTRAK.....</b>	<b>vi</b>
<b>DAFTAR ISI .....</b>	<b>viii</b>
<b>DAFTAR GAMBAR .....</b>	<b>x</b>
<b>DAFTAR TABEL .....</b>	<b>xii</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xiii</b>
<b>1. PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Tujuan.....	2
1.3 Batasan Masalah.....	3
1.5 Sistematika Penulisan.....	4
<b>2. LANDASAN TEORI.....</b>	<b>5</b>
2.1 Sistem Kendali .....	5
2.1.1 Sistem Kendali Loop Terbuka.....	5
2.1.2 Sistem Kendali Loop Tertutup .....	5
2.2 Pulse Width Modulation.....	6
2.3 Analog to Digital Converter (ADC).....	8
2.4 Motor DC .....	9
2.4.1 Pengertian dan Jenis-jenis Motor DC.....	9
a. Motor DC Dengan Daya Terpisah .....	10
b. Motor DC Dengan Daya Sendiri.....	10
c. Motor DC Seri.....	12
d. Motor DC Kompon/Gabungan.....	13
2.4.2 Pengendali Kecepatan Putaran Motor DC.....	15
2.5 Definisi Temperatur .....	16
2.6 Sensor Temperatur .....	16
2.6.1 Jenis-jenis Sensor Temperatur .....	16
2.6.2 Sensor Temperatur LM35 .....	22
2.7 LCD .....	23
2.8 Mikrokontroler .....	25
2.8.1 Sekilas Tentang Mikrokontroler.....	25
2.8.2 Arsitektur Mikrokontroler MCS - 51 .....	26
2.8.3 Fasilitas Mikrokontroler AT89S51 .....	29
2.8.4 Bentuk Fisik dan Fungsi Pin AT89S51.....	29
2.8.4.1 Port 0 .....	30
2.8.4.2 Port 1 .....	31
2.8.4.3 Port 2.....	31
2.8.4.4 Port 3.....	31
2.8.4.5 PSEN (Program Store Enable).....	32
2.8.4.6 ALE (Address Latch Enable).....	32
2.8.4.7 EA (External Address).....	32
2.8.4.8 RST (Reset).....	33
2.8.4.9 On – Chip Oscilator (XTAL).....	33
2.8.4.10 Catu Daya.....	34
2.8.4.11 Organisasi Memori.....	34



2.8.4.12 Memori Program .....	34
2.8.4.13 Memori Data .....	35
2.8.5 Pemrograman MCS - 51 .....	37
2.8.5.1 Bahasa Pemrograman Assembly .....	37
2.8.5.2 Struktur Bahasa Assembly .....	38
2.8.6 Kompiler Bahasa Assembler .....	39
2.8.7 Pembuatan dan Pengisian Program ke MCS - 51 .....	40
2.8.8 Mode Pengalamatan .....	43
2.8.9 Instruksi-instruksi Dalam MCS - 51 .....	44
<b>3. PERANCANGAN .....</b>	<b>56</b>
3.1 Perancangan Hardware .....	56
3.1.1 Perancangan Fisik .....	56
3.1.2 Perancangan Elektrik .....	57
3.1.2.1 Rangkaian Minimum Sistem .....	57
3.1.2.2 Rangkaian Sensor Kecepatan Putaran Motor DC .....	58
3.1.2.3 Sensor Temperatur dan Rangkaian ADC .....	59
3.1.2.4 Rangkaian Pengendali Motor DC 12 Volt .....	60
3.1.2.5 Rangkaian Catu Daya .....	67
3.1.2.6 Perancangan Pulse Width Modulation (PWM) .....	68
3.1.2.7 Perancangan Pemanas Buatan .....	77
3.2 Perancangan Output Sensor Temperatur dan Input ADC .....	77
3.3 Perancangan Software .....	78
3.3.1 Perancangan Downloader Mikrokontroler AT89S51 .....	78
<b>4. HASIL PENGUKURAN DAN ANALISIS .....</b>	<b>81</b>
4. Pengukuran dan Analisis .....	81
4.1 Pengukuran dan Analisis ADC .....	81
4.2 Pengukuran dan Analisis Sensor Temperatur LM35 .....	83
4.3 Pengukuran dan Analisis Sinyal PWM .....	85
4.4 Pengukuran / Pengujian Alat .....	87
<b>5. KESIMPULAN .....</b>	<b>88</b>
<b>DAFTAR REFERENSI .....</b>	<b>90</b>
<b>LAMPIRAN</b>	

## DAFTAR GAMBAR

<b>2. LANDASAN TEORI</b> .....	<b>5</b>
Gambar 2.1 Fungsi Alih Sistem Kendali Loop Terbuka .....	5
Gambar 2.2 Fungsi Alih Sistem Kendali Loop Tertutup .....	6
Gambar 2.3 Tegangan Searah Setelah Dipenggal .....	7
Gambar 2.4 Bentuk Sinyal PWM .....	7
Gambar 2.5 Pinout IC ADC 0804.....	8
Gambar 2.6 Bentuk Fisik Motor DC.....	9
Gambar 2.7 Konstruksi Motor DC .....	9
Gambar 2.8 Rangkaian Motor DC Penguat Terpisah.....	10
Gambar 2.9 Rangkaian Motor DC Shunt .....	11
Gambar 2.10 Karakteristik Motor DC Shunt.....	11
Gambar 2.11 Rangkaian Motor DC Seri.....	12
Gambar 2.12 Karakteristik Motor DC Seri .....	13
Gambar 2.13 Rangkaian Motor DC Kompon .....	14
Gambar 2.14 Rangkaian Motor DC Kompon Pendek .....	14
Gambar 2.15 Karakteristik Motor DC Kompon .....	15
Gambar 2.16 Konfigurasi Sensor Thermocouple.....	17
Gambar 2.17 Bentuk Fisik Sensor RTD .....	18
Gambar 2.18 Kurva Karakteristik Sensor RTD .....	18
Gambar 2.19 Bentuk Fisik Sensor Thermistor .....	19
Gambar 2.20 Konfigurasi Sensor Pyrometer .....	20
Gambar 2.21 Bentuk Fisik Sensor Bimetallic .....	21
Gambar 2.22 Bentuk Fisik Sensor LM35.....	22
Gambar 2.23 Bentuk Gelombang Pengontrol dan Penggerak LCD .....	24
Gambar 2.24 Rangkaian Dasar LCD .....	24
Gambar 2.25 Struktur Sistem Mikrokontroler .....	25
Gambar 2.26 Arsitektik Mikrokontroler MCS - 51 .....	27
Gambar 2.27 Mikrokontroler AT 89S51 .....	30
Gambar 2.28 Peta Memori Program dan Memori Data MCS – 51 (AT 89C/S 51.....	34
Gambar 2.29 Konfigurasi MCS – 51 Menggunakan Memori Program Eksternal .....	35
Gambar 2.30 Memori Data Internal.....	36
Gambar 2.31 Peta Memori Data Internal MCS - 51 .....	36
Gambar 2.32 Proses Kompilasi Program Assembler .....	40
Gambar 2.33 Diagram Alir Pemrograman Mikrokontroler .....	42
<b>3. PERANCANGAN</b> .....	<b>56</b>
Gambar 3.1 Box Pelindung Rangkaian .....	56
Gambar 3.2 Prototype / Simulasi Ruangan .....	56
Gambar 3.3 Rangkaian Minimum Sistem .....	58
Gambar 3.4 Rangkaian Sensor Kecepatan Putaran Motor DC .....	58
Gambar 3.5 Layout Pin IC LM35 .....	59
Gambar 3.6 Rangkaian Analog To Digital Converter (ADC).....	60
Gambar 3.7 Bentuk Motor DC.....	61
Gambar 3.8 Rangkaian Pengendali Motor DC.....	62
Gambar 3.9 Diagram Blok Pengendalian Kecepatan Putaran Motor DC Terhadap Perubahan Temperatur Lingkungan Dengan Sistem Modulasi Lebar Pulsa .....	64
Gambar 3.10 Detail Diagram Blok Sistem .....	64

Gambar 3.11 Grafik Perubahan Kecepatan Putaran Motor DC Terhadap Perubahan Temperatur Lingkungan .....	66
Gambar 3.12 Rangkaian Catu Daya .....	67
Gambar 3.13 Diagram Alir Algoritma Perancangan Program .....	79
<b>4. HASIL PENGUKURAN DAN ANALISIS .....</b>	<b>81</b>
Gambar 4.1 Metode Pengukuran Sensor Temperatur LM35.....	84
Gambar 4.2 Kurva Sinyal PWM VS Kecepatan Putaran Motor DC .....	87



## DAFTAR TABEL

<b>2. LANDASAN TEORI</b> .....	<b>5</b>
Tabel 2.1 Kelebihan dan Kekurangan Thermocouple.....	17
Tabel 2.2 Kelebihan dan Kekurangan RTD.....	18
Tabel 2.3 Kelebihan dan Kekurangan Thermistor .....	19
Tabel 2.4 Kelebihan dan Kekurangan Pyrometer .....	20
Tabel 2.5 Kelebihan dan Kekurangan Bimetallic .....	21
Tabel 2.6 Kelebihan dan Kekurangan Filled System.....	21
Tabel 2.7 Fungsi Pin Out LCD.....	24
Tabel 2.8 Perbandingan Mikrokontroler Keluarga MCS - 51.....	28
Tabel 2.9 8051 Kompatibel (Atmel).....	28
Tabel 2.10 Fungsi Khusus Port 1.....	31
Tabel 2.11 Fungsi Khusus Port 3.....	31
Tabel 2.12 Nilai Register Setelah Reset .....	33
Tabel 2.13 Instruksi Aritmatika .....	45
Tabel 2.14 Instruksi Logika .....	48
<b>3. PERANCANGAN</b> .....	<b>56</b>
Tabel 3.1 Perhitungan nilai arus dan daya terhadap perubahan nilai tegangan input .....	63
Tabel 3.2 Perubahan kecepatan putaran Motor DC terhadap perubahan temperature lingkungan .....	65
Tabel 3.3 Temperatur rata-rata per bulan tahun 2009 di DKI Jakarta .....	66
Tabel 3.4 Hasil perhitungan tegangan, arus dan daya pada Motor DC .....	71
Tabel 3.5 Pembangkitan sinyal PWM .....	77
Tabel 3.6 Data gabungan in – out sensor temperature dan ADC .....	77
<b>4. HASIL PENGUKURAN DAN ANALISIS</b> .....	<b>81</b>
Tabel 4.1 Hasil Pengukuran Temperatur .....	81
Tabel 4.2 Hasil Pengukuran Vout Sensor Temperature Dengan Output Biner ADC ....	83
Tabel 4.3 Tegangan Input Motor DC Setelah Pemrosesan Sinyal PWM .....	83
Tabel 4.4 Perhitungan Kecepatan Putaran Motor DC.....	83
Tabel 4.5 Hasil Pengukuran Tegangan Output Sensor Temperature Dan Output Biner ADC.....	84
Tabel 4.6 Hasil Perhitungan Tegangan Output Sensor Temperature Dan Output Biner ADC.....	84

## DAFTAR LAMPIRAN

LAMPIRAN 1. Tabel Tegangan Input ADC

LAMPIRAN 2. Tabel Tegangan Output Sensor Temperatur LM35

LAMPIRAN 3. Listing Program



# **BAB 1 PENDAHULUAN**

## **1.1 Latar Belakang**

Dalam meningkatkan efektifitas dan efisiensi suatu tugas atau pekerjaan, dapat diterapkan sistem otomatisasi. Sebagai contoh suatu sistem conveyor berjalan pada suatu produksi ban mobil dan contoh lain dalam kehidupan sehari-hari yaitu pada proses pembuatan makanan dengan menggunakan suatu peralatan yang dapat diatur tingkat kepanasan atau waktu pemanasannya.

Dalam Skripsi ini akan dibuat suatu sistem otomatisasi untuk mengendalikan kecepatan putaran Motor DC yang dapat berubah dengan adanya perubahan temperature. Dari perancangan dan prototype ini diharapkan dapat dikembangkan dan diaplikasikan untuk keperluan lain seperti monitoring temperature suatu alat produksi, dan dimana pada lokasi tersebut tidak dimungkinkan seseorang berada didekatnya. Sehingga diperlukan pengendalian secara otomatis dan dapat dimonitor serta dikontrol dari lokasi yang berbeda. Memang hal tersebut memerlukan pengembangan dan modifikasi pada alat sesuai dengan peruntukan dan keperluannya.

Adapun untuk pengendalian kecepatan putaran motor itu sendiri, dapat dilakukan dengan beberapa teknik sebagai berikut :

1. Pengendalian kecepatan putaran motor DC dengan memberikan tegangan sumber yang bervariasi, diperlukan beberapa sumber tegangan. Untuk keperluan tersebut diperlukan juga komponen atau saklar pemilih yang dapat dioperasikan secara manual maupun otomatis serta diperlukan juga pengawatan yang berbeda-beda antara motor DC dengan masing-masing sumber tegangan yang besarnya berbeda. Dalam beberapa kondisi atau keperluan teknik ini tidaklah efektif dan efisien untuk diterapkan.
2. Pengendalian kecepatan putaran motor DC dengan resistor yang dirangkai secara seri dengan motor, tidak jauh berbeda dengan menggunakan sumber tegangan yang bervariasi. Perbedaan dari teknik ini adalah membatasi arus yang mengalir dari sumber tegangan ke motor dan permasalahan yang perlu diperhatikan adalah daya yang dimiliki resistor serta perubahan temperature

pada resistor. Hal ini juga menyebabkan teknik ini tidak efisien dan efektif bila digunakan pada beberapa kondisi atau keperluan.

3. Teknik lain yang dapat digunakan adalah dengan Pulse Width Modulation. Yaitu dengan mengatur lebar pulsa dari sinyal kontrol yang diumpankan ke rangkaian kendali (driver) motor DC. Teknik modulasi ini adalah dengan mengatur durasi atau lebar dari waktu tunda positif atau waktu tunda negatif pulsa-pulsa. Untuk membangkitkan sinyal PWM, digunakan komparator untuk membandingkan dua buah masukan yaitu generator sinyal dan sinyal referensi. Hasil keluaran dari komparator adalah sinyal PWM yang berupa pulsa-pulsa persegi yang berulang-ulang. Durasi atau lebar pulsa dapat dimodulasi dengan cara mengubah sinyal referensi. Metode PWM digunakan untuk mengatur kecepatan motor, informasi yang dibawa oleh pulsa-pulsa persegi merupakan tegangan rata-rata. Besarnya tegangan rata-rata tersebut dapat diperoleh dari :

$$V_{out} = (V_{ref} * duty\ cycle) / periode$$

Semakin lebar durasi waktu tunda positif dari sinyal PWM yang dihasilkan, maka putaran motor akan semakin cepat, demikian juga sebaliknya.

Selain dengan menggunakan komparator dalam pembangkitan sinyal PWM, dapat juga dengan menggunakan perangkat kontrol seperti Programmable Logic Controller atau dengan Mikrokontroller, dimana sinyal PWM dibentuk oleh program yang terdapat pada kontroler.

Dalam perancangan ini pengendalian kecepatan putaran motor dengan menggunakan metode Pulse Width Modulation (PWM) yang dibangkitkan oleh mikrokontroler AT89S51. Motor DC yang digunakan adalah motor DC dengan magnet permanen dengan tegangan operasi 12 Volt. Sedangkan untuk mengetahui perubahan temperatur dengan menggunakan Sensor Temperatur LM35. Untuk mendeteksi perubahan kecepatan putaran motor DC dengan menggunakan infra merah dan photo transistor yang dikemas dalam satu paket. Perubahan kecepatan putaran dan nilai temperatur akan ditampilkan pada LCD 16x2 baris.

## 1.2 Tujuan

Tujuan dari Skripsi adalah untuk membahas tentang pengendalian kecepatan

putaran motor DC, dimana kecepatan putaran motor DC dapat berubah dengan adanya perubahan temperature dengan sistem modulasi lebar pulsa. Diharapkan perancangan dan prototype ini dapat dikembangkan dan diterapkan pada aplikasi lain sehingga dapat meningkatkan efektifitas dan efisiensi. Karena dengan teknik Pulse Width Modulation (PWM) dalam pengendalian kecepatan putaran motor nilai arus dan tegangan yang diperlukan oleh motor dapat diubah besarnya sehingga daya listrik pun akan mengalami perubahan, hal ini tidak berbeda dengan teknik konvensional yang telah dijelaskan pada latar belakang dari Skripsi ini. Hanya saja dengan menggunakan teknik PWM, kita dapat memperoleh beberapa keuntungan. Sebagai contoh adalah pada CPU sebuah komputer, kita ketahui pemakaian komputer saat ini berkembang pesat dan dalam beberapa penggunaan user tidak memperhatikan waktu pengoperasiannya, sehingga dapat meningkatkan temperatur didalam CPU. Untuk mengatur atau mengendalikan temperatur tersebut dapat digunakan perancangan ini. Sehingga pada saat kondisi temperatur normal, kipas pendingin pada CPU berputar normal atau sesuai kecepatan yang ditentukan dan pada saat temperatur semakin tinggi, kipas pendingin akan berputar semakin cepat untuk membuang udara panas yang berada didalam CPU, sehingga dapat mengurangi penggunaan kipas pendingin pada CPU. Contoh lain yang dapat digunakan untuk mengembangkan aplikasi ini adalah bila dalam suatu ruangan memiliki kipas pendingin yang berfungsi untuk menstabilkan temperatur ruangan agar menjadi sejuk atau nyaman. Jadi bila dalam ruangan tersebut terjadi penambahan jumlah orang, maka sistem akan menyesuaikan kecepatan putaran motor dari kipas pendingin agar ruangan tetap terasa sejuk atau nyaman. Selain itu, tujuan dari Skripsi ini adalah untuk memenuhi persyaratan kelulusan pendidikan jenjang Strata 1 (S1) Teknik Elektro serta agar mahasiswa dapat lebih mengimplementasikan pengetahuan yang telah didapat selama masa perkuliahan.

### **1.3 Batasan Masalah**

Skripsi ini dibatasi pada sistem pengendalian kecepatan putaran motor DC, dimana perubahan kecepatan dipengaruhi dengan adanya perubahan temperature. Sistem ini bekerja secara otomatis, yaitu berbasis mikrokontroler AT89S51 dan beberapa sensor yang digunakan sebagai input atau sumber informasi bagi



sistem.

#### **1.4 Sistematika Penulisan**

Sistematika penulisan skripsi ini dibagi atas beberapa bab dan masing-masing bab terbagi menjadi beberapa sub-bab. Dibawah ini diuraikan secara singkat isi dari tiap-tiap bab untuk memberikan gambaran secara keseluruhan mengenai isi dari Skripsi ini. Adapun uraian dimaksud adalah sebagai berikut :

- Bab I Pendahuluan, menguraikan tentang latar belakang, tujuan, batasan masalah dan tentang sistematika penulisan.
- Bab II Landasan Teori, dalam bab ini menjelaskan tentang teori yang berhubungan dengan sistem atau alat yang dirancang, diantaranya teori tentang motor DC, Mikrokontroler, sensor temperatur, sensor putaran dan teori tentang LCD.
- Bab III Perancangan, pada bagian ini merupakan inti dari Skripsi ini, karena berisi pemaparan tentang perancangan sistem atau alat.
- Bab IV Hasil Pengukuran & Analisis, bagian ini menjelaskan hasil pengukuran yang dilakukan seperti pada input dan output alat serta analisis terhadap hasil perhitungan dan pengukuran tersebut.
- Bab VI Kesimpulan, pada bab ini menjelaskan kesimpulan atau inti sari yang dapat diambil dari pembuatan alat.

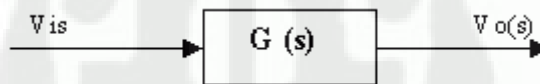
## BAB 2 LANDASAN TEORI

### 2.1 Sistem Kendali

Sistem kendali adalah suatu sistem yang bertujuan untuk mengendalikan suatu proses agar output yang dihasilkan dapat dikontrol sehingga tidak terjadi kesalahan. Dalam hal ini output yang dikendalikan adalah kestabilan, ketelitian dan kedinamisannya. Secara umum sistem kendali dapat dibagi menjadi 2 jenis, seperti dijelaskan dibawah ini.

#### 2.1.1 Sistem Kendali Loop Terbuka

Sistem kendali loop terbuka, keluarannya tidak mempengaruhi input. Atau dengan kata lain sistem kendali loop terbuka keluarannya (output) tidak dapat digunakan sebagai perbandingan umpan balik dengan inputnya. Akibatnya ketetapan dari sistem tergantung dari kalibrasi. Pada umumnya, sistem kendali loop terbuka tidak tahan terhadap gangguan luar. Dibawah ini adalah gambar diagram blok sistem kendali loop terbuka.



Gambar 2.1. Fungsi Alih Sistem Kendali Loop Terbuka

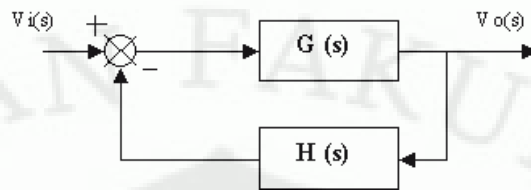
Fungsi alih sistem kendali loop terbuka adalah :  $V_{o(s)} = G_{(s)}.V_{i(s)}$

#### 2.1.2 Sistem Kendali Loop Tertutup

Sistem kendali loop tertutup seringkali disebut sistem kendali umpan balik. Pada sistem kendali loop tertutup, sinyal kesalahan yang bekerja, yaitu perbedaan antara sinyal input dan sinyal umpan balik diinputkan kekontroler sedemikian rupa untuk mengurangi kesalahan dan membawa keluaran sistem ke nilai yang dikehendaki. Pada umumnya sistem kendali loop tertutup tahan terhadap gangguan dari luar. Secara umum sistem kendali loop tertutup ini dibagi menjadi dua jenis, yaitu :

- Sistem kendali continue.
- Sistem kendali diskrit.

Secara umum gambar sistem kendali loop tertutup adalah sebagai berikut :



Gambar 2.2. Fungsi Alih Sistem Kendali Loop Tertutup

Fungsi alih sistem kendali loop tertutup adalah :

$$V_{O(s)} / V_{I(s)} = G(s) / (1+G(s) \cdot H(s))$$

Dimana :  $G(s)$  = Fungsi alih sistem

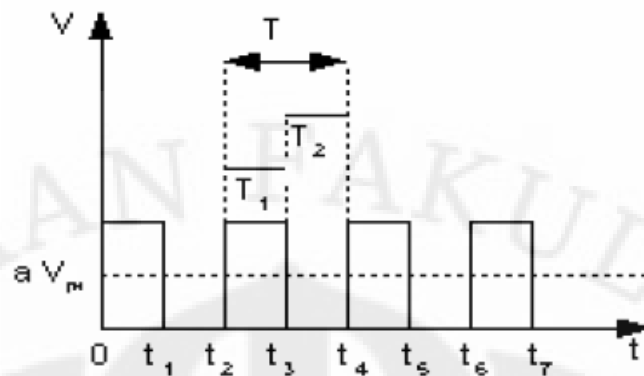
$H(s)$  = Fungsi alih transduser

Secara garis besar, sistem kendali jika ditinjau dari ketelitian dan kestabilan sistem dapat dibagi atas dua bagian, yaitu :

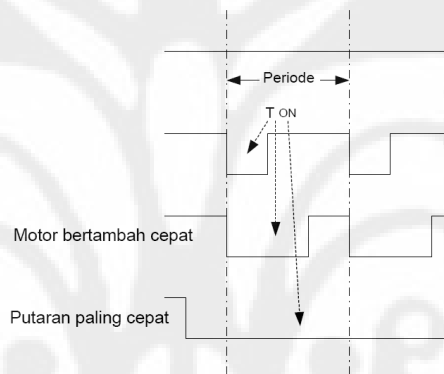
1. Sistem kendali dengan menggunakan PID Controller.
2. Sistem kendali on – off.

## 2.2 Pulse Width Modulation (PWM)

Pulse Width Modulation (PWM) adalah suatu jalan atau cara yang efisien dalam menyediakan sejumlah tegangan listrik antara kondisi "high" dan kondisi "low". Saklar tegangan sederhana dengan sumber tegangan yang tipikal hanya menyediakan tegangan penuh, ketika saklar dihidupkan. PWM adalah teknik pembanding terbaru, membuat efisien dengan menggunakan saklar tegangan elektronik modern, yaitu dengan mengubah tegangan masukan yang kita sebut dengan  $E_a$ , dengan mengingat fluks magnetnya tetap.



Gambar 2.3. Tegangan Searah Setelah Dipnggal



Gambar 2.4. Bentuk Sinyal PWM

Apabila tegangan searah dipnggal secara teratur seperti tercantum pada gambar 2.3 diatas, yaitu pada waktu  $t_1$  dimatikan,  $t_2$  dihidupkan kembali,  $t_3$  dimatikan dan seterusnya, maka ada tegangan searah yang secara teratur terputus-putus, dimana besarnya tergantung pada  $T_1$  dan  $T_2$ , yaitu :

$$V = a.V_m$$

Dimana :

$$a = \frac{T_1}{T_1 + T_2} = \frac{T_1}{T}$$

Untuk mengeset detak pencacah (counter) dan pewaktu (timer) untuk membangkitkan sinyal dan besarnya periode PWM. Sedangkan pada rangkaian penguat sinyal PWM, menggunakan transistor NPN BC517 yang dipasang secara Darlington agar mampu mengalirkan arus  $I_c$  sampai dengan 500 mA dengan

tegangan  $V_{CE}$  dan  $V_{CB}$  maksimal 60 Volt.

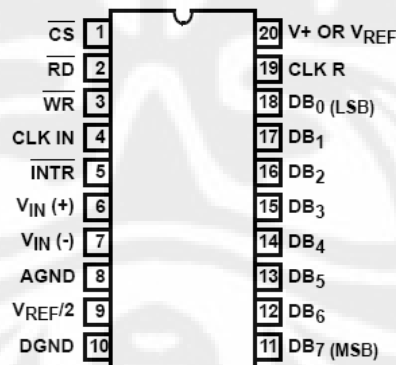
### 2.3 Analog To Digital Converter (ADC)

Pengubah analog ke digital mengambil masukan analog, mencupliknya, kemudian mengubah amplitudo dari setiap cuplikan menjadi sandi digital. Keluarannya adalah sejumlah bit-bit digital paralel yang status logikanya menunjukkan amplitudo dari setiap cuplikan.

Untuk mendapatkan hasil yang memuaskan, isyarat analog harus dicuplik dengan laju paling sedikit dua kali frekuensi tertinggi dari masukan analog asli. Laju pencuplikan ini disebut sebagai laju Nyquist.

Proses pengubahan sejumlah aras, misalnya 0.25, 0.5, 0.75, 1.0 dan seterusnya, disusun dengan sandi binernya. Langkah ini disebut dengan kuantisasi (quantising). Cacah aras kuantum ini ditentukan oleh cacah bit pada keluaran pengubah. Sebagai contoh, untuk ADC 3 bit, keluaran biner dapat bernilai 000 sampai dengan 111, yaitu sejumlah 8 aras. Dimisalkan digunakan skala atau kuantum sebesar 250 mV.

Pada perancangan ini digunakan ADC dalam bentuk integrated circuit dengan tipe 0804.



Gambar 2.5. Pin Out IC ADC0804

Dimana untuk dapat melakukan pembacaan sinyal input yang diberikan kepada ADC dan mengeluarkan sinyal output berupa deretan bilangan biner, perlu diberikan input logika yang tepat ("0" atau "1") pada pin CS, RD dan WR.

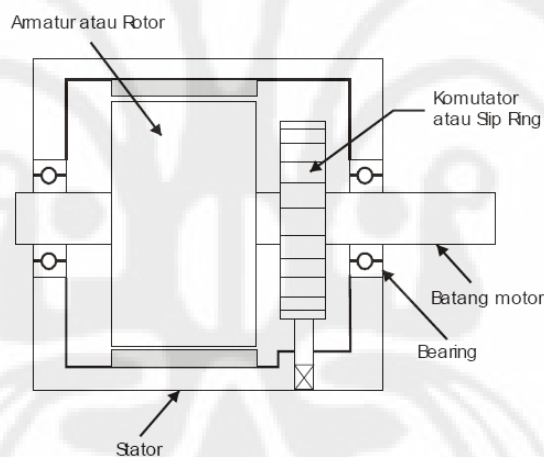
## 2.4 Motor DC

### 2.4.1 Pengertian dan Jenis-jenis Motor DC

Motor DC adalah sebuah aktuator yang mengubah besaran listrik menjadi sistem gerak mekanis. Motor DC beroperasi dengan prinsip-prinsip kemagnetan dasar. Polaritas arus yang mengalir melalui kawat lilitan akan menentukan arah putaran motor. Prinsip penting lainnya adalah nilai arus yang mengalir melalui lilitan. Nilai arus pada lilitan akan menentukan nilai torsi dan kecepatan putaran motor.



Gambar 2.6. Bentuk Fisik Motor DC



Gambar 2.7. Konstruksi Motor DC

- Kutub medan.** Secara sederhana digambarkan bahwa interaksi dua kutub magnet akan menyebabkan perputaran pada motor DC. Motor DC memiliki kutub medan yang stasioner dan dinamo yang menggerakkan *bearing* pada ruang diantara kutub medan. Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan.
- Rotor.** Bila arus masuk menuju rotor (bagian motor yang bergerak), maka arus ini akan menjadi elektromagnet. Rotor yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, rotor berputar dalam medan magnet yang dibentuk oleh

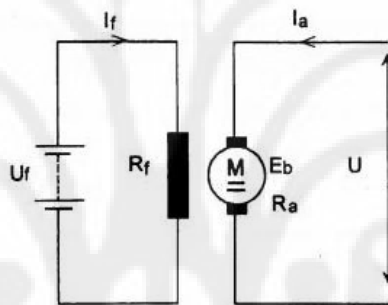
kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi. Jika hal ini terjadi, arusnya berbalik untuk merubah kutub-kutub utara dan selatan dinamo.

- c. **Komutator.** Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk membalikan arah arus listrik dalam dinamo. *Commutator* juga membantu dalam transmisi arus antara dinamo dan sumber daya.

Motor arus searah (DC) dapat dibedakan menjadi beberapa jenis sebagai berikut :

a. **Motor DC sumber daya terpisah/ Separately Excited**

Jika arus medan dipasok dari sumber terpisah maka disebut motor DC sumber daya terpisah/separately excited.



Gambar 2.8. Rangkaian Motor DC Penguat Terpisah

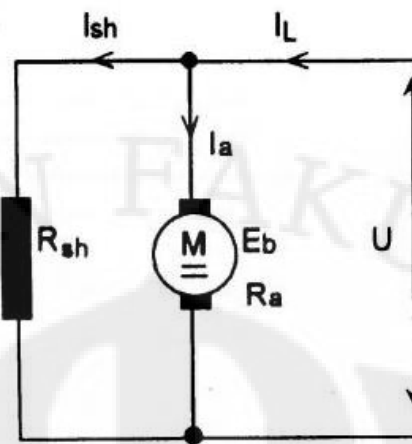
$$U = E_b + I_a \cdot R_a$$

$$I_f = \frac{U_f}{R_f}$$

$$U_f \neq U$$

b. **Motor DC sumber daya sendiri/ Self Excited: motor shunt**

Pada motor shunt, gulungan medan (medan shunt) disambungkan secara paralel dengan gulungan dinamo seperti diperlihatkan dalam gambar 2.9. Oleh karena itu total arus dalam jalur merupakan penjumlahan arus medan dan arus dinamo.



Gambar 2.9. Rangkaian Motor DC Shunt

$$U = E_b + I_a \cdot R_a$$

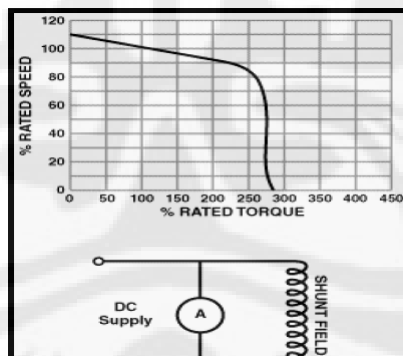
$$I_{sh} = \frac{U}{R_{sh}}$$

$$I_a = I_L - I_{sh}$$

$$T \approx \phi_0 \cdot I_a$$

$$T = f(I_a)$$

$$n = \frac{U - I_a \cdot R_a}{c}$$



Gambar 2.10. Karakteristik Motor DC Shunt (Rodwell International Corporation, 1999)

Berikut tentang kecepatan motor (E.T.E.,1997) :

- a. Kecepatan pada prakteknya konstan tidak tergantung pada beban (hingga torque tertentu setelah kecepatannya berkurang) dan oleh karena itu cocok untuk penggunaan komersial dengan beban awal yang rendah, seperti peralatan mesin.



b. Kecepatan dapat dikendalikan dengan cara memasang tahanan dalam susunan seri dengan dinamo (kecepatan berkurang) atau dengan memasang tahanan pada arus medan (kecepatan bertambah).

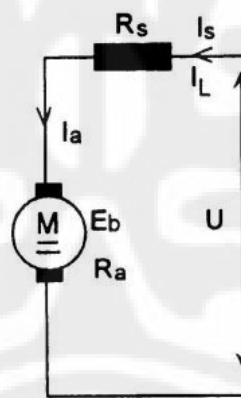
c. **Motor DC daya sendiri: motor seri**

Dalam motor seri, gulungan medan (medan shunt) dihubungkan secara seri dengan gulungan dinamo seperti ditunjukkan dalam gambar 2.11. Oleh karena itu, arus medan sama dengan arus dinamo. Berikut tentang kecepatan motor seri (Rodwell International Corporation, 1997; L.M. Photonics Ltd, 2002) :

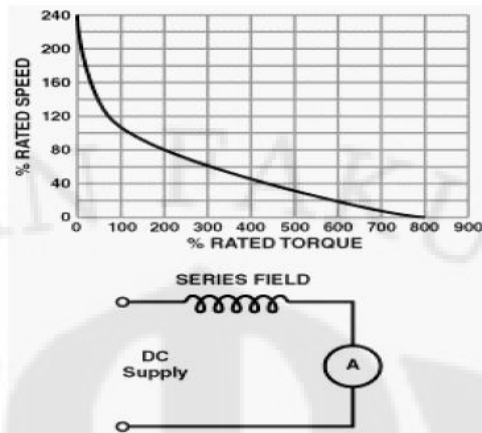
- Kecepatan dibatasi pada 5000 RPM.
- Harus dihindarkan menjalankan motor seri tanpa ada beban sebab motor akan mempercepat tanpa terkendali.

Dari penjelasan diatas dapat dikatakan motor seri mempunyai kecepatan putaran yang tidak konstan, jika beban tinggi maka putaran akan lambat.

Motor-motor seri cocok untuk penggunaan yang memerlukan *torque* penyalan awal yang tinggi, seperti derek dan alat pengangkat *hoist*.



Gambar 2.11. Rangkaian Motor DC Seri



Gambar 2.12. Karakteristik Motor DC Seri (Rodwell International Corporation, 1999)

$$U = E_b + I_a.R_a + I_s.R_s$$

$$I_L = I_a = I_s$$

$$U = E_b + I_a.(R_a + R_s)$$

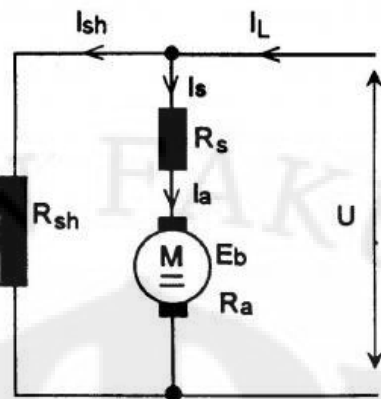
$$T \approx \phi_0.I_a$$

$$T = f(I_a)$$

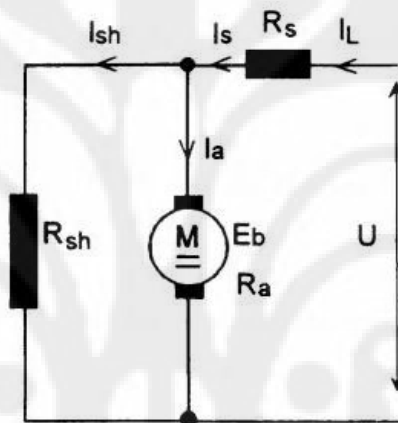
$$n = \frac{U - I_a.R_a}{c}$$

#### d. Motor DC Kompon/Gabungan

Motor Kompon DC merupakan gabungan motor seri dan *shunt*. Pada motor kompon, gulungan medan (medan *shunt*) dihubungkan secara paralel dan seri dengan gulungan dinamo seperti yang ditunjukkan dalam gambar 2.13. Sehingga, motor kompon memiliki *torque* penyalan awal yang bagus dan kecepatan yang stabil. Makin tinggi persentase penggabungan (yakni persentase gulungan medan yang dihubungkan secara seri), makin tinggi pula *torque* penyalan awal yang dapat ditangani oleh motor ini. Contoh penggabungan 40% – 50% menjadikan motor ini cocok untuk alat pengangkat hoist dan derek, sedangkan motor kompon yang standar (12%) tidak cocok (myElectrical, 2005).



Gambar 2.13. Rangkaian Motor DC Kompon

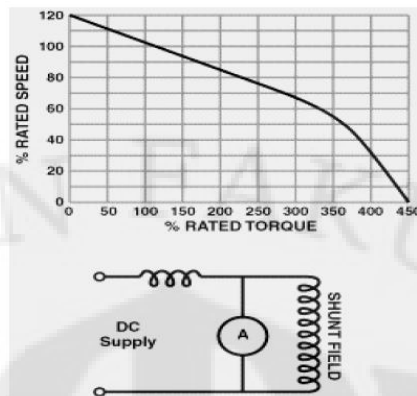


Gambar 2.14. Rangkaian Motor DC Kompon Pendek

$$U = E_b + I_a.R_a + I_s.R_s$$

$$I_L = I_s$$

$$I_L = I_a + I_{sh}$$



Gambar 2.15. Karakteristik Motor DC Kompon (Rodwell International Corporation, 1999)

$$U = Eb + Ia.Ra + Is.Rs$$

$$Ia = Is$$

$$I_L = Ia + Ish$$

$$Ish = \frac{U}{Rsh}$$

Dalam perancangan ini, dipilih motor yang memiliki daya yang tidak terlalu tinggi, yaitu menggunakan motor arus searah (DC) dengan penguat sendiri (motor seri) karena motor tersebut mempunyai magnet permanen pada statornya dan memperoleh sumber arus searah dari motor itu sendiri. Untuk membalik arah putaran motor arus searah (DC), dapat dilakukan dengan dua cara yaitu :

1. Membalik arah arus angkernya, sedangkan katub magnetnya tetap.
2. Membalik katup magnetnya, sedangkan arah arus angkernya tetap.

Dalam perancangan ini untuk membalik arah putaran motor dengan menggunakan cara pertama, yaitu membalik arah arus angkernya.

#### 2.4.2 Pengendali Kecepatan Putaran Motor DC

Seperti telah dijelaskan sebelumnya bahwa :

$$n = \frac{U - Ia.Ra}{c}$$

Maka pengendalian kecepatan putaran Motor DC ini dilakukan dengan cara mengubah tegangan masukan U.

## 2.5 Definisi Temperatur

Temperatur adalah ukuran rata-rata energi kinetik molekul-molekul dalam suatu objek atau sistem dan dapat diukur dengan thermometer atau kalorimeter. Ini adalah cara untuk menentukan energi internal yang terdapat dalam sistem.

## 2.6 Sensor Temperatur

Sebagai penjelasan awal, sensor adalah sesuatu yang digunakan untuk mendeteksi adanya perubahan lingkungan fisik atau kimia. Variabel keluaran dari sensor yang diubah menjadi besaran listrik disebut transducer.

Sensor temperatur digunakan untuk mendeteksi perubahan temperatur lingkungan untuk kemudian diubah menjadi besaran listrik dalam bentuk tegangan.

### 2.6.1 Jenis-jenis Sensor Temperatur

Sensor temperature dapat dibagi menjadi beberapa jenis berdasarkan beberapa hal sebagai berikut ;

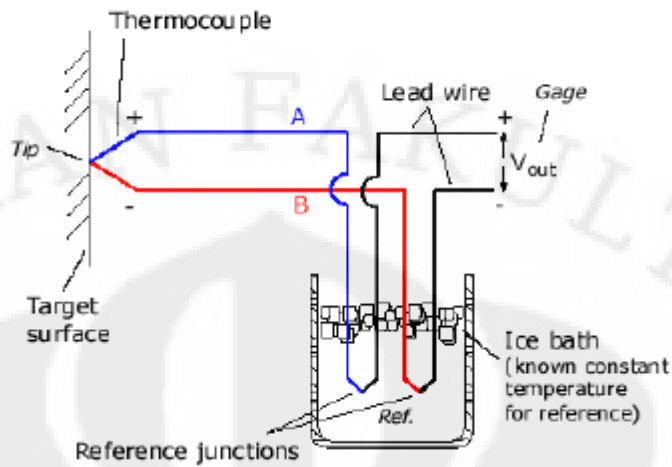
- a. Berdasarkan perubahan dimensi fisis
- b. Berdasarkan perubahan hambatan listrik
- c. Berdasarkan pembangkitan tegangan
- d. Berdasarkan perubahan emisi radiasi thermal

Adapun contoh-contoh sensor temperature berdasarkan perubahan diatas adalah sebagai berikut :

#### 1. Thermocouple

Merupakan sensor temperature yang terdiri dari dua kawat logam berbeda. Salah satu persimpangan merupakan ujung pengukuran dan persimpangan yang lain adalah ujung referensi (suhunya diketahui).

Perbedaan temperature antara ujung pengukuran dan ujung referensi dideteksi dengan pengukuran perubahan voltase (electromotive force, emf).



Gambar 2.16. Konfigurasi Sensor Thermocouple  
[http://adrian\\_nur.staff.uns.ac.id](http://adrian_nur.staff.uns.ac.id)

Tabel 2.1. Kelebihan dan Kekurangan Thermocouple

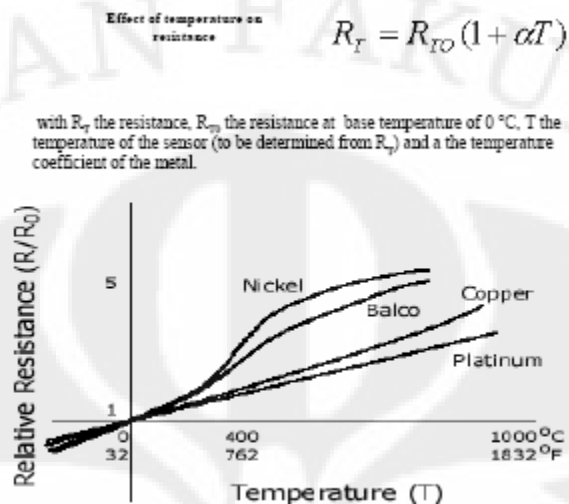
No.	Kelebihan	Kekurangan
1.	Biaya murah	Sensitivity rendah.
2.	Tidak ada pergerakan, sehingga tidak mudah rusak.	Membutuhkan temperature referensi seperti air es ( $0^{\circ}\text{C}$ ).
3.	Kisaran temperature luas.	Nonlinearity : diatasi dengan kurva kalibrasi dengan persamaan polynomial.
4.	Waktu respon cepat.	
5.	Repeatability dan akurasi cukup baik.	

## 2. Resistance Temperature Detectors (RTD)

RTD menggunakan kenaikan hambatan elektrik suatu logam dengan naiknya temperatur.



Gambar 2.17. Bentuk Fisik Sensor Resistance Temperature Detector ([http://adrian\\_nur.staff.uns.ac.id](http://adrian_nur.staff.uns.ac.id))



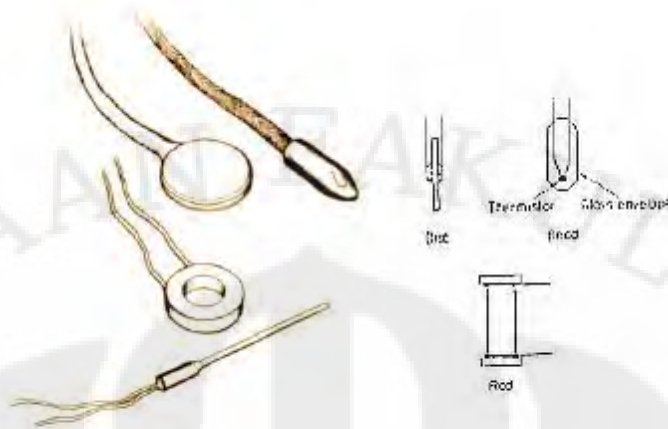
Gambar 2.18. Kurva Karakteristik Sensor Resistance Temperature Detector ([http://adrian\\_nur.staff.uns.ac.id](http://adrian_nur.staff.uns.ac.id))

Tabel 2.2. Kelebihan dan Kekurangan RTD

No.	Kelebihan	Kekurangan
1.	Stabil dan akurasi baik	Lebih mahal
2.	Linierisasi lebih baik daripada Thermocouple	Pemanasan sendiri
3.	-	Membutuhkan sumber arus
4.	-	Waktu respon kurang cepat pada beberapa aplikasi

### 3. Thermistor

Seperti RTD, thermistor menggunakan hambatan untuk mendeteksi temperature. Hanya saja pada thermistor menggunakan keramik semikonduktor yang mempunyai efek menurunkan hambatan pada naiknya temperatur.



Gambar 2.19. Bentuk Fisik Sensor Thermistor  
[http://adrian\\_nur.staff.uns.ac.id](http://adrian_nur.staff.uns.ac.id)

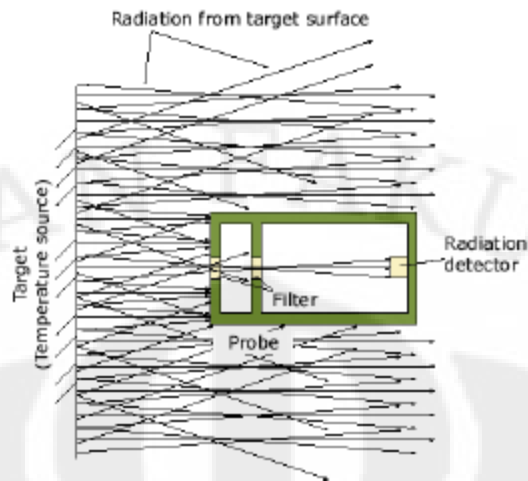
Tabel 2.3. Kelebihan dan Kekurangan Thermistor

No.	Kelebihan	Kekurangan
1.	Akurasi tinggi. Lebih tinggi dari pada RTD dan Thermocouple	Kisaran temperatur terbatas
2.	Sensitivitas tinggi	Linieritas rendah
3.	Ukuran lebih kecil daripada thermocouple	
4.	Waktu respon lebih baik daripada RTD, kurang lebih sama dengan Thermocouple	

#### 4. Pyrometer

Kadang disebut thermometer radiasi, tidak ada kontak langsung dengan bidang temperatur, pengukuran temperatur dari radiasi elektromagnetik (sinar tampak atau infra red) yang dilepaskan oleh objek.





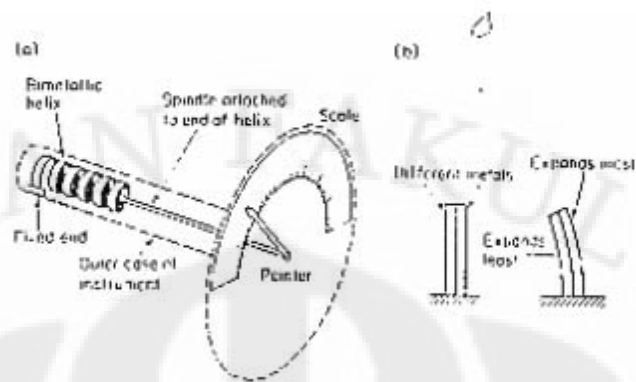
Gambar 2.20. Konfigurasi Sensor Pyrometer  
[http://adrian\\_nur.staff.uns.ac.id](http://adrian_nur.staff.uns.ac.id)

Tabel 2.4. Kelebihan dan Kekurangan Pyrometer

No.	Kelebihan	Kekurangan
1.	Pengukuran tanpa kontak	Mahal
2.	Waktu respon cepat	Akurasi terganggu oleh debu dan asap
3.	Stabilitas baik	

### 5. Bimetallic

Logam akan berekspansi, dengan naiknya temperatur dan kecepatan ekspansi berbeda antar logam satu dengan yang lain. Dua logam dikonstruksikan menjadi sebuah spiral. Perubahan posisi koil akan dideteksi dan digunakan untuk menentukan temperatur.



Gambar 2.21. Bentuk Fisik Sensor Bimetallic ([http://adrian\\_nur.staff.uns.ac.id](http://adrian_nur.staff.uns.ac.id))

Tabel 2.5. Kelebihan dan Kekurangan Bimetallic

No.	Kelebihan	Kekurangan
1.	Murah	Display secara local
2.	Secara fisik baik	-

**6. Filled Systems**

Fluida akan terekspansi dengan naiknya temperatur. Perubahan posisi dideteksi untuk menentukan temperatur.

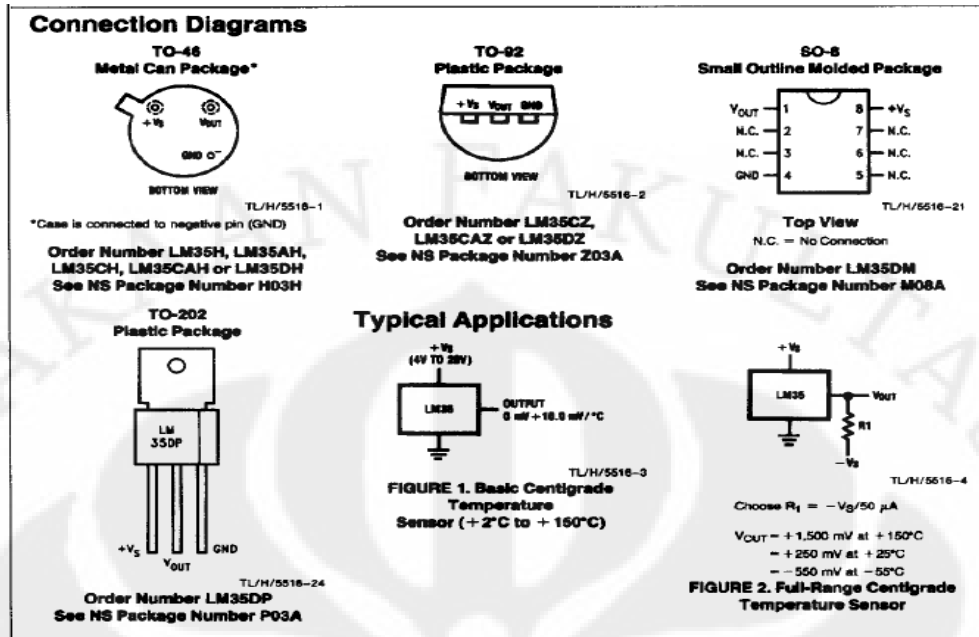
Tabel 2.6. Kelebihan dan Kekurangan Filled Systems

No.	Kelebihan	Kekurangan
1.	Sederhana dan biaya murah	Temperature tidak tinggi
2.	Tidak ada racun	Sensitive terhadap tekanan

**7. Integrated Circuit Temperature Sensor**

Integrated Circuit Temperature Sensor terdiri dari 2 (dua) jenis, yaitu :

1. Seri LM34 : dalam skala Fahrenheit
2. Seri LM35 : dalam skala Celcius



Gambar 2.22. Bentuk Fisik Sensor LM35  
 (<http://elisa.ugm.ac.id>)

## 2.6.2 Sensor Temperatur LM35

Pada perancangan ini digunakan sensor temperatur type LM35. Sensor LM35 ini memiliki keakuratan tinggi dan kemudahan perancangan jika dibandingkan dengan sensor suhu yang lain, LM35 juga mempunyai impedansi keluaran yang rendah dan linieritas 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 yang diberikan ke sensor adalah sebesar 5 Volt, sehingga dapat digunakan dengan catu daya tunggal dengan ketentuan bahwa LM35 hanya membutuhkan arus sebesar  $60 \mu A$ , hal ini berarti LM35 mempunyai kemampuan menghasilkan panas (self heating) dari sensor yang dapat menyebabkan kesalahan pembacaan yang rendah yaitu kurang dari  $0,5^\circ C$  pada suhu  $25^\circ C$ .

Tegangan keluaran atau  $V_{out}$  dengan jangkauan kerja dari 0 Volt sampai dengan 1,5 Volt dengan tegangan operasi yang dapat digunakan antara 4 Volt sampai 30 Volt. Keluaran sensor ini akan naik sebesar 10 mV setiap derajat Celcius, sehingga diperoleh persamaan sebagai berikut :

$$V_{LM35} = \text{suhu} * 10 \text{ mV}$$

Secara prinsip sensor akan melakukan penginderaan pada saat perubahan suhu

setiap  $1^{\circ}\text{C}$  akan menunjukkan tegangan sebesar 10 mV. Adapun besar tegangan output sensor seperti yang telah disebutkan diatas dapat dilihat sebagai berikut :

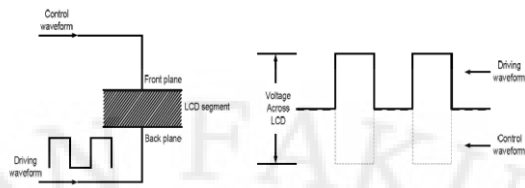
- $V_{out} = +1.500 \text{ mV}$  saat temperatur  $+150^{\circ}\text{C}$ .
- $V_{out} = +250 \text{ mV}$  saat temperatur  $+25^{\circ}\text{C}$ .
- $V_{out} = -550 \text{ mV}$  saat temperatur  $-55^{\circ}\text{C}$

Pada perancangan sistem ini, tegangan operasi yang diberikan kepada sensor temperatur LM35 adalah sebesar  $5 \text{ V}_{DC}$ , mengingat tegangan operasi yang dihasilkan oleh rangkaian catu daya adalah sebesar  $5 \text{ V}_{DC}$  dan  $12 \text{ V}_{DC}$ . Dimana tegangan  $12 \text{ V}_{DC}$  diperuntukan sebagai tegangan sumber bagi Motor DC.

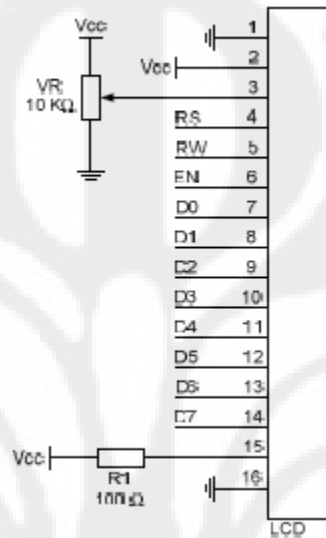
## 2.7 LCD

Persyaratan segment untuk menggerakkan antar muka untuk tampilan kristal cair berbeda dengan persyaratan untuk antar muka LED. Tampilan kristal cair menggunakan interaksi unik antara karakteristik elektrik dan optik dari suatu kelompok cairan yang tetap berada dalam bentuk kristal. Hal ini memberikan sifat optis yang sangat diperlukan sebagai peranti tampilan. Dengan pemakaian LCD, tidak ada cahaya yang dibangkitkan (berbeda halnya dengan LED), sehingga mengurangi konsumsi arus dan dayanya. Karena hal ini, LCD dapat digerakkan langsung oleh untai MOS dan CMOS.

Segment LCD mempunyai dua terminal atau bidang ; bidang depan dan bidang belakang. Arus bolak-balik dilewatkan pada kedua bidang dari LCD. Arus bolak-balik selalu digunakan untuk menghindari kemungkinan adanya pelapisan elektronik dan untuk memastikan umur maksimum dari segment LCD. Cara pengoperasian segment LCD antara lain dengan mengumpankan gelombang kotak frekuensi rendah (sekitar 40 Hz) pada bidang belakang. Untuk mematikan segment, gelombang kotak in-phase diumpankan ke bidang depan yang menghasilkan medan elektris nol sepanjang dua bidang dari segment tersebut. Sebaliknya, untuk menyalakan segment, gelombang kotak out-of-phase diumpankan ke bidang depan. Pada saat gelombang pengontrol dan gelombang penggerak pada keadaan out-of-phase, resultan gelombang pada segmen LCD dua kali amplitude gelombang kotak penggerak LCD akan menyebabkan segment menyala.



Gambar 2.23. Bentuk Gelombang Pengontrol dan Penggerak LCD



Gambar 2.24. Rangkaian Dasar LCD

Tabel 2.7. Fungsi Pin Out LCD

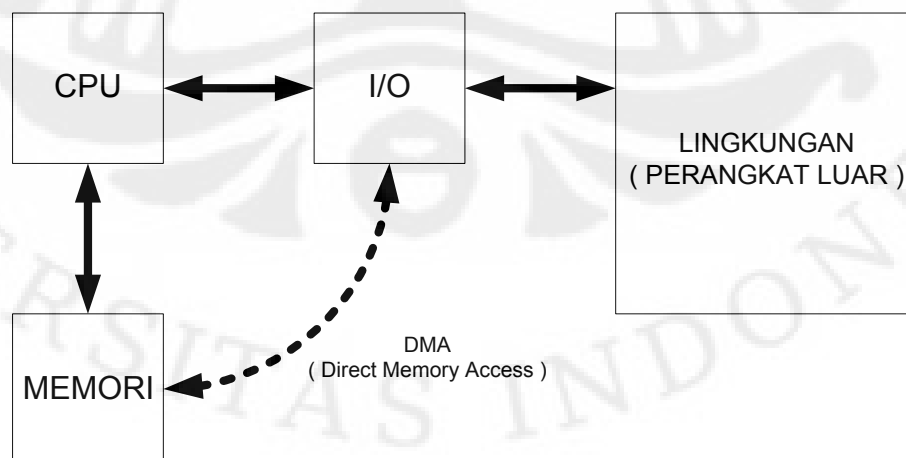
No.	Simbol	Level	Fungsi
1.	GND	-	Ground
2.	VCC	-	Supply +5V, ±5%
3.	VEE	-	Kontras
4.	RS	H/L	H : Data Masuk L : Data Keluar
5.	RW	H/L	H : Fungsi Read L : Fungsi Write
6.	EN	H	Sinyal Aktif LCD
7.	D0 – D7	H/L	Jalur Data
8.	BL1	-	Supply Cahaya Latar
9.	BL0	-	Supply Cahaya Latar

## 2.8 Mikrokontroler

### 2.8.1 Sekilas Tentang Mikrokontroler

Mikrokontroler merupakan mikrokomputer yang dikemas secara internal dalam sebuah IC / Chip atau biasa disebut single chip computer yang memiliki kemampuan untuk diprogram dan digunakan khusus untuk keperluan instrumentasi dan pengendalian. Jadi di dalam sebuah mikrokontroler, selain memiliki CPU ( Central Processing Unit ) juga terdapat memori dan perangkat I/O. Secara umum struktur sistem mikrokomputer adalah sebagai berikut :

1. *Mikroprosesor*, sebagai CPU yang berfungsi sebagai unit pengolah pusat seluruh sistem.
2. *Memori*, terdiri dari ROM ( *Read Only Memory* ) dan RAM ( *Random Access Memory* ). ROM berfungsi untuk menyimpan program / perangkat lunak yang akan dijalankan oleh CPU. Sedangkan RAM berfungsi sebagai tempat penyimpanan data sementara yang mungkin diperlukan oleh CPU sewaktu menjalankan perangkat lunak, misal digunakan untuk menyimpan nilai – nilai pada suatu variabel.
3. *Perangkat I/O*, berfungsi untuk menghubungkan sistem mikrokomputer dengan dunia luar.
4. *Clock*, merupakan perangkat tambahan yang terletak diluar sistem mikrokomputer dan berfungsi untuk menyinkronkan kerja semua perangkat dalam sistem. Sumber sinyal clock biasanya didapatkan dari oscilator kristal.

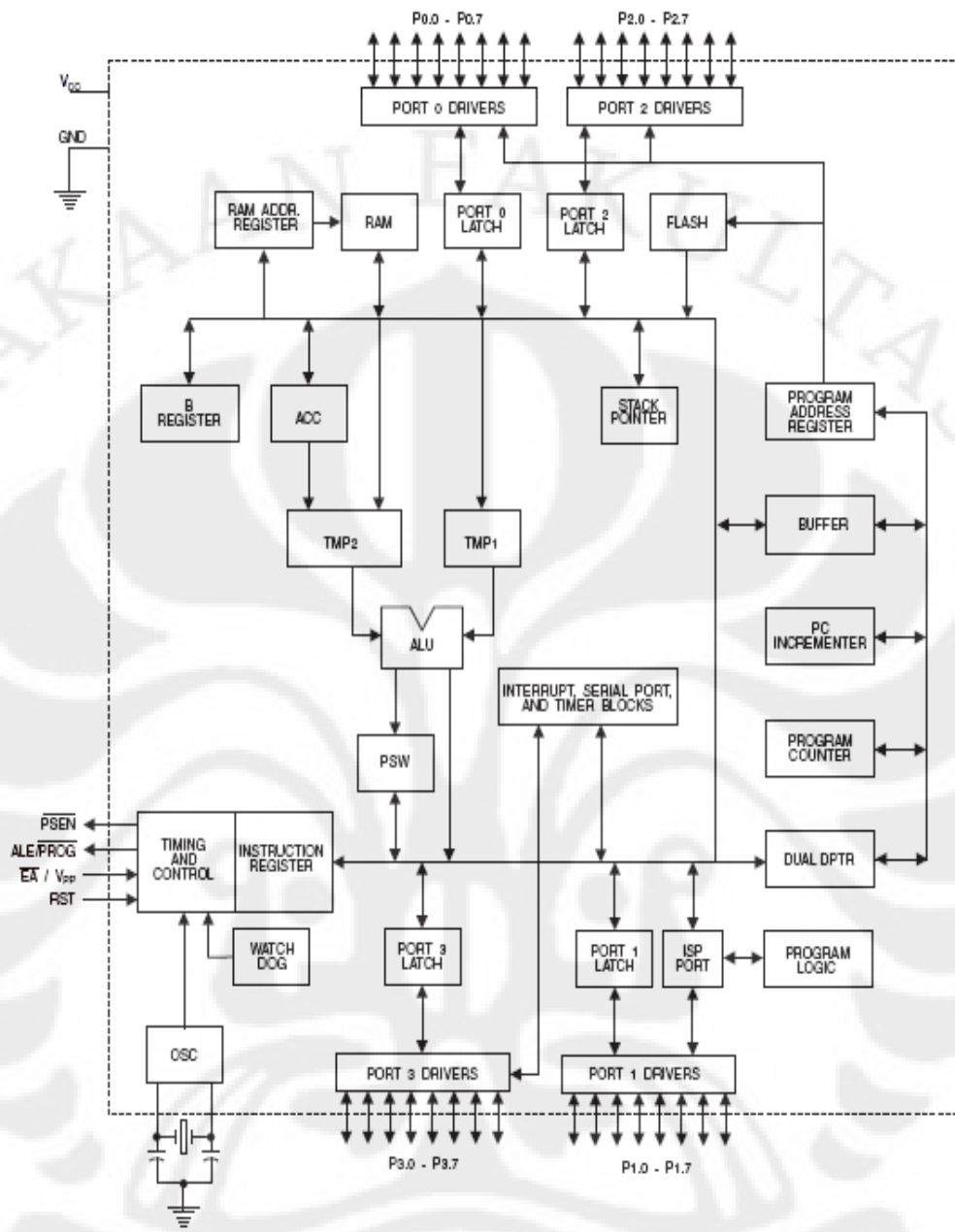


Gambar 2.25. Struktur Sistem Mikrokontroler

Adapun keunggulan dari mikrokontroler adalah adanya sistem interrupt. Sebagai perangkat kontrol penyesuaian, mikrokontroler sering disebut juga untuk menaikkan respon semangot eksternal (interrupt) di waktu yang nyata. Perangkat tersebut harus melakukan hubungan switching cepat, menunda satu proses ketika adanya respon eksekusi yang lain.

### **2.8.2 Arsitektur Mikrokontroler MCS – 51**

Mikrokontroler seri 8051 keluaran Intel merupakan generasi kedua mikrokontroler 8 bit yang diproduksi tahun 1980. Arsitektur mikrokontroler seri 8051 merupakan salah satu jenis arsitektur yang paling banyak digunakan, khususnya dikalangan industri, sehingga banyak diadopsi oleh berbagai vendor seperti Atmel, Philips, Siemens, dan beberapa vendor lain pada mikrokontroler yang diproduksinya. Mikrokontroler produksi Intel yang menggunakan arsitektur 8051 tergabung dalam satu keluarga mikrokontroler yaitu keluarga MCS – 51. Arsitektur mikrokontroler MCS – 51 dapat digambarkan sebagai berikut :



Gambar 2.26. Arsitektur Mikrokontroler MCS – 51

Mikrokontroler keluarga MCS – 51 merupakan jenis mikrokontroler yang sederhana, murah dan mudah didapatkan dipasaran. Keluarga MCS – 51 terdiri dari beberapa seri mikrokontroler seperti 8031, 80C31, 8051AH, dan 8751. Perbedaan antara seri yang satu dengan yang lain pada keluarga MCS – 51 seperti terlihat pada tabel 2.2 dibawah ini.



Tabel 2.8. Perbandingan Mikrokontroler Keluarga MCS – 51

<b>Seri Mikrokontroler</b>	<b>ROM Bytes</b>	<b>RAM Bytes</b>	<b>Timers</b>
8051	4KB ROM	128 bytes	2
8031	0	128 bytes	2
8751	4KB EPROM	128 bytes	2
8052	8KB ROM	256 bytes	3
8032	0	256 bytes	3
8752	8KB EPROM	256 bytes	3
8951	4KB EEPROM	128 bytes	2

Salah satu vendor lain yang menggunakan arsitektur MCS – 51 adalah Atmel pada beberapa mikrokontroler yang dibuatnya seperti seri AT89C51, AT89C52, AT89S52, dan beberapa seri yang lain. Tabel dibawah ini memperlihatkan beberapa seri mikrokontroler produksi Atmel yang kompatibel dengan mikrokontroler keluarga MCS – 51.

Tabel 2.9. 8051 Kompatibel ( Atmel )

<b>Seri Mikrokontroler</b>	<b>ROM Bytes</b>	<b>RAM Bytes</b>	<b>Timers</b>
AT89C1051	1K	64 bytes	1
AT89C2051	2K	128 bytes	2
AT89C/S51	4K	128 bytes	2
AT89C/S52	8K	256 bytes	3
AT89LV51	4K	128 bytes	2
AT89LV52	8K	256 bytes	3
AT89LV55L	20K	256 bytes	3

Dalam Skripsi ini, mikrokontroler yang digunakan adalah mikrokontroler produksi Atmel seri AT89S51. Mikrokontroler seri ini merupakan mikrokontroler yang mempunyai fasilitas ISP ( In System Programming ), artinya mikrokontroler dapat memprogram dirinya sendiri, sehingga dalam penggunaannya tidak perlu bongkar pasang untuk proses pengisian program karena dapat langsung diprogram pada rangkaian minimum sistem-nya.

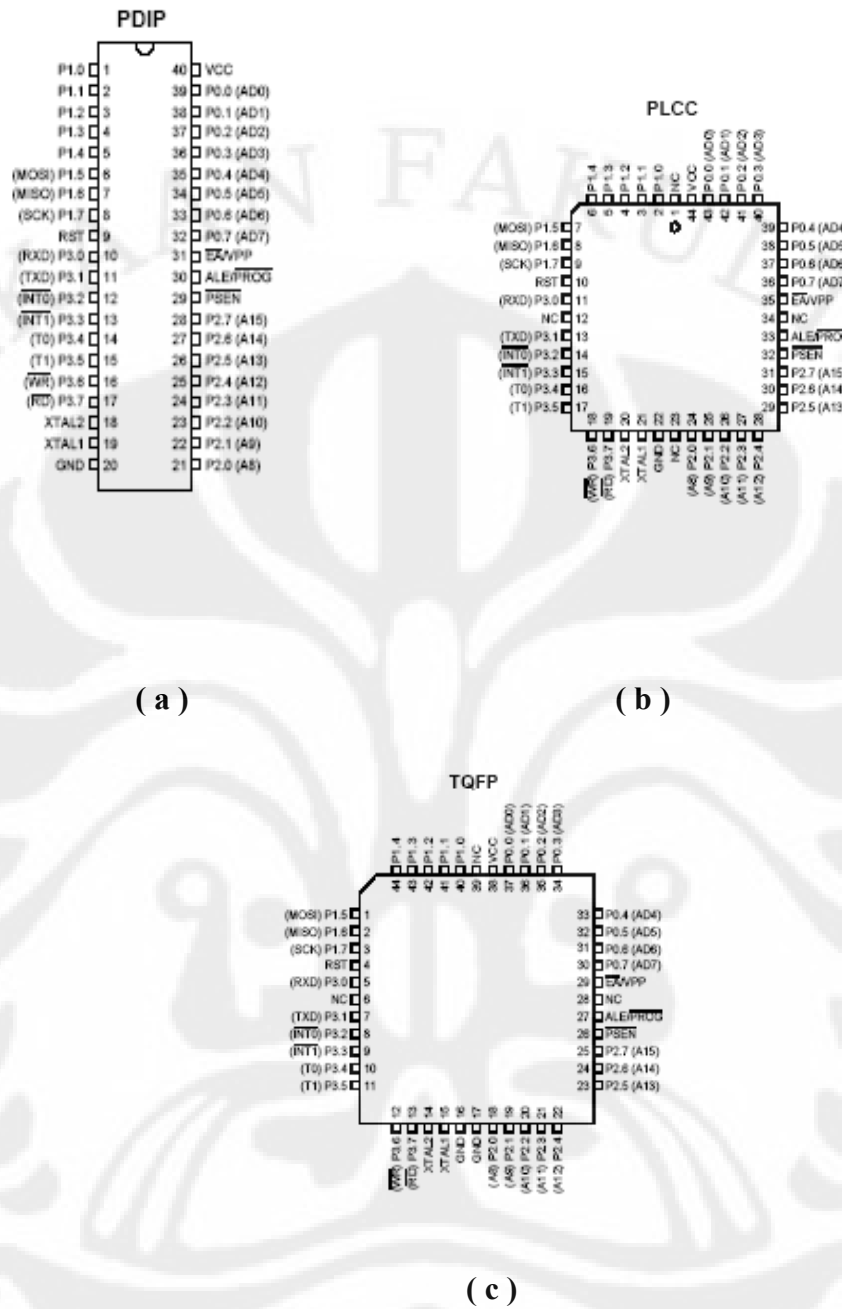
### 2.8.3 Fasilitas Mikrokontroler AT89S51

Fasilitas dan kemampuan ( Fitur ) dari mikrokontroler AT89S51 adalah sebagai berikut :

- Kompatibel dengan mikrokontroler keluarga MCS – 51 sebelumnya.
- 4 Kbyte ISP ( *In System Programming* ) flash PEROM yang digunakan untuk menyimpan program. Flash ini dapat dihapus dan diprogram ulang ( sampai 1000 kali ).
- 128 byte Internal Memori ( RAM ).
- 32 jalur I/O yang dikelompokkan dalam 4 buah port 8-bit ( port0, port1, port2, dan port3 ).
- 2 buah Timer / Counter 16 bit.
- 6 sumber Interrupt.
- Jalur komunikasi serial ( *full duplex serial UART* ).
- Frekuensi clock sampai 0 – 33 MHz.
- Low-Power Idle and Power-Down Modes
- Power-off flag
- Interrupt Recovery.
- Watchdog Timer.
- Dual Data Pointer.
- Mode pemrograman yang fleksibel ( Byte dan Page Mode ).
- 3 Level Program Memory Lock Bits, digunakan untuk mengunci program agar tidak dapat dibaca oleh orang lain.
- Tegangan operasi dinamis dari 4 – 5,5 volt.

### 2.8.4 Bentuk Fisik dan Fungsi Pin AT89S51

Bentuk fisik dan konfigurasi pin mikrokontroler seri AT89S51 adalah seperti terlihat pada gambar 2.26 di bawah ini.



Gambar 2.27. Mikrokontroler AT89S51 (a) PDIP 40-pin ; (b) PLCC 44-pin ; (c) TQFP 44-pin

Keterangan fungsi pada masing – masing Port dari AT89S51 adalah sebagai berikut :

#### 2.8.4.1 Port 0

Terdapat pada pin 32 – 39. Port 0 merupakan port yang memiliki dua fungsi ( dual – purpose port ). Untuk sistem yang sederhana ( mikrokontroler menggunakan memori internal ), port 0 berfungsi sebagai jalur I/O dua arah.

Sedangkan apabila mikrokontroler menggunakan memori eksternal, port 0 digunakan sebagai jalur data dan alamat yang di – multiplex. Port 0 dapat diakses secara byte ( P0 ) ataupun secara bit ( P0.X ).

#### 2.8.4.2 Port 1

Terdapat pada pin 1 – 8. Port 1 merupakan port yang berfungsi sebagai jalur I/O dua arah. Selain itu port 1 juga memiliki beberapa fungsi tambahan seperti terlihat pada tabel dibawah ini.

Tabel 2.10. Fungsi Khusus Port 1

Pin Port 1	Fungsi Khusus
P1.5	MOSI ( Jalur input instruksi pada saat pemrograman / ISP )
P1.6	MISO ( Jalur output data pada saat pemrograman / ISP )
P1.7	CLOCK IN ( Jalur masukan clock pada saat pemrograman / ISP )

#### 2.8.4.3 Port 2

Terdapat pada pin 21 – 28. Sama seperti port 0, port 2 juga memiliki dua fungsi ( Dual – Purpose Port ). Untuk sistem yang sederhana ( mikrokontroler menggunakan memori internal ), port 2 berfungsi sebagai jalur I/O dua arah. Sedangkan pada saat mikrokontroler menggunakan memori eksternal, port 2 digunakan sebagai penyalur alamat high byte.

#### 2.8.4.4 Port 3

Terdapat pada pin 11 – 18. Port 3 memiliki fungsi ganda, selain sebagai port masukan/keluaran data byte atau bit secara umum ( seperti P1.X ) port 3 juga mempunyai fungsi khusus seperti terlihat pada tabel di bawah ini.

Tabel 2.11. Fungsi Khusus Port 3

Pin Port 3	Fungsi Khusus
P3.0	RXD ( Port Input komunikasi serial )
P3.1	TXD ( Port Output komunikasi serial )
P3.2	INT0 ( Jalur Interupsi eksternal 0 ( aktif rendah ))
P3.3	INT1 ( Jalur Interupsi eksternal 1 ( aktif rendah ))

<b>P3.4</b>	T0 ( Input eksternal Timer 0 )
<b>P3.5</b>	T1 ( Input eksternal Timer 1 )
<b>P3.6</b>	WR ( Berfungsi sebagai sinyal kendali tulis, saat prosesor akan menulis data ke memori / IO luar )
<b>P3.7</b>	RD ( Berfungsi sebagai sinyal kendali baca, saat prosesor akan membaca data dari memori / IO luar )

#### **2.8.4.5 PSEN ( *Program Store Enable* )**

Terdapat pada pin 29. PSEN berfungsi sebagai sinyal pengendali yang mengizinkan mikrokontroler mengakses/membaca program dari memori eksternal. Pada saat mikrokontroler menggunakan memori eksternal, pin ini dihubungkan dengan pin OE ( Output Enable ) dari EEPROM. Sinyal PSEN akan low ( 0 ) pada tahap penjemputan ( fetch ) instruksi. PSEN akan selalu bernilai 0 pada pembacaan program memori internal.

#### **2.8.4.6 ALE ( *Address Latch Enable* )**

Terdapat pada pin 30. ALE berfungsi untuk menahan sementara alamat byte rendah pada proses pengalamatan ke memori eksternal. Ketika menggunakan program memori eksternal port 0 akan berfungsi sebagai address dan data bus. Pin ALE akan mengeluarkan sinyal high ( 1 ) pada saat keluaran dari port 0 adalah address, dan akan mengeluarkan sinyal low ( 0 ) pada saat keluaran dari port 0 adalah data. Maksud dari keberadaan ALE adalah untuk memberitahu piranti eksternal yang terhubung ke port 0 bahwa pada saat itu keluaran port 0 adalah address atau data.

#### **2.8.4.7 EA ( *External Address* )**

Terdapat pada pin 31. Pin ini digunakan untuk pilihan program, apakah akan menggunakan program internal atau program eksternal. Jika pin EA diberi masukan 1, maka mikrokontroler akan menjalankan program memori internal saja. Sedangkan jika diberi masukan 0, mikrokontroler akan menjalankan program memori eksternal. Pin PSEN otomatis tidak akan berfungsi jika EA bernilai High.

#### 2.8.4.8 RST (Reset)

Terdapat pada pin 9. Berfungsi sebagai pin Reset yang akan mereset mikrokontroler jika diberi masukan 1 selama minimal 2 siklus mesin, sehingga isi dari register – register internal AT89S51 akan bernilai default. Nilai default dari register - register internal AT89S51 setelah sistem reset adalah seperti terlihat pada tabel 2.12 di bawah ini.

Tabel 2.12. Nilai Register Setelah Reset

Register	Isi
Program Counter ( PC )	0000 0000b
Accumulator	0000 0000b
B Register	0000 0000b
PSW	0000 0000b
Stack Pointer ( SP )	0000 0111b
DPTR	0000 0000b
Port 0 – 3	1111 1111b
IP	XX00 0000b
IE	0X00 0000b
Timer Register	0000 0000b
SCON	0000 0000b
SBUF	XXXX XXXXb
PCON	0XXX 0000b

#### 2.8.4.9 On – Chip Oscilator ( XTAL )

AT89S51 memiliki on – chip oscilator yang dapat bekerja jika di – drive menggunakan kristal. Kristal yang digunakan untuk men – drive on – chip oscilator dihubungkan dengan pin 18 dan 19 dan ditambahkan kapasitor untuk menstabilkan sistem. Nilai kristal yang digunakan disesuaikan dengan frekuensi clock yang mampu dihasilkan. Untuk seri AT89S51 biasanya digunakan kristal dengan nilai 11,9 atau 12 MHz. Selain dapat di – drive dengan menggunakan kristal, on – chip oscilator juga dapat di – drive dengan menggunakan TTL oscilator.

#### 2.8.4.10 Catu Daya

Pin Vcc terdapat pada pin 40 dengan catu daya 5 volt, sedangkan pin ground ( Vss ) terdapat pada pin 20.

#### 2.8.4.11 Organisasi Memori

Berdasarkan sifat data yang disimpannya, memori dikelompokkan menjadi dua yaitu ROM ( Read Only Memory ) dan RAM ( Random Access Memory ). Perbedaan antara ROM dan RAM adalah sebagai berikut :

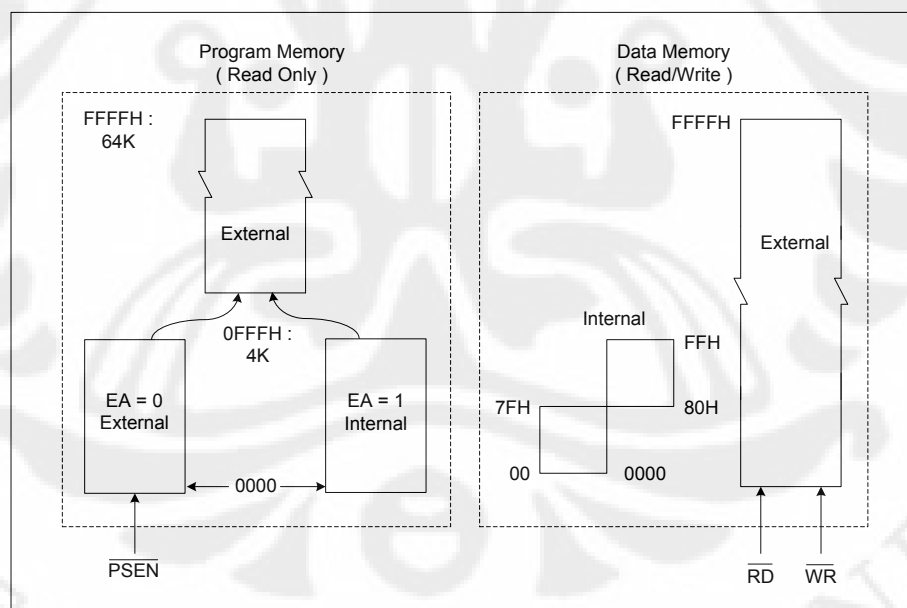
- **ROM ( Read Only Memory )**

ROM hanya bisa dibaca isinya tetapi tidak bisa ditulis dan bersifat non-volatile (isinya tidak hilang jika satu daya diputus).

- **RAM ( Random Access Memory )**

RAM bisa dibaca isinya maupun ditulis dan bersifat volatile (isinya tidak hilang jika catu daya diputus).

Peta memory dimaksud dapat dilihat pada gambar dibawah ini .

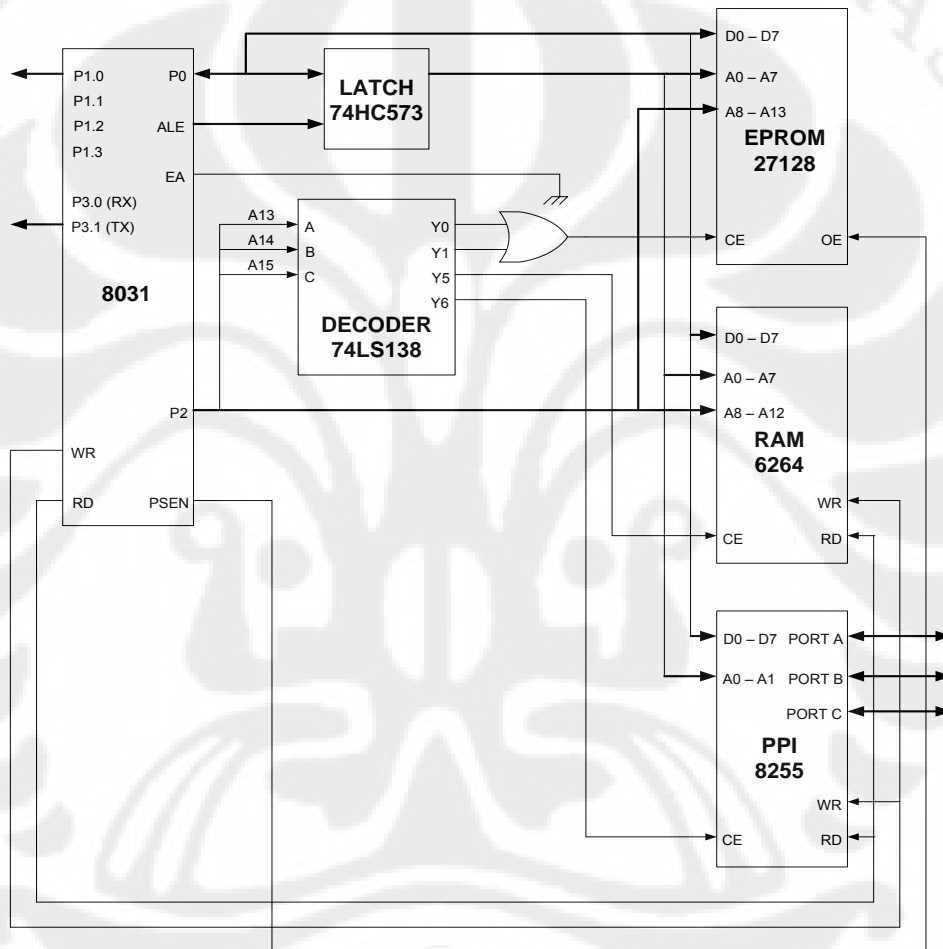


Gambar 2.28. Peta Memori Program dan Memori Data MCS – 51 ( AT89C /S51 )

#### 2.8.4.12 Memory Program

Memory program dalam mikrokontroler MCS-51 dapat berupa memori internal maupun memori eksternal. Pemilihan memori program ini tergantung dari kondisi

pin  $\overline{EA}$  (pin 31) dari mikrokontroler. Jika pin  $\overline{EA} = 1$  (dihubungkan ke Vcc), maka memori program yang digunakan adalah memori program internal/flash PEROM (Programmable Erasable ROM). Apabila memori program internal tidak cukup, maka dapat ditambah memori eksternal (EPROM, EEPROM, dll) dengan cara menghubungkan pin  $\overline{EA}$  dengan ground ( $\overline{EA} = 0$ ). Gambar di bawah ini memperlihatkan konfigurasi MCS-51 yang menggunakan memori eksternal.

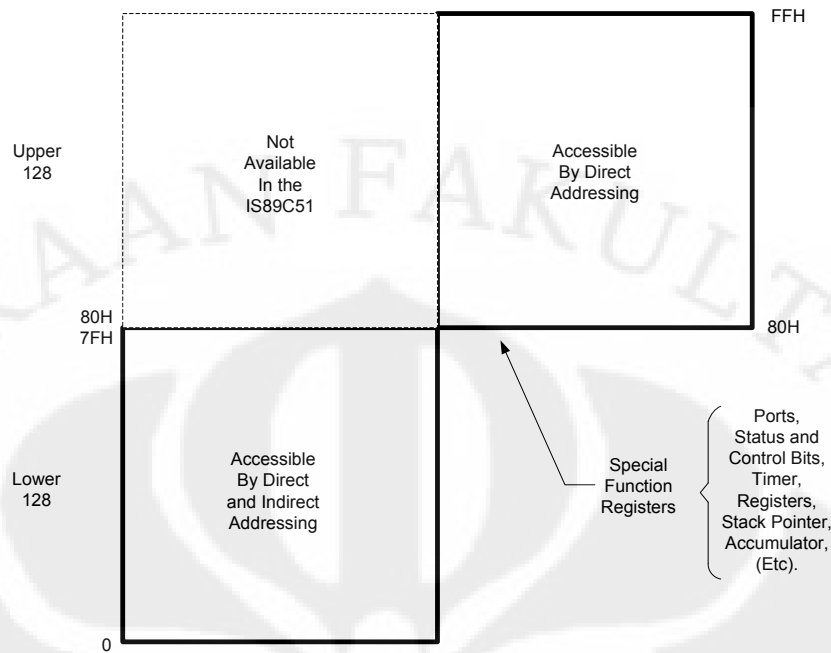


Gambar 2.29. Konfigurasi MCS – 51 Menggunakan Memori Program Eksternal

#### 2.8.4.13 Memori Data

Peta memori data dapat dikelompokkan menjadi dua, yaitu memori data internal dan memori data eksternal. Memori data internal dapat digambarkan seperti dibawah ini.





Gambar 2.30. Memori Data Internal

Peta lokasi alamat memori data internal seperti terlihat pada gambar dibawah ini.

No Memori	Keterangan								
\$00	R0	R1	R2	R3	R4	R5	R6	R7	Register – Kelompok 0
\$08	R0	R1	R2	R3	R4	R5	R6	R7	Register – Kelompok 1
\$10	R0	R1	R2	R3	R4	R5	R6	R7	Register – Kelompok 2
\$18	R0	R1	R2	R3	R4	R5	R6	R7	Register – Kelompok 3
\$20	Bit No \$00..\$07	Bit No \$08..\$0F	Bit No \$10..\$17	Bit No \$18..\$1F	Bit No \$20..\$27	Bit No \$28..\$2F	Bit No \$30..\$37	Bit No \$38..\$3F	Bit nomor \$00 sampai \$3F
\$28	Bit No \$40..\$47	Bit No \$48..\$4F	Bit No \$50..\$57	Bit No \$58..\$5F	Bit No \$60..\$67	Bit No \$68..\$6F	Bit No \$70..\$77	Bit No \$78..\$7F	Bit nomor \$40 sampai \$7F
\$30	Memori untuk keperluan umum sebanyak \$50 (80) byte dengan nomor memori \$30..\$7F								
\$7F	Special Function Register (SFR) dengan nomor memori \$80..\$FF								
\$80									
\$FF									

Tanda \$ merupakan awalan untuk menyatakan angka di belakangnya adalah bilangan heksa-desimal

Gambar 2.31. Peta Memori Data Internal MCS – 51

Memori data internal dapat dibagi menjadi 4 bagian, yaitu:

- a. Register Serba Guna (General Purpose Register).
- b. Memori Level Bit ( Bit-Addressable RAM).
- c. RAM fungsi umum.
- d. Register Fungsi Khusus ( SFR / Special Function Register ). Register ini terdiri atas :

- REGISTER A (*Accumulator*).
- REGISTER B.
- Register Penyimpan Status Program (*PSW / Program Status Word*).
- Register Pengontrol Prioritas Interupsi (*IPC / Interrupt Priority Control Bit*).
- Port Register.
- Pengatur Interupsi (*IE / Interrupt Enable*).
- Register Penampung Data Serial (*SBUF / Serial Buffer*).
- Register Pengatur Serial (*SCON / Serial Port Control*).
- Register Pewaktu (Register 16 bit).
- Register Pengatur Daya (*PCON / Power Control Register*).
- Register Penunjuk Data (*DPTR / Data Pointer*).
- Register Penunjuk Tumpukan (*SP / Stack Pointer*).
- Register Pencacah Program (*Program Counter*).

## **2.8.5 Pemrograman MCS-51**

### **2.8.5.1 Bahasa Pemrograman Assembly**

Bahasa pemrograman komputer dibagi menjadi dua tingkatan yaitu bahasa pemrograman tingkat tinggi ( high level language ) dan bahasa pemrograman tingkat rendah ( low level language ). Pada bahasa pemrograman tingkat tinggi instruksi – instruksi ( source code ) yang ditulis dalam program mudah dimengerti oleh manusia ( berorientasi kepada manusia ), contohnya adalah bahasa C, Pascal, dan Basic. Sedangkan pada bahasa pemrograman tingkat rendah, instruksi – instruksi yang ditulis harus dapat dimengerti dan diinterpretasikan oleh mesin ( berorientasi pada mesin ), contohnya adalah bahasa assembly.

Program yang ditulis dalam bahasa assembly relatif lebih panjang dan sulit dimengerti, namun mempunyai keunggulan dalam ukuran file yang kecil, kecepatan eksekusi, dan kemudahan dalam manipulasi sistem komputer. Penggunaan bahasa assembly memerlukan program assembler untuk mengkonversi instruksi – instruksi yang ditulis menjadi bahasa mesin. Bahasa assembly merupakan bahasa pemrograman yang paling banyak digunakan dalam pemrograman mikrokontroler. Selain bahasa assembly, bahasa C juga mulai banyak digunakan dalam pemrograman mikrokontroler turunan yang lebih tinggi seperti AVR dari Atmel dan Microchip PIC.

### 2.8.5.2 Struktur Bahasa Assembly

Struktur bahasa Assembly adalah susunan perintah yang terdiri dari : assembler directive, assembler control, label, mnemonic, dan komentar yang disusun membentuk suatu bahasa yang dimengerti oleh kompiler untuk diterjemahkan ke dalam bahasa mesin.

Secara umum format penulisan program dalam bahasa assembly adalah sebagai berikut :

Label <i>(isi memori)</i>	Mnemonic <i>(opcode)</i>	operand1	operand2	komentar
↓	↓	↓	↓	↓
Start_Program:	MOV	A,	#35H	;kopi 35H ke akumulator A

#### A. Label

Label merupakan simbol yang digunakan untuk menunjukkan alamat lokasi memori fisik yang berkaitan dengan pernyataan yang diberi label tersebut. Label harus diakhiri dengan tanda titik dua [ : ]. Aturan penulisan label dalam bahasa assembly adalah sebagai berikut :

- Penulisan huruf dan kecil tidak dibedakan.
- Penulisan label harus diawali dengan menggunakan huruf, tidak boleh diawali dengan angka.
- Tidak boleh menggunakan spasi. Untuk memisahkan dua karakter menggunakan garis bawah.

## **B. Mnemonic**

Mnemonic atau opcode adalah kode yang akan melakukan aksi terhadap operand. Mnemonic berkaitan dengan instruksi pada pemrograman bahasa assembly. Nama – nama mnemonic dan operand tidak boleh digunakan untuk simbol yang lain, misal untuk label.

## **C. Operand**

Operand adalah data atau register yang akan diproses oleh mnemonic / opcode. Sebuah opcode bisa membutuhkan 1, 2 atau lebih operand, tapi kadang juga tidak memerlukan operand.

## **D. Komentar**

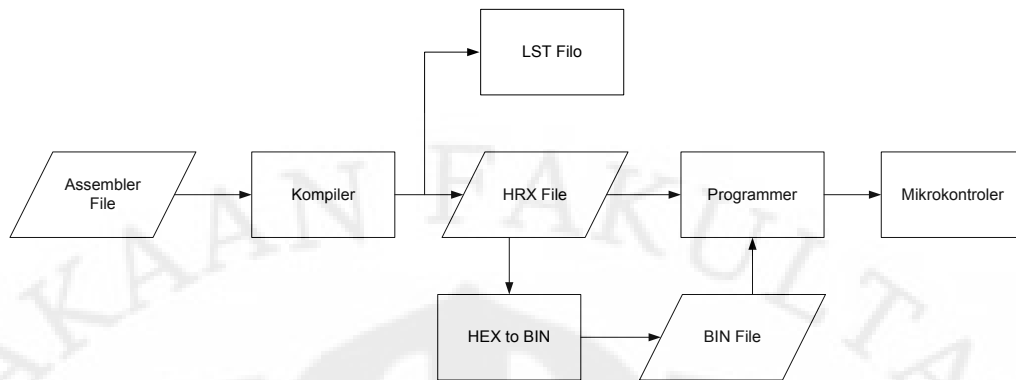
Komentar digunakan untuk memberi keterangan pada program assembler, dan bisa ditempatkan di baris mana saja dengan memberi tanda semicolon ( ; ) di awalnya.

### **2.8.6 Kompiler Bahasa Assembler**

Salah satu kompiler yang paling banyak digunakan adalah ASM51.EXE. Proses kompilasi harus bebas dari kesalahan. Hasil proses kompilasi terdiri dari dua buah file, yaitu file berekstensi .LST ( Listing File ) dan file berekstensi .HEX ( Intel Hex File ). Kedua file ini adalah file text yang dapat dibuka dengan text editor, misal Notepad.

File berekstensi .LST berisi informasi tentang alokasi memori, kode mesin, cross reference, adanya kesalahan program dan keterangan yang lain. Sedangkan file berekstensi .HEX berisi kode – kode bahasa mesin yang akan di-download ke mikrokontroler.

Diagram blok proses kompilasi program assembler seperti terlihat pada gambar 2.34 dibawah ini.



Gambar 2.32. Proses Kompilasi Program Assembler

### 2.8.7 Pembuatan dan Pengisian Program ke MCS-51

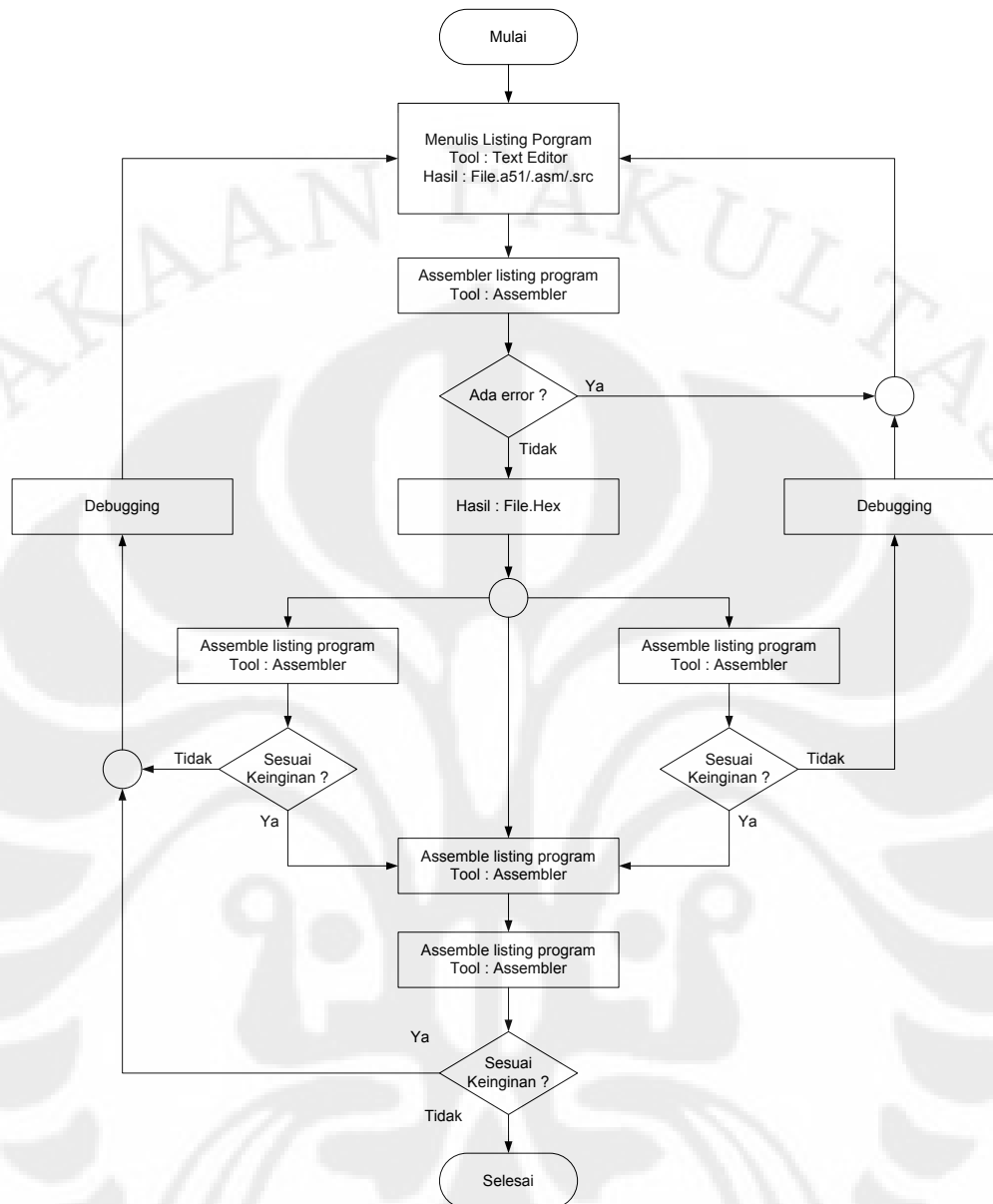
Secara garis besar, pembuatan dan pengisian program ke mikrokontroler dilakukan melalui beberapa tahap sebagai berikut :

1. Merancang algoritma program yang diinginkan (*flow - chart*).
2. Membuat source program dengan mengacu pada algoritma. Dalam bahasa pemrograman assembly, source code biasanya berupa file text yang ditulis dengan menggunakan text editor seperti Notepad, atau dengan menggunakan program (*software*) aplikasi khusus yang saat ini mulai banyak digunakan seperti software **Reads51**.
3. Meng-compile source program yang telah dibuat sehingga menjadi bahasa mesin (file tipe **.hex**). Jika source program yang dibuat menggunakan bahasa assembly maka di-compile menggunakan program assembler, sedangkan jika source program dibuat menggunakan bahasa tingkat tinggi (misal bahasa C) maka di-compile dengan menggunakan suatu program compiler. Pada software **Reads51**, source code dapat langsung di-compile dengan menggunakan fasilitas compiler yang sudah disediakan tanpa harus menggunakan program compiler tambahan.
4. Sebelum di-download ke mikrokontroler, hasil program dapat diuji coba terlebih dahulu, baik menggunakan simulasi software maupun emulasi hardware. Uji coba dengan simulasi software dapat menunjukkan hasil program pada simulasi komputer, sedangkan emulasi hardware dilakukan dengan menggunakan hardware emulator yang mewakili hardware mikrokontroler sehingga uji coba lebih nyata (*real*).

5. Langkah selanjutnya adalah mengecek apakah hasil uji coba sudah sesuai dengan yang diharapkan. Jika sudah sesuai, file **.hex** hasil compile ini dapat langsung di-download ke mikrokontroller. Sedangkan jika terjadi kesalahan, maka dapat dilakukan *debugging* untuk mencari letak kesalahan program.
6. Setelah diketahui letak kesalahan program, langkah berikutnya adalah memperbaiki kesalahan tersebut dan meng-compile ulang source code sehingga didapatkan file tipe **.hex** yang sudah benar.
7. File tipe **.hex** hasil compiler inilah yang akan diisikan kedalam memori program ( ROM ) mikrokontroler dengan menggunakan software downloader yang banyak tersedia seperti **AEC ISP Downloader** dan **ATMEL Microcontroller ISP Software**.

#### **Microcontroller ISP Software**

Alur tahapan pembuatan dan pengisian program ke mikrokontroller dapat dilihat pada gambar 2.35 dibawah ini.



Gambar 2.33. Diagram Alir Pemrograman Mikrokontroler

### Cara Pengisian Program

1. Hubungkan konektor DB-25 kabel ISP ke LPT1 (port paralel) komputer.
2. Hubungkan konektor White Housing kabel ISP modul CCBoard-51.
3. Hubungkan power supply ke modul CCBoard-51.
4. Jalankan program AEC\_ISP.
5. Pilih **Setup**
6. Pilih seri mikrokontroler yang akan digunakan (dalam hal ini AT89S51)
7. Pilih jenis port printer yang akan digunakan (LPT1/LPT2).

8. Jika program mau dikunci agar tidak bisa di bajak, Lock protect bit1, Lock protect bit2, dan Lock protect bit3 pilih YES
9. Untuk keluar pilih Save setup
10. Pilih menu Load Hex file to Flash buffer untuk membuka file .HEX yang akan di download ke Mikrokontroler. Kemudian ketikan nama file .HEX tersebut.
11. Pilih menu **Program** untuk memulai proses download file ke mikrokontroler
12. Setelah selesai proses download, untuk melihat hasil program tanpa melepas kabel ISP, pilih menu **Reset** menjadi **Low**

### **2.8.8 Mode Pengalamatan**

Mode pengalamatan adalah cara penulisan operand dari suatu instruksi, untuk mengalamatkan suatu data yang diinginkan, baik asalnya dan juga tujuannya. Bahasa pemrograman assembly untuk mikrokontroler MCS – 51 memiliki 8 jenis mode pengalamatan.

#### **a. Register Addressing**

Adalah pengalamatan yang melibatkan register.

#### **b. Direct Addressing**

Adalah pengalamatan langsung yang dilakukan dengan cara menuliskan langsung nomor alamat memori ( biasanya dalam format heksadesimal ).

#### **c. Indirect Addressing**

Adalah pengalamatan tidak langsung yang dilakukan dengan cara meletakkan alamat yang dimaksud ke suatu register ( khususnya register R0 dan R1 ).

#### **d. Immediate Addressing**

Adalah pengalamatan yang dilakukan dengan cara menuliskan langsung data yang diinginkan. Pengalamatan data terjadi pada sebuah perintah ketika nilai operasi merupakan alamat dari data yang akan diisi atau yang akan dipindahkan.

#### **e. Relative Addressing**

Adalah pengalamatan secara relatif terhadap alamat yang ada diprogram counter. Dalam program pengalamatan ini dilakukan dengan menggunakan label, sedangkan yang menghitung offset relative-nya terhadap program adalah



assembler.

#### **f. Absolute Addressing**

Adalah pengalamatan dengan menggunakan 11 bit alamat pasti dari tujuan, sehingga dapat menjangkau sampai 2 Kbyte memori program. Dalam program pengalamatan ini dilakukan dengan menggunakan label, sedangkan yang menghitung alamat absolute-nya adalah assembler.

#### **g. Long Addressing**

Adalah pengalamatan dengan menggunakan 16 bit alamat pasti dari tujuan, sehingga dapat menjangkau sampai 64 Kbyte memori program. Dalam program pengalamatan ini dilakukan dengan menggunakan label, sedangkan yang menghitung alamat absolute-nya adalah assembler.

#### **h. Indexed Addressing**

Adalah pengalamatan dengan menggunakan offset dan base register tertentu. Pengalamatan ini digunakan untuk look up tables dan jump tables.

### **2.8.9 Instruksi – Instruksi Dalam MCS–51**

Secara keseluruhan MCS – 51 mempunyai 255 macam instruksi yang dikelompokkan menjadi dua kelompok instruksi, yaitu instruction set dan assembler directives. Instruction set adalah instruksi – instruksi yang digunakan untuk memprogram mikrokontroler MCS – 51 dengan bahasa assembly. Sedangkan assembler directive adalah instruksi – instruksi yang digunakan untuk mengarahkan program assembler.

#### **a. Instruction Set**

Berdasarkan fungsinya, instruksi – instruksi dalam bahasa assembly dibagi menjadi lima kelompok, yaitu :

##### **A. Kelompok Instruksi Transfer data**

Instruksi dasar untuk transfer data adalah MOV ( MOVE ), yang artinya memindahkan atau lebih tepatnya meng-copy, karena proses pemindahan data dengan instruksi MOV dari sumber ke tujuan sebenarnya adalah proses meng-copy data dari sumber tetap / tidak berubah. Proses perpindahan data dapat terjadi antar memori internal atau dengan memori eksternal.

## B. Kelompok Instruksi Aritmatika

Instruksi aritmatika terdiri dari penjumlahan ( **ADD** ), pengurangan ( **SUB** ), perkalian ( **MUL** ), pembagian ( **DIV** ) serta operasi penambahan isi register dengan 1 ( **INC** ) dan pengurangan isi register dengan 1 ( **DEC** ). Instruksi – instruksi dalam kelompok aritmatika ini selalu melibatkan Akumulator.

Tabel 2.13. Instruksi Aritmatika

Mnemonic	Operation	Addressing Modes				Waktu ( $\mu$ S )
		Dir	Ind	Reg	Imm	
ADD A,<byte>	A=A+<byte>	X	X	X	X	1
ADDC A,<byte>	A=A+<byte>+C	X	X	X	X	1
SUBB A,<byte>	A=A-<byte>-C	X	X	X	X	1
INC A	A=A+1	ACCUMULATOR SAJA				1
INC <byte>		X	X	X		1
INC DPTR	DPTR=DPTR+1	DPTR SAJA				2
DEC A	A=A-1	ACCUMULATOR SAJA				1
DEC <byte>	<byte>=<byte>-1	X	X	X		1
MUL AB	B:A = BxA	ACC DAN B SAJA				4
DIV AB	A= int (A/B) B= mod (A/B)	ACC DAN B SAJA				4
DA A	Decimal Adjust	ACCUMULATOR SAJA				1

### • Operasi Penjumlahan ( **ADD** dan **ADDC** )

Digunakan untuk menambahkan isi akumulator dengan bilangan 1 byte ( bisa berasal dari bilangan konstan, register serba guna, memori data yang dialamati secara direct maupun indirect ), dan hasil penjumlahan akan ditampung kembali dalam akumulator.

ADD A,#35H ; tambahkan isi A dengan bilangan 127

; hasilnya disimpan di A (immediate

addressing)

ADD A,R7 ; tambahkan isi A dengan isi R7 hasilnya

; disimpan di A ( register addressing )

ADD A,7FH ; tambahkan isi A dengan isi memori lokasi

; 7FH

; hasilnya disimpan di A (direct addressing)

ADD A,@R0 ; tambahkan isi A dengan isi memori yang

; alamatnya ditunjukkan oleh R0 hasilnya di

; simpan di A ( indirect addressing )

- **Operasi Pengurangan ( SUBB )**

Digunakan untuk mengurangi isi akumulator A dengan bilangan 1 byte berikut dengan nilai bit carry, hasil pengurangan akan ditampung kembali dalam akumulator. Dalam operasi ini bit carry juga berfungsi sebagai penampung limpahan hasil pengurangan. Jika hasil pengurangan tersebut melimpah ( nilainya kurang dari 0 ) bit carry akan bernilai '1', sebaliknya bit carry akan bernilai ',0'.

Contoh :

SUBB A,R0 ; A=A-R0-Carry

SUBB A,#23H ; A=A-23H

SUBB A,@R1 ; A=A-[R1] ( [R1] artinya isi dari R1 )

SUBB A,P0 ; A=A-P0

- **Operasi Perkalian ( MUL AB )**

Instruksi **MUL AB** digunakan untuk mengalikan data 8 bit di akumulator dengan data 8 bit yang ada pada register B dan meletakkan hasil 16 bit ke dalam register A dan B. Register A berisi low-byte ( 8 bit rendah ) sedangkan register B berisi high-byte ( 8 bit atas ). Bit OV dalam PSW ( *Program Status Word* ) digunakan untuk menandai nilai hasil perkalian yang ada dalam register B. Jika hasilnya lebih besar dari 255 ( 0FFH ), maka bit overflow di-set ( 1 ), sedangkan bit carry selalu akan di clear-kan ( 0 ).

Contoh :

MOV A,#10

MOV B,#20

MUL AB

- **Operasi Pembagian ( DIV )**

Proses pembagian dilakukan dengan menggunakan Instruksi **DIV AB** yang akan membagi isi akumulator dengan data dalam register B dan meletakkan hasil bagi (*quotient*) 8 bit dalam akumulator, dan sisanya (*remainder*) 8 bit dalam register B. Operasi DIV akan membuat bit – bit carry dan overflow menjadi „0’.

- **Incremental ( INC ) dan Decremental ( DEC )**

Digunakan untuk menaikkan atau menurunkan data di dalam memori internal sebanyak 1 tanpa melalui akumulator. Instruksi yang digunakan adalah **INC** (*Increment*) dan **DEC** (*Decrement*). Pada DPTR operasi ini digunakan untuk menghasilkan pengalamatan 16 bit di memori eksternal.

Contoh :

DEC	A	; A=A-1
DEC	Ri	; Ri=Ri-1 ( Ri bisa R0 s/d R7 )
DEC	#50h	; hasilnya 4Fh
DEC	@Ri	; [Ri] = [Ri]-1
INC	A	; A=A+1
INC	Ri	; Ri=Ri+1
INC	#50h	; hasilnya 51h
INC	@Ri	; [Ri] = [Ri]+1

- **Decimal Adjust ( DA )**

Instruksi **DA** digunakan untuk operasi aritmatika pada bilangan BCD. Perintah DA digunakan setelah ADD dan ADDC atau SUBB untuk merubah nilai biner 8 bit yang tersimpan dalam akumulator menjadi bilangan BCD.

### C. Kelompok Instruksi Logika

Kelompok instruksi logika digunakan untuk melakukan operasi logika, yaitu operasi AND (instruksi **ANL**), operasi OR (instruksi **ORL**), operasi Exclusive-OR (instruksi **XRL**), operasi clear (instruksi **CLR**), instruksi negasi / komplement (instruksi **CPL**), operasi geser kanan atau kiri (instruksi **RR**, **RRC**, **RL**, dan **RLC**) serta operasi penukaran data (instruksi **SWAP**). Operasi logika

dilakukan pada bit demi bit dalam suatu byte data yang terdapat pada register.

Tabel 2.14. Instruksi Logika

Mnemonic	Operation	Addressing Modes				Waktu (µS)
		Dir	Ind	Reg	Imm	
ANL A,<byte>	A=A AND <byte>	X	X	X	X	1
ANL <byte>,A	<byte>=<byte> and A	X				1
ANL <byte>,#data	<byte>=<byte> and #data	X				2
ORL A,<byte>	A=A OR <byte>	X	X	X	X	1
ORL <byte>,A	<byte>=<byte> OR A	X				1
ORL <byte>,#data	<byte>=<byte> OR #data	X				2
XRL A,<byte>	A=A XOR <byte>	X	X	X	X	1
XRL <byte>,A	<byte>=<byte> XOR A	X				1
XRL <byte>,#data	<byte>=<byte> XOR #data	X				2
CRL A	A = 00H	Accumulator saja				1
CPL A	A = NOT A	Accumulator saja				1
RL A	Rotate ACC left 1 bit	Accumulator saja				1
RLC A	Rotate left through C	Accumulator saja				1
RR A	Rotate ACC right 1 bit	Accumulator saja				1
RRC A	Rotate right through C	Accumulator saja				1
SWAP A	Swap Nibbles in A	Accumulator saja				1

- Operasi logika **AND**

Biasanya digunakan untuk me-nol-kan beberapa bit tertentu dari sebuah data 8 bit. Cara yang dilakukan adalah data biner 8 bit di-ANL-kan dengan suatu bilangan tertentu disesuaikan dengan bit yang akan di-nol-kan.

Contoh :

```
MOV  A,#10101010B      ; isi A dengan data 10101010
ANL  A,#10011001B      ; isi A di-AND-kan dengan
                        10011001
                        ; hasilnya = 10001000 disimpan di A
```

- Operasi logika **OR**

Biasanya digunakan untuk men-satu-kan beberapa bit tertentu dari sebuah data 8 bit. Cara yang dilakukan adalah data biner 8 bit di-OR-kan dengan suatu bilangan tertentu disesuaikan dengan bit yang di-satu-kan.

Contoh :

```
MOV A,#10101010B ; isi A dengan data 10101010
ORL A,#10011001B ; isi A di-OR-kan dengan 10011001
; hasilnya = 10111011 disimpan di A
```

- Operasi logika **EXCLUSIVE – OR**

Biasanya digunakan untuk membalik nilai beberapa bit tertentu dari sebuah bilangan biner 8 bit. Cara yang dilakukan adalah data biner 8 bit di-XOR-kan dengan suatu bilangan tertentu disesuaikan dengan bit yang akan dibalik.

Contoh :

```
MOV A,#10001000B ; isi A dengan data 10001000
ORL A,#11111111B ; isi A di-XOR-kan dengan
11111111
; hasilnya = 01110111 disimpan di A
```

- Instruksi **RL, RR, RLC, dan RRC**

Digunakan untuk menggeser bit – bit dalam suatu data byte ke kiri atau ke kanan, melalui atau tidak melalui carry.

- Instruksi **SWAP**

Digunakan untuk menukar data nibble ( 4 bit ) dari suatu data byte ( 8 bit ) di akumulator, dimana nibble rendah ditukar dengan nibble tinggi.

Contoh :

```
MOV A,#0FAh ; isi register A dengan data FAh
SWAP A ; isi A menjadi Afh
```

#### **D. Kelompok Instruksi Boolean**

Kelompok instruksi Boolean digunakan untuk melakukan operasi bit tunggal. Kelompok instruksi ini terdiri dari : instruksi **Pemberian Nilai Data Biner**, **instruksi Pemindahan Data Biner**, **instruksi Operasi logika Bit**, dan **instruksi Pengujian Nilai Boolean**.

- **Pemberian Nilai Data Biner**

Terdiri dari dua ( 2 ) instruksi yaitu **SETB** ( *set bit* ) dan **CLR** ( *Clear* ). SETB digunakan untuk memberikan nilai '1' pada data satu bit tertentu, sedangkan CLR digunakan untuk memberikan nilai '0' pada satu bit tertentu.

Contoh :

```

SETB ACC.0      ; bit 0 akumulator bernilai '1'
SETB P0.0       ; bit 0 port 0 bernilai '1'
CLR  P1.1       ; bit 1 port 1 bernilai '0'

```

- **Pemindahan Data Biner**

Digunakan untuk memindahkan data dalam level bit dari satu posisi ke posisi yang lain dengan bantuan *bit carry* di dalam PSW ( lokasi PSW D0h ) yang bersifat sebagai akumulator bit.

Contoh :

```

MOV  C,P1.1     ; pindahkan data P1.1 ke bit carry
MOV  P1.0,C     ; pindahkan isi bit carry ke P1.0

```

Perintah di atas digunakan untuk memindahkan data dari P1.1 ke P1.0 memulai bit carry sehingga data pada P1.0 akan sama dengan data pada P1.1.

- **Operasi Logika ( Bit )**

Sama halnya seperti pada data byte, operasi logika juga dapat digunakan untuk data level bit. Dalam operasi logika untuk level bit ini, bit carry pada PSW diperlukan sebagai akumulator.

Contoh :

```

ANL  C,P1.1     ;nilai pada bit carry di-AND-kan dengan
                ;nilai
                ;nilai di P1.1 dan hasilnya disimpan di bit
                ;carry
ANL  C,P1.2     ;nilai komplemen P1.2 di-AND-kan dengan
                ;bit carry dan hasilnya disimpan di bit carry

```

- **Pengujian Nilai Boolean**

Pengujian nilai Boolean dilakukan dengan instruksi **JUMP bersyarat**. Ada lima ( 5 ) instruksi yang digunakan yaitu : instruksi **JB** ( *Jump if Bit set* ), **JNB** ( *Jump if Not set* ), **JC** ( *Jump if Carry bit set* ), **JNC** ( *Jump if Carry bit Not set* ), dan **JBC** ( *Jump if Bit set and Clear bit* ).

Contoh :

```

JB   P1.1,$     ; cek nilai P1.1, jika bernilai '1'

```

```

; ulangi instruksi ini
JNB P1.1,$ ; cek nilai P1.1, jika berniali '0'
; ulangi instruksi ini
JC PERIKSA ; cek bit carry, jika bernilai '1'
; lompat ke PERIKSA

```

### E. Kelompok Instruksi Percabangan

Dalam pemrograman assembler, biasanya diperlukan percabangan untuk tujuan tertentu, yaitu lompat ke lokasi instruksi dengan alamat tertentu. Percabangan program terdiri dari : **pelaksanaan sub rutin**, **percabangan tanpa syarat** ( *unconditional jump* ), dan **percabangan bersyarat** ( *conditional jump* ).

- **Subrutin**

Subrutin adalah penggalan program yang sering digunakan dalam suatu program ( dipanggil berkali – kali ) tanpa harus menulis ulang perintahnya. Instruksi yang digunakan untuk memanggil subrutin antara lain **CALL**, **RET**, dan **RETI**.

Perintah **CALL** digunakan untuk memanggil subrutin dan terdiri dari **ACALL** ( untuk memanggil sub rutin dengan jarak maksimum 11 address line / 2Kbyte memori program ) dan **LCALL** ( untuk memanggil sub rutin dengan jarak maksimum 64Kbyte memori program ).

Contoh :

```

TUNGGU:   ACALL DELAY      ; memanggil subrutin DELAY
;-----
; subrutin DELAY
;-----
DELAY:    MOV    R1,#0
DELAY1:   MOV    R2,#0
DELAY2:   DJNZ  R2,DELAY2
          DJNZ  R1,DELAY1
RET      ; kembali dari
subrutin

```



Subrutin harus diakhiri oleh instruksi RET ( *Return From Subroutine* ) yang menandakan bahwa subrutin telah selesai dilaksanakan dan program kembali ketempat semula. Sedangkan perintah RETI digunakan untuk kembali dari *Interrupt Service Routine* ( ISR ). Jika tidak ada interupsi lain yang menunggu maka RETI berfungsi sama seperti RET.

- **Pencabangan Tanpa Syarat ( *Unconditional Jump* )**

Pencabangan tanpa syarat adalah lompat ke lokasi alamat tertentu tanpa adanya suatu persyaratan untuk mengeksekusi penggalan program pada alamat tersebut dan setelah penggalan program tersebut dilaksanakan biasanya tidak kembali ke tempat semula. Instruksi yang digunakan adalah SJMP ( pencabangan relatif, untuk lompat sejauh -128 ( mundur ) atau +127 ( maju )), LJMP ( pencabangan jauh, dapat melompat hingga 64Kbyte memori program ), dan AJMP ( pencabangan absolut, dapat melompat sejauh 2 Kbyte memori program ).

Contoh :

```
START:   CLR   P2.0           ; P2.0 di clear
          MOV  A,P2         ; pindahkan data di
                               ; P2 ke A
SATU:    CJNE  A,#01101110b,DUA
          MOV  P0,#11111110b
          MOV  P1,#ANGKA1
          SJMP START       ; lompat ke START
```

- **Pencabangan Bersyarat**

Pencabangan bersyarat adalah lompat ke alamat tertentu bila persyaratan terpenuhi. Instruksi yang biasa digunakan adalah CJNE ( *Compare and Jump if Not Equal* ).

Contoh :

```
START:
          MOV  R0,#5        ; isi R0 dengan 5
          MOV  R1,#40h      ; isi R1 dengan 40h
```

LOOP:

```
MOV @R1,#0Aah      ; isi memori di < R1 >
                   ; dengan 0Aah
INC R1              ; R1=R1+1
DJNZ R0,LOOP       ; R0=R0-1, jika belum 0
                   ; kembali ke LOOP
SJMP $              ; usai
```

- **Jump Table**

Perintah `JMP @A+DPTR` digunakan untuk perpindahan yang bersyarat ganda. Alamat tujuan dihitung sebagai jumlah dari `DPTR` 16-bit sebagai jump table dan akumulator sebagai indeksinya. Jika ada 5 syarat yang diinginkan, nilai 0 sampai 4 dimasukkan ke akumulator.

Contoh :

```
MOV DPTR,#JMP_TABLE
JMP @A+DPTR
JMP_TABLE: AJMP LABEL0
           AJMP LABEL1
           .....
```

- b. Pengarah Assembler ( Assembler Directive )**

Assembler Directive adalah instruksi – instruksi yang ditujukan pada program assembler dan digunakan sebagai arahan dalam mengubah program yang ditulis dalam bahasa assembly menjadi bahasa mesin. Beberapa jenis Assembler Directives yang paling banyak digunakan ( assembler ASM 51 ) adalah :

- A. Pengarah Untuk Kontrol Kondisi Assembler ( Assembler State Control )**

Terdiri dari :

- **ORG ( Set Origin )**, digunakan untuk menunjukkan lokasi alamat awal dari suatu program atau penggalan program dalam memori program untuk instruksi – instruksi atau label dibawahnya. Tidak mencantumkan ORG dalam program, dianggap program dimulai dari 0000h.

- **END**, adalah Assembler Directive yang menunjukkan akhir dari program. Tidak ada baris program lagi yang akan dieksekusi setelah pengarah ini.
- **USING**, digunakan untuk memberitahukan assembler ASM 51 tentang register bank yang sedang aktif.

#### **B. Pengarah Untuk Pendefinisian Simbol ( *Symbol Definition Directive* )**

Terdiri dari :

- **EQU ( *Equate* )**, berarti “sama dengan” atau “identik”. Digunakan untuk menetapkan suatu nilai numerik yang tetap pada sebuah simbol. Nilai numerik adalah data, alamat, atau register.

#### **C. Pengarah Untuk Pemilihan Segmen ( *Segment Selection Directive* )**

Terdiri dari :

- **CSEG ( *Code Segment* )**, digunakan untuk memilih suatu segmen absolut baru pada kode segmen / memilih lokasi memori program.
- **DSEG ( *Data Segment* )**, yaitu untuk memilih segmen absolut baru pada data segmen / memilih lokasi memori RAM internal.
- **ISEG ( *Interval Data Segment* )**, yaitu memilih suatu absolut segmen baru pada internal data segmen / memilih lokasi memori RAM internal yang dialamati secara tidak langsung.
- **BSEG ( *Bit Data Segment* )**, yaitu memilih suatu absolut segmen baru pada bit data segmen / memilih lokasi memori yang dapat dialamati secara pengalamatan bit.
- **XSEG ( *External Data Segment* )**, yaitu memilih suatu absolut segment baru pada eksternal data segmen / memilih lokasi memori eksternal.

#### **D. Pengarah Untuk Pemesanan Memori Penyimpan ( *Memory Reservation and Storage Directive* )**

Terdiri dari :

- **DS ( *Define Storage* )**, digunakan untuk reservasi memori penyimpanan dalam satuan byte.

- **DBIT** ( *Define Bit* ), digunakan untuk reservasi memori penyimpanan dalam satuan bit.
- **DB** ( *Define Byte* ), digunakan untuk reservasi memori penyimpanan di code segment dalam satuan byte.
- **DW** ( *Define Word* ), digunakan untuk reservasi memori penyimpanan di code segment dalam satuan word.

#### **E. Pengarah Untuk Penggabungan Program**

Terdiri dari :

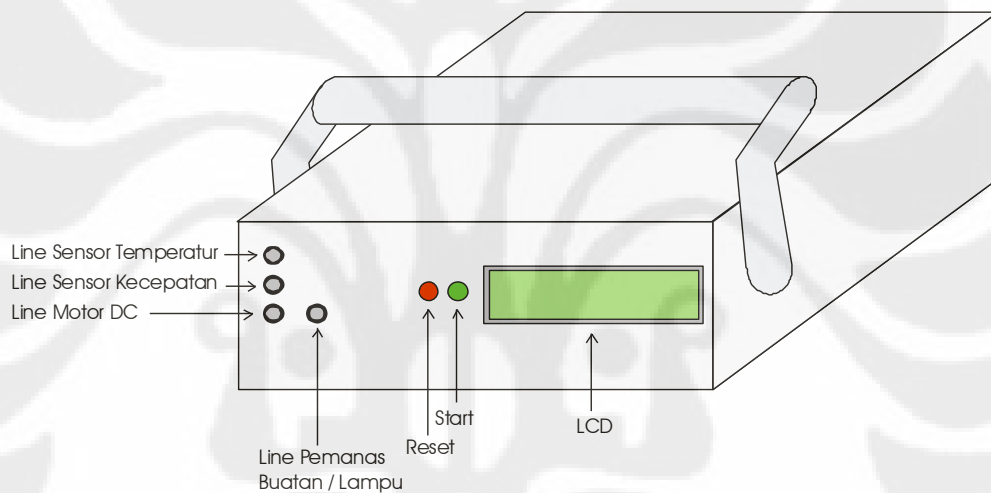
- **PUBLIC**, yaitu membuat agar suatu simbol / label juga berlaku di luar modul / bagian program yang memuatnya ( membuat menjadi bersifat global ).
- **EXTERN**, yaitu mereferensikan suatu simbol yang telah dideklarasikan secara global di modul / bagian program lain. Pengarah ini bekerjasama dengan pengarah PUBLIC.

## BAB 3 PERANCANGAN

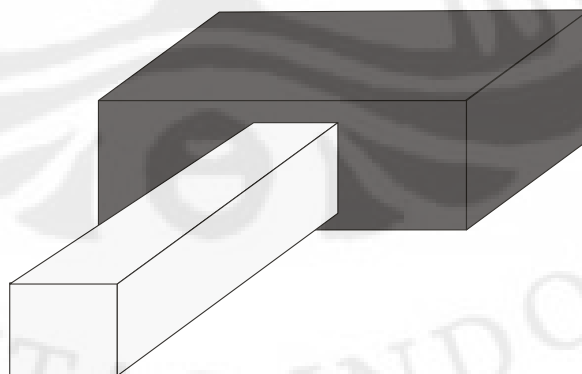
### 3.1 Perancangan Hardware

#### 3.1.1 Perancangan Fisik

Perancangan fisik yang dimaksud disini adalah suatu bagian yang berfungsi untuk melindungi bagian elektrik dan untuk meningkatkan penampilan dari alat, dimana hasil perancangan ini dapat disebut Box. Box pada alat ini menggunakan bahan Plastik dengan ketebalan 3 mm, dengan desain seperti gambar dibawah ini.



Gambar 3.1. Box Pelindung Rangkaian

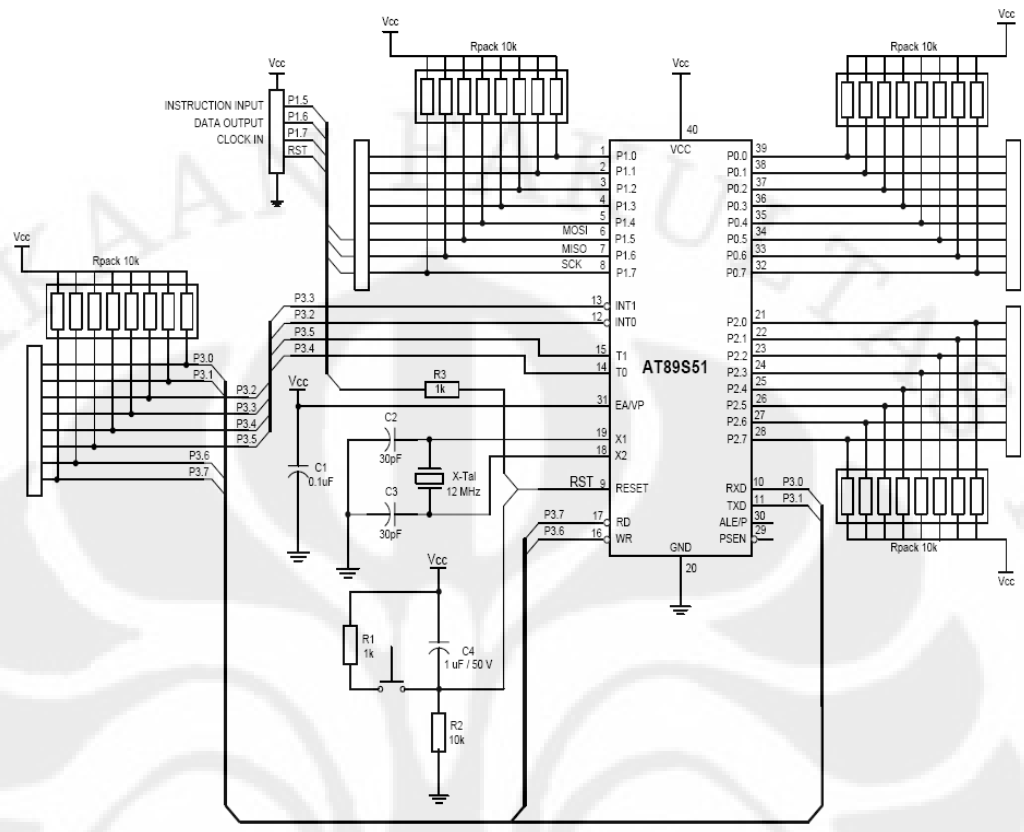


Gambar 3.2. Prototype / Simulasi Ruang

### **3.1.2 Perancangan Elektrik**

#### **3.1.2.1 Rangkaian Minimum Sistem**

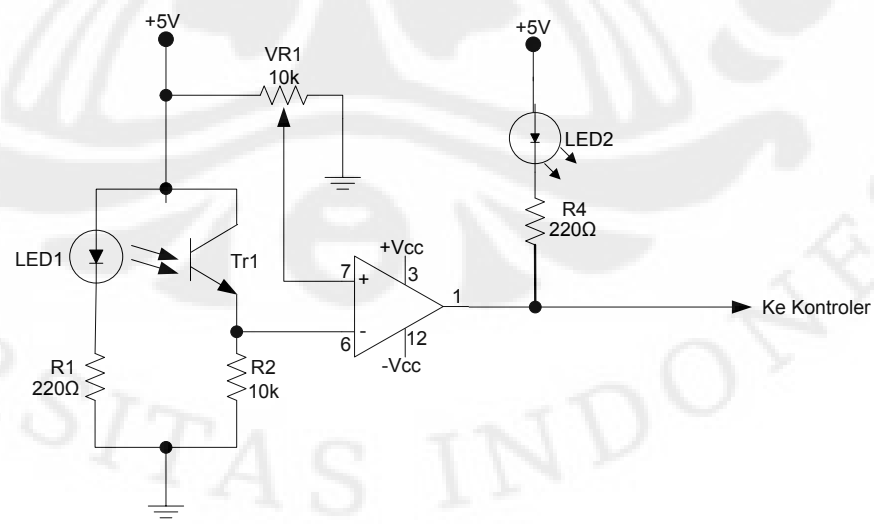
Pada gambar rangkaian minimum system dibawah chip mikrokontroller AT89S51 yang terdiri dari empat buah jalur I/O. Port 3 yaitu pin P3.0 s.d port P3.7, digunakan sebagai input masukan sinyal pada rangkaian LCD untuk mengendalikan sinyal Data ( D0 – D7 ) dari LCD. Port 1 yaitu pin P1.0 s.d P1.7, digunakan sebagai penerima sinyal input yang berasal rangkaian ADC. Sedangkan untuk pin P2.0 digunakan untuk memberikan sinyal input kepada pin WR dari ADC0804 sehingga ADC dapat mengeluarkan atau menampilkan output biner hasil Konversi sinyal analog yang berasal dari sensor temperature. Pin P2.2 digunakan untuk mengendalikan rangkaian driver Motor DC, sehingga motor DC dapat bekerja atau tidak bekerja atau dalam perancangan ini sebagai pin output untuk mengirimkan sinyal PWM yang berasal dari kontroler ke rangkaian driver Motor DC. Pin P2.1 berfungsi sebagai input pada rangkaian kontrol agar sistem mulai berjalan atau bekerja (tombol start). Pin P2.7 digunakan sebagai penerima sinyal output dari rangkaian sensor kecepatan putaran Motor DC, dimana sinyal ini sebagai input bagi mikrokontroler untuk melakukan perhitungan mundur (decrement) register "R0". Sedangkan Pin P2.5, P2.6 dan P2.7 berfungsi untuk menghasilkan sebagai sinyal input untuk rangkaian LCD pada pin EN, RW dan RS.



Gambar 3.3. Rangkaian Minimum Sistem

**3.1.2.2 Rangkaian Sensor Kecepatan Putaran Motor DC**

Rangkaian sensor kecepatan putaran motor DC dapat dilihat pada gambar berikut :



Gambar 3.4. Rangkaian Sensor Kecepatan Putaran Motor DC

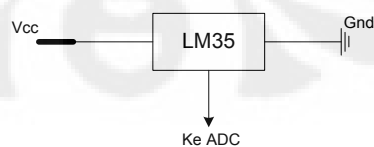
Komponen utama yang digunakan dalam sensor ini menggunakan photo transistor dan sebuah infra merah yang dikemas dalam satu paket dan bekerja secara reflektif, yaitu dengan menerima sinar infra merah yang telah dipantulkan oleh objek. Komponen utama lain yang digunakan dalam sensor ini adalah komparator, digunakan untuk membandingkan suatu kondisi dengan kondisi lainnya. Karena pada Motor DC akan diberi suatu tanda dengan menggunakan warna putih, sehingga pada saat sensor mendeteksi warna putih, maka menandakan motor telah melakukan putaran sebanyak 1 kali.

Pada saat sensor tidak mendeteksi warna putih maka output sensor akan berlogika “1” dan pada saat sensor mendeteksi warna putih, maka output sensor akan berlogika “0”. Dimana output sensor ini akan dikirimkan ke rangkaian kontroler.

Prinsip kerja rangkaian ADC secara detail adalah pada saat  $T_{r1}$  mendapatkan sinar infra red yang dipantulkan oleh objek, maka tegangan pada  $R_2$  akan lebih besar dari tegangan di  $T_{r1}$ . Karena  $V_{input}$  non-inverting (pin7) di adjust lebih kecil dari tegangan  $V_{input}$  inverting (pin6) pada saat aktif, maka output dari komparator akan berlogika “0” dan LED<sub>2</sub> sebagai indikator akan menyala. Dan sebaliknya saat tidak ada objek yang memantulkan sinar infra red, maka output komparator akan berlogika “1”.

### 3.1.2.3 Sensor Temperatur dan Rangkaian ADC

Sensor temperature disini menggunakan LM35, yaitu sebuah sensor temperature yang dikemas dalam bentuk IC.

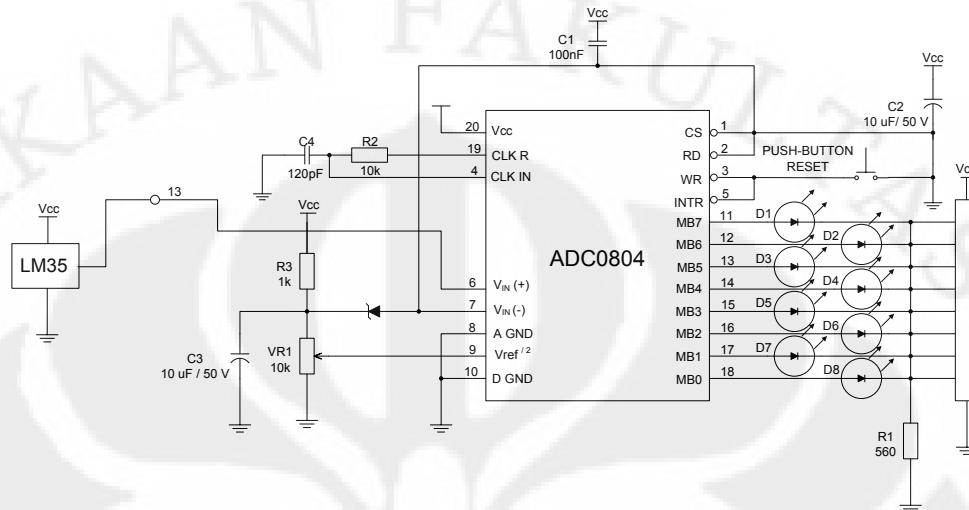


Gambar 3.5. Layout Pin IC LM35

Output Sensor temperature ini diumpangkan terlebih dahulu ke rangkaian Analog to Digital Converter (ADC), agar sinyal analog dari LM35 dapat diubah ke sinyal digital 8 bit, sehingga dapat dibaca oleh Mikrokontroler



untuk mendapatkan suatu perintah yang tepat bagi sistem. Adapun rangkaian ADC dimaksud dapat dilihat pada gambar dibawah ini.

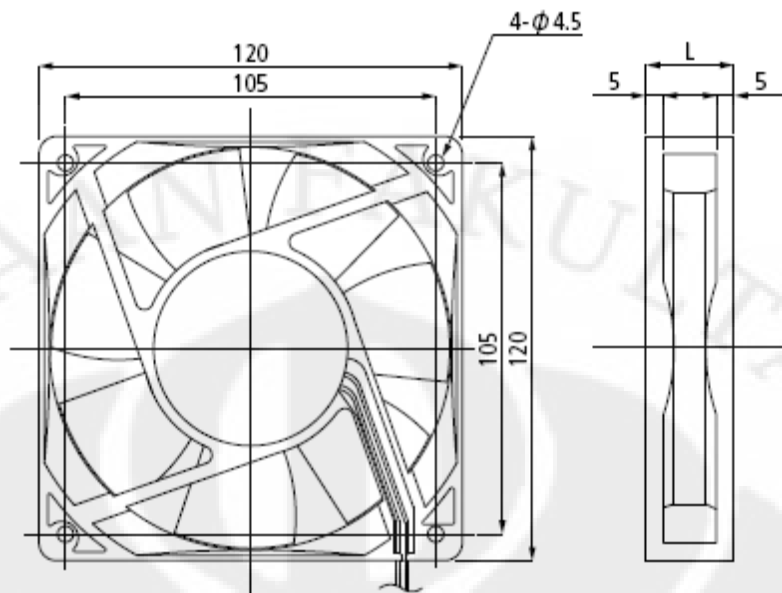


Gambar 3.6. Rangkaian Analog to Digital Converter (ADC)

LED pada rangkaian ADC digunakan sebagai indikator sinyal digital atau logika biner. Sehingga dapat memudahkan dalam pembacaan untuk pembuatan program. Untuk proses pembacaan dan pengiriman sinyal analog, maka perlu memberikan logika yang tepat (“0” atau “1”) pada pin RD dan WR dari IC ADC0804.

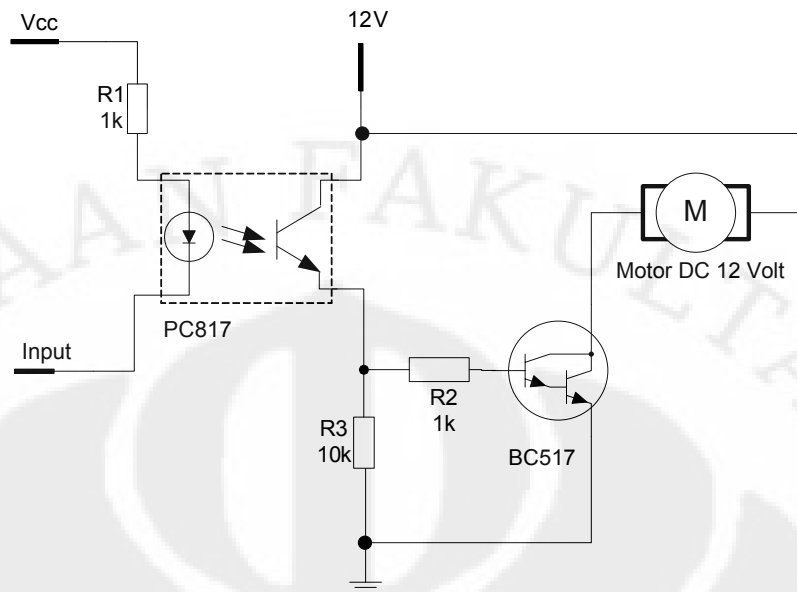
### 3.1.2.4 Rangkaian Pengendali Motor DC 12 Volt

Pada perancangan ini akan digunakan motor yang memiliki daya yang tidak terlalu tinggi, yaitu menggunakan motor arus searah (DC) dengan penguat sendiri (motor seri), dimana motor ini mempunyai magnet permanen pada statornya dan memperoleh sumber arus searah dari motor itu sendiri. Spesifikasi dari motor yang akan digunakan adalah sebagai berikut :



Gambar 3.7. Bentuk Motor DC

Model	: CF120-T / H201N1D
Rated Voltage	: 12 Volt
Rated Input (Power)	: 3,72 Watt
Rated Speed	: 2.400 r/min
Rated Current	: 310 mA
Max. Air Flow	: 2,15 m <sup>3</sup> /min
Max. Static Pressure	: 50 Pa
Sound Level	: 39 dB(A)



Gambar 3.8. Rangkaian Pengendali Motor DC

Pada rangkaian pengendali Motor DC ini terdiri dari 2 komponen utama yang berfungsi sebagai saklar atau pengontrol kondisi on dan off dari Motor DC serta beberapa komponen pendukung. Komponen utama dari rangkaian dimaksud adalah Photo Transistor type PC817 dan Transistor Darlington jenis NPN type BC517. Komponen pendukung terdiri dari resistor yang berfungsi sebagai rangkaian pembagi tegangan dan pembatas arus listrik.

Secara garis besar prinsip kerja dari rangkaian pengendali Motor DC adalah pada saat pin input dari rangkaian diberikan logika “0”, maka arus listrik akan mengalir dari  $V_{CC}$  melalui resistor 1 k $\Omega$  dan LED, sehingga LED dapat memancarkan cahaya yang diperlukan sebagai bias transistor agar dapat bekerja. Ketika transistor dalam kondisi on, maka seperti sebuah saklar yang tertutup sehingga arus listrik dapat mengalir dari sumber tegangan 12 Volt melalui Motor DC menuju ground, sehingga Motor DC dapat bekerja. Sebaliknya, saat transistor rdalam kondisi off, maka seperti sebuah saklar terbuka sehingga arus listrik tidak dapat mengalir dan Motor DC tidak dapat bekerja.

Dari type Motor DC yang digunakan diketahui spesifikasi sebagai berikut :

$$V = 12 \text{ Volt}$$

$$I = 310 \text{ mA}$$

$$R = \frac{V}{I} = \frac{12\text{Volt}}{310\text{mA}} = 38.709\Omega$$

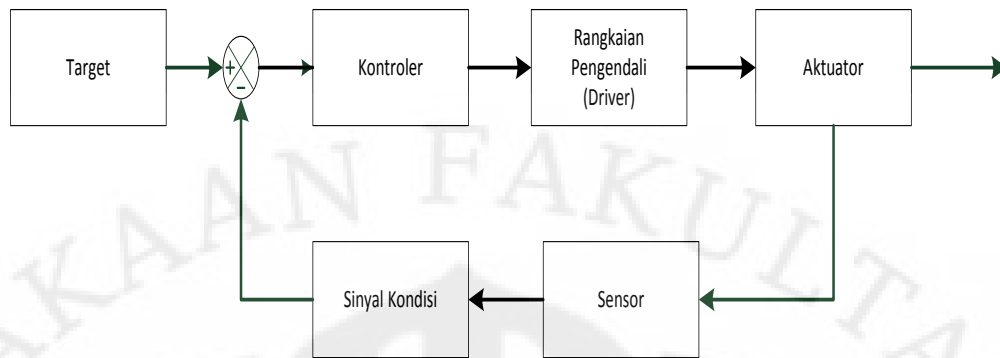
Dari nilai tegangan, arus dan hambatan Motor DC diatas, sebagai analisa rancangan digunakan asumsi tegangan input yang variabel yaitu sebesar 6 V, 10 V, 18 V dan 24 V. Maka dengan menggunakan nilai hambatan Motor DC diatas untuk setiap nilai tegangan input, maka akan didapati nilai arus yang akan melalui Motor DC sebesar sebagai berikut :

Tabel 3.1. Perhitungan Perubahan Nilai Arus dan Daya Terhadap Perubahan Nilai Tegangan Input

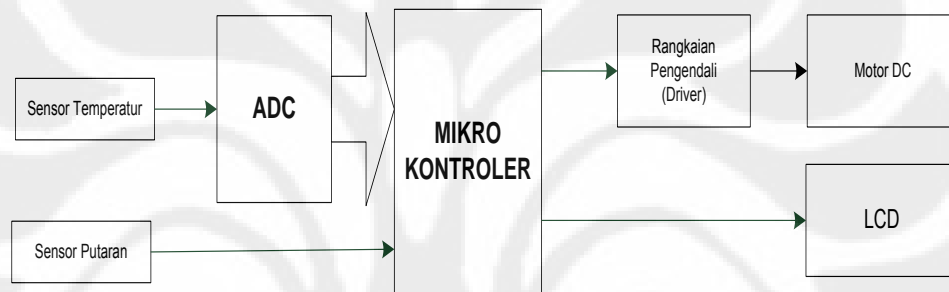
No.	V <sub>in</sub> (Volt)	Resistansi (Ω)	I <sub>in</sub> (mA)	P <sub>in</sub> (Watt)
1.	6	38,709	155	0.93
2.	10	38,709	258	2.58
3.	18	38,709	465	8.37
4.	24	38,709	620	14.88

Dari tabel diatas dapat dijelaskan bahwa dengan merubah besaran dari tegangan input, maka kecepatan putaran motor dapat berubah. Tetapi akibat dari pemberian tegangan input yang semakin besar, maka nilai arus yang melalui Motor DC akan semakin besar dan bila tegangan input yang diberikan melebihi tegangan kerja yang diperuntukkan untuk Motor DC, maka dapat meningkatkan daya (P<sub>in</sub>), yang dapat menimbulkan panas pada lilitan atau motor DC. Hal ini tentunya dapat merusak Motor DC bila berlangsung dalam waktu yang cukup lama. Untuk mengatasi hal tersebut akan digunakan sistem Modulasi Lebar Pulsa, yaitu dengan mengatur lebar pulsa sinyal input yang diumpankan kepada rangkaian pengendali Motor DC.

Secara garis besar sistem kendali yang akan dirancang untuk mengatur kecepatan putaran Motor DC secara otomatis ini dapat dijelaskan pada gambar dibawah ini.



Gambar 3.9. Diagram Blok Pengendalian Kecepatan Putaran Motor DC Terhadap Perubahan Temperatur Dengan Sistem Modulasi Lebar Pulsa



Gambar 3.10. Detail Diagram Blok Sistem

Prinsip kerja dari perancangan atau sistem ini adalah sebagai berikut :

Sensor temperatur akan mendeteksi temperatur sekitar untuk menghasilkan sinyal output berupa tegangan yang akan diinformasikan kepada mikrokontroler. Sinyal output ini diperlukan sebagai data informasi bagi mikrokontroler untuk menghasilkan sinyal PWM yang sesuai dengan perencanaan sistem. Sinyal PWM akan diumpankan kepada rangkaian pengendali (driver) motor untuk dapat menggerakkan Motor DC. Rangkaian pengendali (driver) motor ini juga berfungsi sebagai penguat sinyal PWM yang dihasilkan oleh mikrokontroler. Sensor putaran digunakan untuk mendeteksi kecepatan putaran dari Motor DC. Kecepatan putaran dan perubahan temperatur akan ditampilkan oleh LCD, sehingga kita dapat mengetahui kondisi dari setiap perubahan. Secara garis besar dengan semakin tinggi temperatur, maka kecepatan putaran Motor DC akan semakin cepat. Hal ini diperlukan untuk menjaga temperatur lingkungan mendekati kondisi stabil. Dan temperatur dari Motor DC itu sendiri, mengingat

temperatur Motor DC dapat dipengaruhi juga oleh temperatur lingkungan.

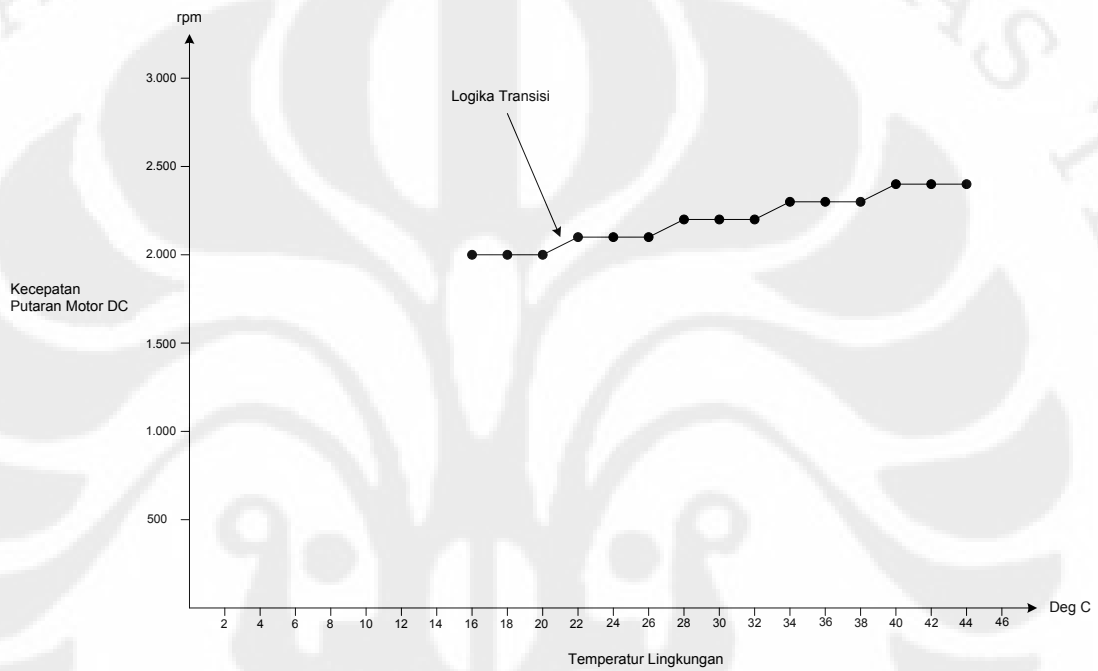
Berikut adalah tabel perencanaan sistem perubahan kecepatan putaran motor terhadap perubahan temperatur :

Tabel 3.2. Perubahan Kecepatan Putaran Motor DC Terhadap Perubahan Temperatur Lingkungan

No.	Temperatur (Yang Diinginkan)	Temperatur (Ref. ADC)	Kecepatan Putaran
1.	16 <sup>0</sup> C	16 <sup>0</sup> C	2.000 rpm
2.	17 <sup>0</sup> C	◇	
3.	18 <sup>0</sup> C	18 <sup>0</sup> C	2.000 rpm
4.	19 <sup>0</sup> C	◇	
5.	20 <sup>0</sup> C	20 <sup>0</sup> C	2.000 rpm
6.	21 <sup>0</sup> C	◇	
7.	22 <sup>0</sup> C	22 <sup>0</sup> C	2.100 rpm
8.	23 <sup>0</sup> C	◇	
9.	24 <sup>0</sup> C	24 <sup>0</sup> C	2.100 rpm
10.	25 <sup>0</sup> C	◇	
11.	26 <sup>0</sup> C	26 <sup>0</sup> C	2.100 rpm
12.	27 <sup>0</sup> C	◇	
13.	28 <sup>0</sup> C	28 <sup>0</sup> C	2.200 rpm
14.	29 <sup>0</sup> C	◇	
15.	30 <sup>0</sup> C	30 <sup>0</sup> C	2.200 rpm
16.	31 <sup>0</sup> C	◇	
17.	32 <sup>0</sup> C	32 <sup>0</sup> C	2.200 rpm
18.	33 <sup>0</sup> C	◇	
19.	34 <sup>0</sup> C	34 <sup>0</sup> C	2.300 rpm
20.	35 <sup>0</sup> C	◇	
21.	36 <sup>0</sup> C	36 <sup>0</sup> C	2.300 rpm
22.	37 <sup>0</sup> C	◇	
23.	38 <sup>0</sup> C	38 <sup>0</sup> C	2.300 rpm
24.	39 <sup>0</sup> C	◇	
25.	40 <sup>0</sup> C	40 <sup>0</sup> C	2.400 rpm
26.	41 <sup>0</sup> C	◇	

27.	42 <sup>0</sup> C	42 <sup>0</sup> C	2.400 rpm
28.	43 <sup>0</sup> C	◊	
29.	44 <sup>0</sup> C	44 <sup>0</sup> C	2.400 rpm

Dari tabel diatas, akan digunakan range temperatur dari 16<sup>0</sup>C sampai dengan 44<sup>0</sup>C dan kecepatan putaran Motor DC dengan range 2.000 rpm sampai dengan 2.400 rpm.



Gambar 3.11. Grafik Perubahan Kecepatan Putaran Motor DC Terhadap Perubahan Temperature Lingkungan

Penjelasan dari logika transisi pada gambar diatas yaitu kondisi logika biner yang dapat memasuki ke logika sebelumnya atau memasuki ke logika setelahnya. Hal ini disebabkan, karena adanya perbedaan level sinyal output yang dihasilkan oleh sensor temperature LM 35 dan level perubahan sinyal input yang dapat diproses oleh ADC. Pada sensor temperature setiap perubahan 1<sup>0</sup>C maka tegangan output yang dihasilkan sensor adalah 10 mV, sedangkan perubahan sinyal input yang dapat dibaca oleh ADC adalah setiap 20 mV.

Tabel 3.3. Temperatur Rata-rata per Bulan Tahun 2009 di DKI Jakarta

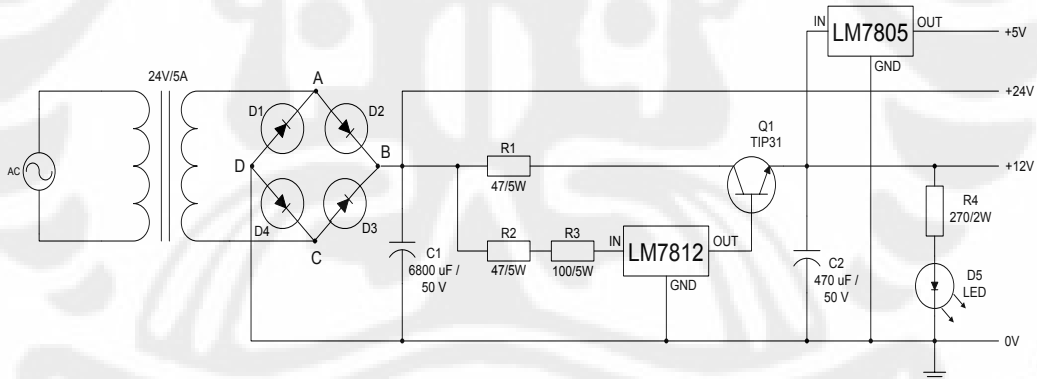
Cuaca untuk Jakarta													
Bulan	Jan	Feb	Mar	Apr	Mei	Jun	Jul	Agt	Sep	Okt	Nov	Des	Tahun
Rata2 tinggi	29,9	30,3	31,5	32,5	32,5	31,4	32,3	32,0	33,0	32,7	31,3	32,0	31,8

°C (°F)	(86)	(87)	(89)	(91)	(91)	(89)	(90)	(90)	(91)	(91)	(88)	(90)	(89)
Rata2 rendah °C (°F)	24,2 (76)	24,3 (76)	25,2 (77)	25,1 (77)	25,4 (78)	24,8 (77)	25,1 (77)	24,9 (77)	25,5 (78)	25,5 (78)	24,9 (77)	24,9 (77)	25,0 (77)
Presipitasi mm (inci)	384,7 (15,1)	309,8 (12,2)	100,3 (3,9)	257,8 (10,1)	133,4 (5,3)	83,1 (3,3)	30,8 (1,2)	34,2 (1,3)	29,0 (1,1)	33,1 (1,3)	175,0 (6,9)	84,0 (3,3)	1.655,2 (65,2)

[http://id.wikipedia.org/wiki/Daerah\\_Khusus\\_Ibukota\\_Jakarta#Iklim](http://id.wikipedia.org/wiki/Daerah_Khusus_Ibukota_Jakarta#Iklim) – Sumber : *World Meteorological Organization* <sup>[7]</sup> February 2010

Suhu kamar adalah 20°C s/d 25°C ([http://id.wikipedia.org/wiki/Suhu\\_kamar](http://id.wikipedia.org/wiki/Suhu_kamar)), namun mengingat temperature lingkungan rata-rata di DKI Jakarta sebagaimana tabel diatas, maka temperature referensi atau temperatur lingkungan dalam kondisi normal yang digunakan dalam perancangan ini adalah 28°C s/d 32°C dengan kecepatan putaran Motor DC adalah 2.200 rpm. Kecepatan maksimum putaran Motor DC yang digunakan mengacu kepada spesifikasi Motor DC yang digunakan, yaitu 2.400 rpm seperti telah dijelaskan diatas pada bab ini, sehingga dapat menghindari kerusakan karena batas kemampuan kerja yang dimiliki.

### 3.1.2.5 Rangkaian Catu Daya



Gambar 3.12. Rangkaian Catu Daya

Prinsip penyearahan tegangan dari rangkaian catu daya ini adalah sebagai berikut :

Pada saat A lebih positif dari C, maka arus listrik akan mengalir titik A melalui D2 menuju kapasitor C1 lalu melalui D4 menuju titik C. Saat C lebih positif dari A, maka arus listrik akan mengalir dari C melalui D3 menuju kapasitor C1 lalu melalui D1 menuju titik A. sehingga tegangan sumber yang berbentuk sinyal AC (bolak-balik) akan menjadi searah (DC). LM 7805 pada rangkaian catu daya



berfungsi untuk menghasilkan tegangan 5 Volt<sub>DC</sub> yang stabil. LM 7812 berfungsi untuk mendapatkan tegangan sebesar 12 Volt<sub>DC</sub>. Karena output yang dihasilkan dari penyearahan dioda adalah sebagai berikut :

*Diketahui:*

$$V_{eff} = 12\text{Volt}_{AC}$$

$$V_{max} = V_{DC} = \sqrt{2} \times 12 = 1,414 \times 12 = 17,28\text{V}_{DC}$$

### 3.1.2.6 Perancangan Pulse Width Modulation (PWM)

Diketahui spesifikasi Motor DC yang digunakan sebagai berikut ;

$$V_{in} = 12\text{Volt}_{DC}$$

$$\text{Speed}_{ref} = 2.400\text{rpm}$$

$$R = 38,709\Omega$$

Jika ingin didapatkan kecepatan putaran Motor DC sebesar : 2.000 rpm, 2.100 rpm, 2.200 rpm, 2.300 rpm dan 2.400 rpm. Maka dibutuhkan tegangan sumber ( $V_x$ ) yang diperlukan oleh Motor DC adalah sebagai berikut :

a. Speed = 2.000 rpm;  $V_x = \dots ?$

$$\frac{V_{in}}{\text{Speed}_{ref}} = \frac{V_x}{\text{speed}_x}$$

$$\frac{12}{2.400} = \frac{V_x}{2.000}$$

$$V_x = \frac{12 \times 2.000}{2.400}$$

$$V_x = \frac{24.000}{2.400}$$

$$V_x = 10\text{Volt}$$

$$I = \frac{V}{R}$$

$$I = \frac{10\text{Volt}}{38,709\Omega}$$

$$I = 0,258A$$

$$I = 258mA$$

$$P = I^2 \times R$$

$$P = (0,258)^2 \times 38,709$$

$$P = 2,576\text{Watt}$$

b. Speed = 2.100 rpm;  $V_x = \dots ?$

$$\frac{V_{in}}{\text{Speed}_{ref}} = \frac{V_x}{\text{speed}_x}$$

$$\frac{12}{2.400} = \frac{V_x}{2.100}$$

$$V_x = \frac{12 \times 2.100}{2.400}$$

$$V_x = \frac{25.200}{2.400}$$

$$V_x = 10,5\text{Volt}$$

$$I = \frac{V}{R}$$

$$I = \frac{10,5\text{Volt}}{38,709\Omega}$$

$$I = 0,271A$$

$$I = 271mA$$

$$P = I^2 \times R$$

$$P = (0,271)^2 \times 38,709$$

$$P = 2,842\text{Watt}$$

c. Speed = 2.200 rpm;  $V_x = \dots$ ?

$$\frac{V_{in}}{Speed_{ref}} = \frac{V_x}{speed_x}$$

$$\frac{12}{2.400} = \frac{V_x}{2.200}$$

$$V_x = \frac{12 \times 2.200}{2.400}$$

$$V_x = \frac{26.400}{2.400}$$

$$V_x = 11 \text{ Volt}$$

$$I = \frac{V}{R}$$

$$I = \frac{11 \text{ Volt}}{38,709 \Omega}$$

$$I = 0,284 A$$

$$I = 284 \text{ mA}$$

$$P = I^2 \times R$$

$$P = (0,284)^2 \times 38,709$$

$$P = 3,122 \text{ Watt}$$

d. Speed = 2.300 rpm;  $V_x = \dots$ ?

$$\frac{V_{in}}{Speed_{ref}} = \frac{V_x}{speed_x}$$

$$\frac{12}{2.400} = \frac{V_x}{2.300}$$

$$V_x = \frac{12 \times 2.300}{2.400}$$

$$V_x = \frac{27.600}{2.400}$$

$$V_x = 11,5 \text{ Volt}$$

$$I = \frac{V}{R}$$

$$I = \frac{11,5 \text{ Volt}}{38,709 \Omega}$$

$$I = 0,297 A$$

$$I = 297 \text{ mA}$$

$$P = I^2 \times R$$

$$P = (0,297)^2 \times 38,709$$

$$P = 3,414 \text{ Watt}$$

e. Speed = 2.400 rpm,  $V_x = \dots?$

$$\frac{V_{in}}{\text{Speed}_{ref}} = \frac{V_x}{\text{speed}_x}$$

$$\frac{12}{2.400} = \frac{V_x}{2.400}$$

$$V_x = \frac{12 \times 2.400}{2.400}$$

$$V_x = \frac{28.800}{2.400}$$

$$V_x = 12 \text{ Volt}$$

$$I = \frac{V}{R}$$

$$I = \frac{12 \text{ Volt}}{38,709 \Omega}$$

$$I = 0,310 \text{ A}$$

$$I = 310 \text{ mA}$$

$$P = I^2 \times R$$

$$P = (0,310)^2 \times 38,709$$

$$P = 3,719 \text{ Watt}$$

Tabel 3.4. Hasil Perhitungan Tegangan, Arus dan Daya Pada Motor DC

No.	Speed (rpm)	$V_x$ (Volt)	I (mA)	P (Watt)
1.	2.000	10	258	2,576
2.	2.100	10,5	271	2,842
3.	2.200	11	284	3,122
4.	2.300	11,5	297	3,414
5.	2.400	12	310	3,719

Seperti telah dijelaskan pada bab sebelumnya bahwa  $V = a.V_m$

Jika diketahui :

$$T_{2.000rpm} = \frac{60s}{2.000rpm}$$

$$T_{2.000rpm} = 0,030s = 30ms$$

$$T_{2.100rpm} = \frac{60s}{2.100rpm}$$

$$T_{2.100rpm} = 0,029s = 29ms$$

$$T_{2.200rpm} = \frac{60s}{2.200rpm}$$

$$T_{2.200rpm} = 0,027s = 27ms$$

$$T_{2.300rpm} = \frac{60s}{2.300rpm}$$

$$T_{2.300rpm} = 0,026s = 26ms$$

$$T_{2.400rpm} = \frac{60s}{2.400rpm}$$

$$T_{2.400rpm} = 0,025s = 25ms$$

Berikut adalah perhitungan untuk mendapatkan nilai dari  $T_2$  (kondisi high) dari waktu tunda (delay), sebagai berikut :

Rumus perhitungan :  $V = V_x$

$$a = \frac{T_1}{T_1 + T_2}$$

Untuk  $V_x = 10$  Volt dan  $T_1 = 25$  ms (posisi “low”), maka :

$$V_x = \frac{T_1}{T_1 + T_2} \times V_m$$

$$10 = \frac{(25 \times 10^{-3})}{(25 \times 10^{-3}) + T_2} \times 12$$

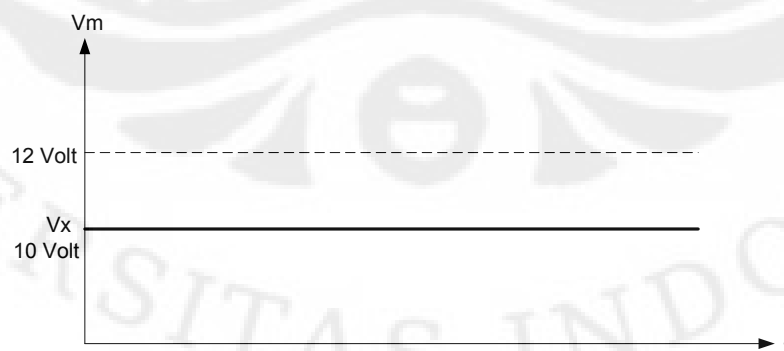
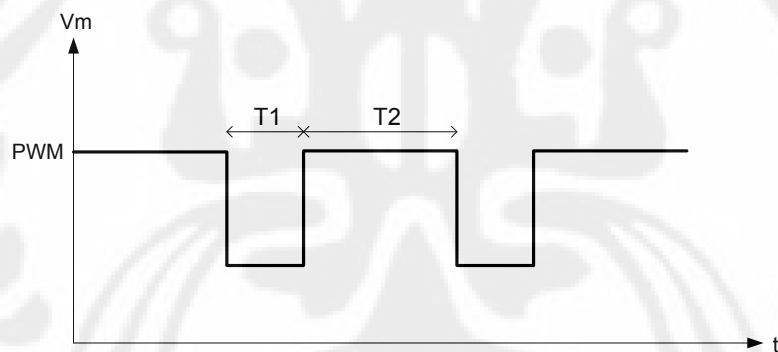
$$(250 \times 10^{-3}) + 10T_2 = 300 \times 10^{-3}$$

$$10T_2 = (300 \times 10^{-3}) - (250 \times 10^{-3})$$

$$T_2 = \frac{50}{10}$$

$$T_2 = 5 \text{ ms}$$

Jadi  $T_2 = 5 \text{ ms}$  (posisi "high")



Untuk  $V_x = 10,5 \text{ Volt}$  dan  $T_1 = 25 \text{ ms}$  (posisi "low"), maka :

$$V_x = \frac{T_1}{T_1 + T_2} \times V_m$$

$$10,5 = \frac{(25 \times 10^{-3})}{(25 \times 10^{-3}) + T_2} \times 12$$

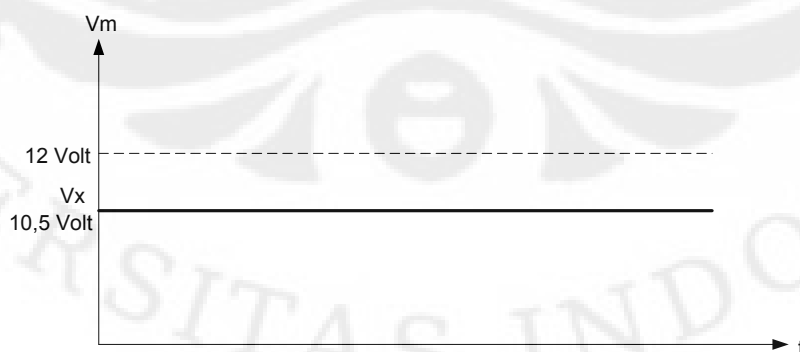
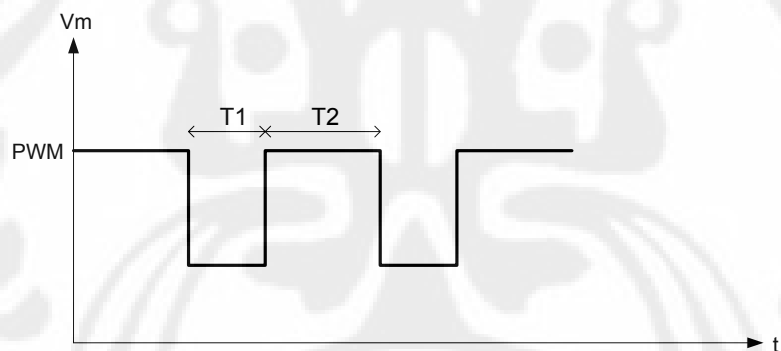
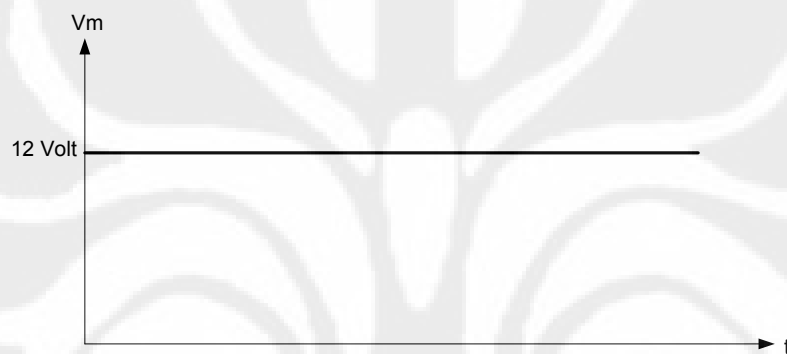
$$(263 \times 10^{-3}) + 10,5 T_2 = 300 \times 10^{-3}$$

$$10,5 T_2 = (300 \times 10^{-3}) - (263 \times 10^{-3})$$

$$T_2 = \frac{37}{10,5}$$

$$T_2 = 4 \text{ ms}$$

Jadi  $T_2 = 4 \text{ ms}$  (posisi "high")



Untuk  $V_x = 11 \text{ Volt}$  dan  $T_1 = 25 \text{ ms}$  (posisi "low"), maka :

$$V_x = \frac{T_1}{T_1 + T_2} \times V_m$$

$$11 = \frac{(25 \times 10^{-3})}{(25 \times 10^{-3}) + T_2} \times 12$$

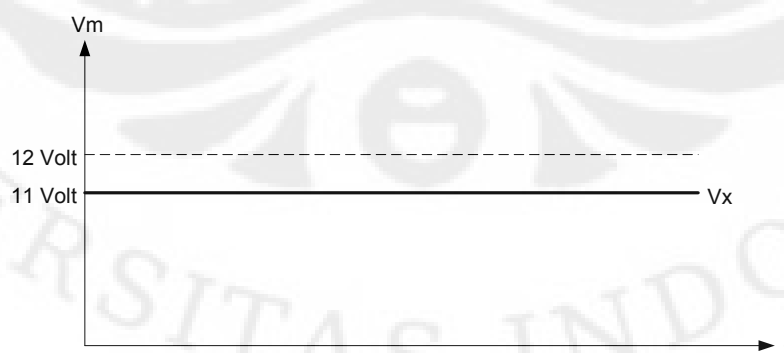
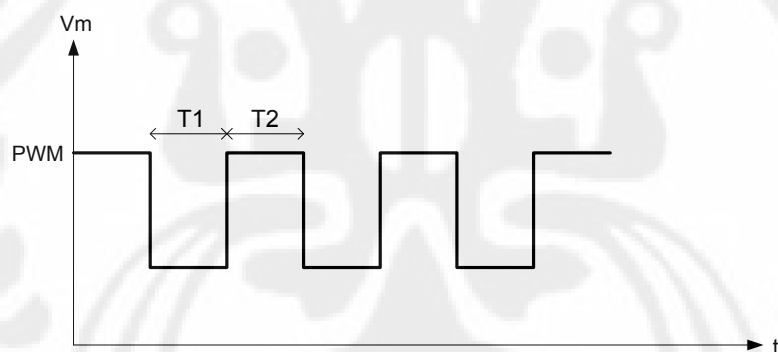
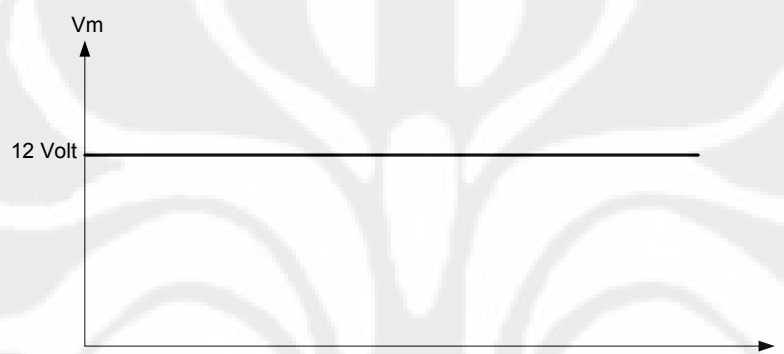
$$(275 \times 10^{-3}) + 11T_2 = 300 \times 10^{-3}$$

$$11T_2 = (300 \times 10^{-3}) - (275 \times 10^{-3})$$

$$T_2 = \frac{25}{11}$$

$$T_2 = 2 \text{ ms}$$

Jadi  $T_2 = 2 \text{ ms}$  (posisi "high")



Untuk  $V_x = 11,5 \text{ Volt}$  dan  $T_1 = 25 \text{ ms}$  (posisi "low"), maka :



$$V_x = \frac{T_1}{T_1 + T_2} \times V_m$$

$$11,5 = \frac{(25 \times 10^{-3})}{(25 \times 10^{-3}) + T_2} \times 12$$

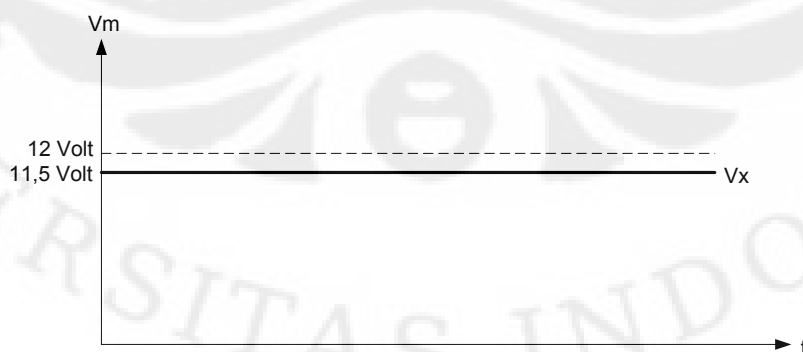
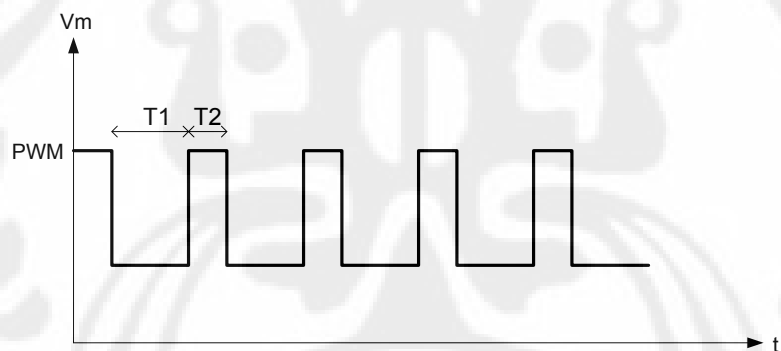
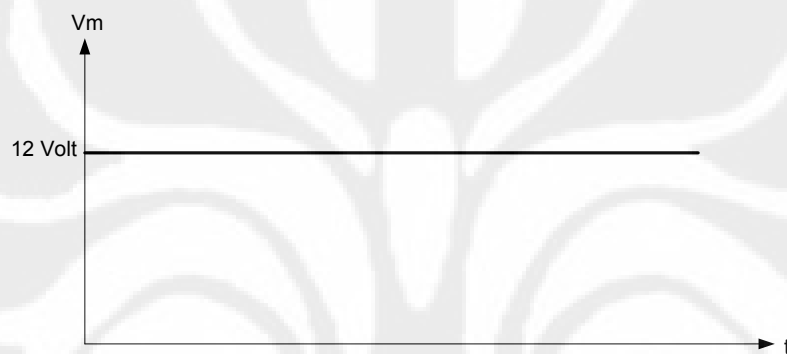
$$(288 \times 10^{-3}) + 11,5 T_2 = 300 \times 10^{-3}$$

$$11,5 T_2 = (300 \times 10^{-3}) - (288 \times 10^{-3})$$

$$T_2 = \frac{12}{11,5}$$

$$T_2 = 1 \text{ ms}$$

Jadi  $T_2 = 1 \text{ ms}$  (posisi "high")



Dari perhitungan diatas dapat dibuat suatu tabel data sebagai berikut :

Tabel 3.5. Pembangkitan Sinyal PWM

Speed (rpm)	T (s)	T (ms)	T <sub>1</sub> low (ms)	T <sub>2</sub> high (ms)
2.000	0,030	30	25	5
2.100	0,029	29	25	4
2.200	0,027	27	25	2
2.300	0,026	26	25	1
2.400	0,025	25	25	0

### 3.1.2.7. Perancangan Pemanas Buatan

Perancangan pemanas buatan ini dibuat dengan menggunakan Lampu Pijar menggunakan tegangan 220 Volt AC, dimana lampu diletakkan berdekatan dengan sensor Temperatur untuk mendapatkan perubahan temperature sesuai dengan yang diharapkan. Direncanakan Lampu tersebut akan dinyalakan saat alat belum diaktifkan atau dalam kondisi off. Hal ini dilakukan untuk mendapatkan perubahan temperatur yang signifikan dan cukup cepat kenaikannya.

### 3.2 Perancangan Output Sensor Temperatur dan Input ADC

Perancangan output sensor temperature dan input ADC disini, yaitu besar tegangan output yang dihasilkan oleh sensor temperature LM35 yang akan diumpankan ke input dari ADC0804. Hal ini diperlukan untuk mengetahui nilai dari temperature ruangan, sehingga dapat dibuatkan logika pemrogramannya untuk menghasilkan kecepatan putaran Motor DC yang diharapkan serta tampilan pada LCD. Perencanaan perubahan nilai temperature terhadap perubahan logika output ADC dalam bentuk deret bilangan biner dapat dilihat pada tabel dibawah ini.

Tabel 3.6. Data Gabungan In-Out Sensor Temperatur & ADC

No.	Temp. (°C)	Vout Sensor (mV)	Vin ADC (mV)	Desimal	Biner							
					D7	D6	D5	D4	D3	D2	D1	D0
1	16	160	160	08	0	0	0	0	1	0	0	0
2	17	170	✕									
3	18	180	180	09	0	0	0	0	1	0	0	1
4	19	190	✕									
5	20	200	200	0A	0	0	0	0	1	0	1	0
6	21	210	✕									

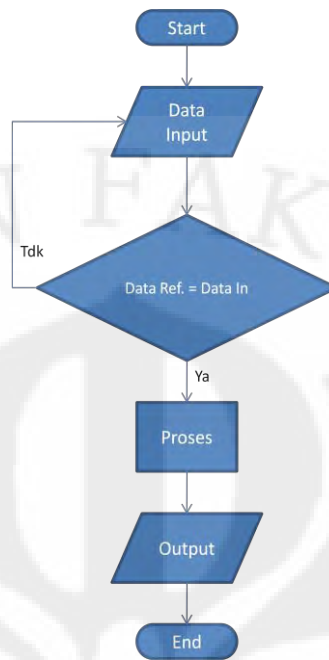
7	22	220	220	0B	0	0	0	0	1	0	1	1
8	23	230	✖									
9	24	240	240	0C	0	0	0	0	1	1	0	0
10	25	250	✖									
11	26	260	260	0D	0	0	0	0	1	1	0	1
12	27	270	✖									
13	28	280	280	0E	0	0	0	0	1	1	1	0
14	29	290	✖									
15	30	300	300	0F	0	0	0	0	1	1	1	1
16	31	310	✖									
17	32	320	320	10	0	0	0	1	0	0	0	0
18	33	330	✖									
19	34	340	340	11	0	0	0	1	0	0	0	1
20	35	350	✖									
21	36	360	360	12	0	0	0	1	0	0	1	0
22	37	370	✖									
23	38	380	380	13	0	0	0	1	0	0	1	1
24	39	390	✖									
25	40	400	400	14	0	0	0	1	0	1	0	0
26	41	410	✖									
27	42	420	420	15	0	0	0	1	0	1	0	1
28	43	430	✖									
29	44	440	440	16	0	0	0	1	0	1	1	0
30	45	450	✖									

### 3.3 Perancangan Software

#### 3.3.1 Perancangan Downloader Mikrokontroler AT89S51

Perancangan software disini yaitu pembuatan program yang akan dimasukkan kedalam sebuah chip mikrokontroler AT89S51 untuk mengendalikan kerja dari sistem secara keseluruhan.

Pemrograman dibuat dengan menggunakan bahasa pemrograman assembler, dengan algoritma sebagai berikut :



Gambar 3.13. Diagram Alir Algoritma Perancangan Program

Berikut adalah prosedur pengisian program dengan menggunakan downloader AEC\_ISP :

1. Menjalankan program AEC\_ISP, sehingga dilayar akan muncul tampilan sebagai berikut :



2. Pilih menu Setup.



3. Pilih Seri IC mikrokontroler yang akan digunakan (dalam hal AT89S51).



4. Pilih jenis port yang akan digunakan.

5. Pilih menu Load Hex File to Flash Buffer untuk membuka file.hex yang akan di download ke mikrokontroler. Kemudian ketikkan nama file.hex tersebut.

Perancangan program dalam sistem ini dibagi menjadi beberapa bagian, sebagai berikut :

1. Program untuk tampilan LCD.
2. Program untuk Sensor Temperatur.
3. Program untuk Sensor Kecepatan Putaran Motor DC.
4. Program untuk membangkitkan sinyal PWM.

## **BAB 4**

### **HASIL PENGUKURAN DAN ANALISIS**

#### **4. Pengukuran Dan Analisis**

Pengujian dan pengukuran alat dalam perancangan ini dilakukan untuk mendapatkan data yang dapat dibandingkan dengan data hasil perhitungan. Hal ini diperlukan untuk mengetahui tingkat error dari alat yang dirancang. Pengukuran dan analisis dilakukan dalam 4 bagian, yaitu pengukuran ADC, pengukuran sensor temperatur LM35, pengukuran sinyal PWM dan pengukuran alat secara keseluruhan atau pengujian alat.

Untuk mendapatkan perubahan nilai temperatur sebagai keperluan pengambilan data dimaksud, dibuat suatu simulasi ruangan dengan menggunakan acrylic, seperti telah dijelaskan pada Bab 3 dari Skripsi ini.

##### **4.1. Pengukuran Dan Analisis ADC**

Dalam melakukan pengukuran ADC digunakan beberapa peralatan sebagai berikut :

- a. Lampu Pijar 220 Volt AC / 10 Watt (sebagai sumber panas buatan).
- b. Multimeter Digital.
- c. Alat ukur temperatur (Thermometer Digital).

Pengukuran ADC dilakukan untuk mengetahui unjuk kerja dari ADC, dimana pengukuran dilakukan pada tegangan input ADC dengan memperhatikan juga logika output yang berupa deretan bilangan biner yang diperlukan oleh kontroler untuk mendapatkan suatu perintah output yang tepat. Pengukuran ADC dalam perancangan ini dilakukan dengan 2 cara, yaitu :

1. Pengukuran ADC dengan sinyal input menggunakan sensor temperatur LM35. Perlu dijelaskan bahwa pengukuran dilakukan pada waktu pagi hari, siang hari dan malam hari. Sehingga temperatur atau tegangan input yang diperlu tidak keseluruhan dari nilai sinyal atau tegangan input ADC yang direncanakan. Adapun hasil pengukuran ini dapat dilihat pada tabel dibawah ini.

Tabel 4.1. Pengukuran Input / Output ADC Dengan Menggunakan Sensor Temperatur LM35

No	Temp <sup>Pengukuran</sup> (°C)	Vout <sup>Pengukuran</sup> LM35 (mV)	D7	D6	D5	D4	D3	D2	D1	D0
1	26.9	270	0	0	0	0	1	1	0	1
2	28.2	281	0	0	0	0	1	1	0	1
3	30.0	301	0	0	0	0	1	1	1	1
4	32.1	323	0	0	0	1	0	0	0	1

- Pengukuran ADC dengan menggunakan trimmer potensiometer (trimpot), yaitu pengukuran tegangan input ADC, dimana tegangan input diperoleh dengan mengatur nilai tahanan dari trimmer potensiometer (trimpot) untuk mendapatkan nilai tegangan input yang diinginkan. Adapun hasil perhitungan dan pengukuran tersebut dapat dilihat pada tabel dibawah ini.

Tabel 4.1. Hasil Perhitungan Input dan Output ADC0804

No.	Vin ADC (mV)	D7	D6	D5	D4	D3	D2	D1	D0	Hex
1	160	0	0	0	0	1	0	0	0	08
2	180	0	0	0	0	1	0	0	1	09
3	200	0	0	0	0	1	0	1	0	0A
4	220	0	0	0	0	1	0	1	1	0B
5	240	0	0	0	0	1	1	0	0	0C
6	260	0	0	0	0	1	1	0	1	0D
7	280	0	0	0	0	1	1	1	0	0E
8	300	0	0	0	0	1	1	1	1	0F
9	320	0	0	0	1	0	0	0	0	10
10	340	0	0	0	1	0	0	0	1	11
11	360	0	0	0	1	0	0	1	0	12
12	380	0	0	0	1	0	0	1	1	13
13	400	0	0	0	1	0	1	0	0	14
14	420	0	0	0	1	0	1	0	1	15
15	440	0	0	0	1	0	1	1	0	16

Tabel 4.2. Hasil Pengukuran Input dan Output ADC0804 Dengan Menggunakan Trimpot Sebagai Pengatur Besaran Nilai Tegangan Input ADC

No.	Vout Trimpot (mV)	Vin ADC (mV)	D7	D6	D5	D4	D3	D2	D1	D0	Hex
1	162	162	0	0	0	0	0	1	1	1	07
2	181	181	0	0	0	0	1	0	0	1	09

3	202	202	0	0	0	0	1	0	0	1	09
4	223	223	0	0	0	0	1	0	1	1	0B
5	241	241	0	0	0	0	1	0	1	1	0B
6	260	260	0	0	0	0	1	1	0	1	0D
7	283	283	0	0	0	0	1	1	0	1	0D
8	301	301	0	0	0	0	1	1	1	1	0F
9	322	322	0	0	0	0	1	1	1	1	0F
10	340	340	0	0	0	1	0	0	0	1	11
11	362	362	0	0	0	1	0	0	0	1	11
12	383	383	0	0	0	1	0	0	1	1	13
13	400	400	0	0	0	1	0	0	1	1	13
14	421	421	0	0	0	1	0	1	0	1	15
15	442	442	0	0	0	1	0	1	0	1	15

Tabel 4.3. Perbandingan Data Perhitungan dan Pengukuran Tegangan Input ADC

No.	Temp. (°C)	Vin ADC (mV)	Hex	Vout Trimpot (mV)	Hex	Error (%)
1	16	160	08	162	07	1%
2	18	180	09	181	09	1%
3	20	200	0A	202	09	1%
4	22	220	0B	223	0B	1%
5	24	240	0C	241	0B	0%
6	26	260	0D	260	0D	0%
7	28	280	0E	283	0D	1%
8	30	300	0F	301	0F	0%
9	32	320	10	322	0F	1%
10	34	340	11	340	11	0%
11	36	360	12	362	11	1%
12	38	380	13	383	13	1%
13	40	400	14	400	13	0%
14	42	420	15	421	15	0%
15	44	440	16	442	15	0%

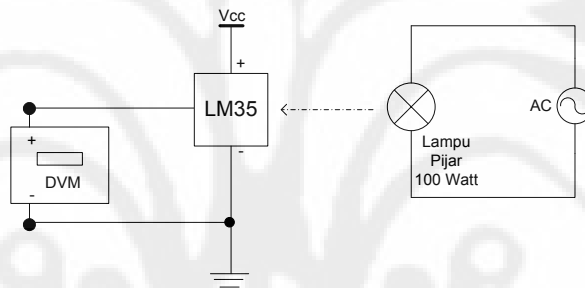


Dari hasil data perhitungan dan pengukuran input dan output ADC diatas, dapat dilihat bahwa terdapat perbedaan logika output yang dihasilkan oleh ADC0804, baik pengukuran dengan menggunakan trimer potensiometer ataupun dengan menggunakan sensor temperature LM35 sebagai input ADC-nya.

#### 4.2. Pengukuran Dan Analisis Sensor Temperatur LM35

Pengukuran dan analisis ini dilakukan untuk mengetahui unjuk kerja dari sensor temperature LM35, serta untuk mendapatkan kondisi real yang dihasilkan oleh sensor sehingga dapat diketahui prosentase error pembacaan sensor temperature dalam perancangan alat ini.

Adapun metode atau teknik pengukuran sensor temperatur yang dilakukan adalah sebagai berikut :



Gambar 4.1. Metode pengukuran sensor temperature LM35

Tabel 4.4. Hasil Pengukuran Sensor Temperatur LM35

No	Temp <sub>Perhitungan</sub> (°C)	Vout <sub>LM35Perhitungan</sub> (mV)	Temp <sub>Pengukuran</sub> (°C)	Vout <sub>LM35Pengukuran</sub> (mV)	Error (%)
1	26	260	26.9	270	3%
2	28	280	28.2	281	1%
3	30	300	30.0	301	0%
4	32	320	32.1	323	0%

Hasil pengukuran diatas sama dengan hasil pengukuran tegangan input ADC dengan menggunakan sensor temperatur LM35. Dimana pengukuran dilakukan dengan menggunakan kondisi temperature real dari lingkungan, yaitu pagi, siang dan malam hari. Sehingga nilai temperature yang didapat terbatas dan tidak memenuhi semua nilai temperature yang direncanakan. Hasil pengukuran sensor temperature dengan menggunakan pemanas buatan pada ruang simulasi adalah sebagai berikut :

Tabel 4.5. Hasil Pengukuran Sensor Temperatur LM35 Menggunakan Ruang Simulasi

No.	Vin ADC ( mV )	Biner							
		D7	D6	D5	D4	D3	D2	D1	D0
1	270	0	0	0	0	1	1	1	1
2	310	0	0	0	1	0	0	0	1
3	330	0	0	0	1	0	0	1	1
4	350	0	0	0	1	0	1	1	1
5	370	0	0	0	1	0	1	0	1

Pengukuran input sensor temperatur LM35 dilakukan dengan menggunakan Thermometer Digital, sedangkan pengukuran tegangan output sensor temperatur LM35 dengan menggunakan Multimeter Digital. Data hasil perhitungan dan pengukuran tersebut terdapat perbedaan yang tidak terlalu signifikan.

### 4.3. Pengukuran dan Analisis Sinyal PWM

Berikut adalah tabel hasil pengukuran tegangan input Motor DC setelah dilakukan proses modulasi lebar pulsa pada sinyal control rangkaian driver Motor DC.

Tabel 4.6. Tegangan Input Motor DC Setelah Pemrosesan Sinyal PWM

No.	Temp. (oC)	Speed (rpm)	Vin (Volt) Teoritis	Daya (Watt) Teoritis	Vx (Volt) Pengukuran
1.	Off	-	12	3,719	11,11
2.	39 - 44	2.400	12	3,719	10,71
3.	34 – 38	2.300	11,5	3,414	10,53
4.	27 – 33	2.200	11	3,122	10,22
5.	21 – 26	2.100	10,5	2,842	9,98
6.	16 – 20	2.000	10	2,576	-

Tabel 4.7. Perhitungan Kecepatan Putaran Motor DC

Kecepatan (rpm)	Vin (Volt)	T <sub>1</sub> “low” (ms)	T <sub>2</sub> “high” (ms)
2.000	10	25	5

2.100	10,5	25	4
2.200	11	25	2
2.300	11,5	25	1
2.400	12	25	0

Tabel perhitungan diatas diambil dari perhitungan sebelumnya, pada perancangan sinyal PWM dalam Bab 3. Bila diamati pada tabel diatas bahwa semakin besar nilai dari  $T_2$ , maka putaran Motor DC akan semakin lambat dan sebaliknya bila nilai dari  $T_2$  semakin kecil, maka putaran Motor DC akan semakin cepat.

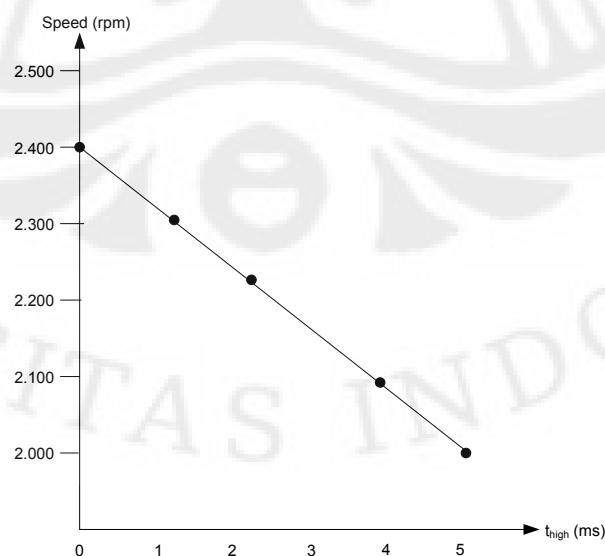
$$V_x = V_m \cdot \frac{T_1}{T_1 + T_2}$$

$$V_x \cdot T_1 + V_x \cdot T_2 = V_m \cdot T_1$$

$$V_x \cdot T_2 = V_m \cdot T_1 - V_x \cdot T_1$$

$$V_x = \frac{V_m \cdot T_1 - V_x \cdot T_1}{T_2}$$

Sehingga untuk memperlambat putaran Motor DC, periode waktu “kondisi off” dari sinyal PWM semakin diperlebar, karena nilai  $V_x$  berbanding terbalik dengan nilai  $T_2$ . Hal ini dapat dilihat pada tabel 4.3 diatas, dimana nilai dari  $T_2$  mempengaruhi besar tegangan  $V_x$ . Secara grafik dapat digambarkan sebagai berikut.



Gambar 4.2. Kurva Sinyal PWM VS Kecepatan Putaran Motor DC

Dari data diatas dapat dilihat bahwa terdapat perbedaan tegangan input (sumber) untuk Motor DC hasil perhitungan dengan hasil pengukuran sebesar  $\pm 1$  Volt. Hal ini dapat dipengaruhi oleh komponen-komponen yang digunakan dalam rangkaian catu daya dan dapat juga dipengaruhi oleh banyaknya beban yang membutuhkan tegangan supply dari rangkaian catu daya. Karena pada perancangan ini tegangan input Motor DC disupply dari sumber tegangan yang sama dengan rangkaian yang lain, dimana Motor DC membutuhkan energi listrik yang cukup besar untuk beroperasi.

Hal lain yang dapat dijelaskan dari data-data diatas adalah, bahwa dengan menggunakan sistem modulasi lebar pulsa (PWM), maka energi listrik yang disalurkan sesuai dengan kebutuhan.

#### 4.4. Pengukuran / Pengujian Alat

Tabel 4.8. Pengukuran / Pengujian Alat

No	Pengujian		Perancangan		% Error Temp.	% Error Speed Motor	% Error Rata2
	Temp. (°C)	Speed Motor (rpm)	Temp. (°C)	Speed Motor (rpm)			
1	26.9	2,200	26	2,100	3.35%	4.76%	4.05%
2	27.9	2,200	28	2,200	0.36%	0.00%	0.18%
3	30.0	2,200	30	2,200	0.00%	0.00%	0.00%
4	31.0	2,200	30	2,200	3.23%	0.00%	1.61%
5	30.7	2,200	30	2,200	2.28%	0.00%	1.14%
6	30.2	2,200	30	2,200	0.66%	0.00%	0.33%
7	32.1	2,300	32	2,300	0.31%	0.00%	0.16%

## **BAB 5 KESIMPULAN**

Dari hasil eksperimen yang telah dilakukan dapat ditarik beberapa kesimpulan sebagai berikut:

1. Pengendalian kecepatan putaran motor DC dengan menggunakan Pulse Width Modulation (PWM) memiliki keuntungan dan kerugian sebagai berikut :
  - Keuntungannya adalah dengan menggunakan PWM, perubahan tegangan hasil pemrosesan sinyal PWM dapat dibuat lebih teliti dan keuntungan lain adalah rangkaian yang dibuat menjadi lebih sederhana dan dapat memproteksi rangkaian kontrol dan motor itu sendiri karena proses pengaturan “on” dan “off” disini adalah pada rangkaian pengendali motornya dan bukan pada motornya.
  - Kerugiannya adalah perlu suatu rangkaian control yang memerlukan pembuatan program. Meskipun pada beberapa aplikasi, pembentukan sinyal PWM dapat dibuat dengan menggunakan rangkaian digital dan tentunya sinyal PWM yang dihasilkan menjadi terbatas.
2. Dengan adanya sistem otomatis ini dapat meningkatkan efektifitas dan efisiensi kerja dari suatu proses, seperti contoh yang telah dijelaskan pada awal Skripsi ini.
3. Dengan menggunakan PWM dalam pengendalian kecepatan putaran motor, arus dan tegangan yang diperlukan oleh motor dapat dikendalikan besarnya, sehingga daya listrik akan berubah besarnya. Hal ini dapat meningkatkan efisiensi daya listrik tersebut, karena pada beberapa kondisi atau pekerjaan sebuah Motor DC tidak diperlukan untuk bekerja atau berputar dengan kecepatan maksimal dan pada pada kondisi yang lain diperlukan putaran motor dengan kecepatan maksimal.
4. Dari hasil perhitungan dan pengukuran output logika biner ADC0804 terdapat perbedaan. Hal ini dapat dipengaruhi oleh komponen ADC yang digunakan, mengingat untuk ADC dapat bekerja atau mengkonversi sinyal analog menjadi sinyal digital diperlukan inputan-inputan tertentu

yang dibutuhkan oleh ADC.

5. Perhitungan dan pengukuran sensor temperature LM35 terdapat perbedaan yang tidak terlalu besar. Hal ini dapat disebabkan adanya perbedaan karakteristik dari sensor temperature yang digunakan pada alat ukur temperatur (Thermometer Digital), dan dapat dipengaruhi oleh akurasi dan karakteristik dari alat ukur tegangan (Multimeter Digital). Sehingga dapat disimpulkan untuk melakukan pengukuran perlu memperhatikan akurasi serta karakteristik dari alat ukur yang digunakan, serta kondisi dan karakteristik dari peralatan atau objek yang akan diukur.
6. Tegangan input Motor DC hasil pemroses sinyal PWM terdapat perbedaan sebesar  $\pm 1$  Volt dengan tegangan input hasil perhitungan. Seperti telah dijelaskan pada bab 4 diatas, bahwa perbedaan data tersebut dapat disebabkan oleh banyaknya beban yang perlu disupply oleh rangkaian catu daya dan Motor DC yang memerlukan energy yang cukup besar untuk beroperasi. Hal ini tentunya dapat mempengaruhi unjuk kerja dari sistem atau rangkaian secara keseluruhan. Karena dimungkinkan terdapat variabel-variabel input, tegangan atau arus listrik yang dibutuhkan oleh komponen tidak terpenuhi sesuai dengan peruntukannya. Sehingga dalam hal ini perlu diperhatikan dalam pembuatan catu daya, yaitu besar daya dan tegangan yang dihasilkan serta daya dan tegangan yang diperlukan oleh peralatan atau sistem untuk beroperasi. Hal lain yang dapat disimpulkan adalah perlu dilakukan pemisahan tegangan sumber (catu daya) untuk Motor DC atau komponen yang memerlukan energy yang cukup besar dengan rangkaian lainnya, terutama dengan rangkaian kontrol. Karena hal ini dapat mempengaruhi kerja dari rangkaian kontrol.

## DAFTAR REFERENSI

*“Pedoman Efisiensi Energi untuk Industri di Asia ”,*

*[http:// www.energyefficiencyasia.org/docs/ee\\_modules/indo/Chapter%20-%20Electric%20motors%20\(Bahasa%20Indonesia\).pdf](http://www.energyefficiencyasia.org/docs/ee_modules/indo/Chapter%20-%20Electric%20motors%20(Bahasa%20Indonesia).pdf), November 2009*

Rahardjo, Pratolo, 2 Juli – Desember 2004, *“Aplikasi Mikrokontroler 80C31 Sebagai Alat Pengendali Motor DC Magnet Permanen Dengan Metode PID Dan Metode PWM”*, November 2009.

Zulkarnain, Mohamad, Maret 2009, *“Parkir Mobil Otomatis Pada Ruang Bertingkat Berbasis Mikrokontroler AT89S51”*, November 2009.

Nalwan, P. Andi, *Teknik Antar Muka dan Pemrograman Mikrokontroler AT89C51*, PT. ELEX MEDIA KOMPUTINDO, Jakarta, 2003.

*Modul Pelatihan MCS – 51*, Universitas Budi Luhur, Jakarta, 2005.

Lampiran 1. Tabel Tegangan Input ADC

No.	Vin ADC ( mV )	Desimal	Biner							
			D7	D6	D5	D4	D3	D2	D1	D0
1	0	00	0	0	0	0	0	0	0	0
2	20	01	0	0	0	0	0	0	0	1
3	40	02	0	0	0	0	0	0	1	0
4	60	03	0	0	0	0	0	0	1	1
5	80	04	0	0	0	0	0	1	0	0
6	100	05	0	0	0	0	0	1	0	1
7	120	06	0	0	0	0	0	1	1	0
8	140	07	0	0	0	0	0	1	1	1
9	160	08	0	0	0	0	1	0	0	0
10	180	09	0	0	0	0	1	0	0	1
11	200	0A	0	0	0	0	1	0	1	0
12	220	0B	0	0	0	0	1	0	1	1
13	240	0C	0	0	0	0	1	1	0	0
14	260	0D	0	0	0	0	1	1	0	1
15	280	0E	0	0	0	0	1	1	1	0
16	300	0F	0	0	0	0	1	1	1	1
17	320	10	0	0	0	1	0	0	0	0
18	340	11	0	0	0	1	0	0	0	1
19	360	12	0	0	0	1	0	0	1	0
20	380	13	0	0	0	1	0	0	1	1
21	400	14	0	0	0	1	0	1	0	0
22	420	15	0	0	0	1	0	1	0	1
23	440	16	0	0	0	1	0	1	1	0
24	460	17	0	0	0	1	0	1	1	1
25	480	18	0	0	0	1	1	0	0	0
26	500	19	0	0	0	1	1	0	0	1
27	520	1A	0	0	0	1	1	0	1	0
28	540	1B	0	0	0	1	1	0	1	1
29	560	1C	0	0	0	1	1	1	0	0
30	580	1D	0	0	0	1	1	1	0	1
31	600	1E	0	0	0	1	1	1	1	0
32	620	1F	0	0	0	1	1	1	1	1
33	640	20	0	0	1	0	0	0	0	0
34	660	21	0	0	1	0	0	0	0	1
35	680	22	0	0	1	0	0	0	1	0



Lampiran 1. Tabel Tegangan Input ADC  
(Lanjutan)

36	700	23	0	0	1	0	0	0	1	1
37	720	24	0	0	1	0	0	1	0	0
38	740	25	0	0	1	0	0	1	0	1
39	760	26	0	0	1	0	0	1	1	0
40	780	27	0	0	1	0	0	1	1	1
41	800	28	0	0	1	0	1	0	0	0
42	820	29	0	0	1	0	1	0	0	1
43	840	2A	0	0	1	0	1	0	1	0
44	860	2B	0	0	1	0	1	0	1	1
45	880	2C	0	0	1	0	1	1	0	0
46	900	2D	0	0	1	0	1	1	0	1
47	920	2E	0	0	1	0	1	1	1	0
48	940	2F	0	0	1	0	1	1	1	1
49	960	30	0	0	1	1	0	0	0	0
50	980	31	0	0	1	1	0	0	0	1
51	1000	32	0	0	1	1	0	0	1	0
52	1020	33	0	0	1	1	0	0	1	1
53	1040	34	0	0	1	1	0	1	0	0
54	1060	35	0	0	1	1	0	1	0	1
55	1080	36	0	0	1	1	0	1	1	0
56	1100	37	0	0	1	1	0	1	1	1
57	1120	38	0	0	1	1	1	0	0	0
58	1140	39	0	0	1	1	1	0	0	1
59	1160	3A	0	0	1	1	1	0	1	0
60	1180	3B	0	0	1	1	1	0	1	1
61	1200	3C	0	0	1	1	1	1	0	0
62	1220	3D	0	0	1	1	1	1	0	1
63	1240	3E	0	0	1	1	1	1	1	0
64	1260	3F	0	0	1	1	1	1	1	1
65	1280	40	0	1	0	0	0	0	0	0
66	1300	41	0	1	0	0	0	0	0	1
67	1320	42	0	1	0	0	0	0	1	0
68	1340	43	0	1	0	0	0	0	1	1
69	1360	44	0	1	0	0	0	1	0	0
70	1380	45	0	1	0	0	0	1	0	1
71	1400	46	0	1	0	0	0	1	1	0
72	1420	47	0	1	0	0	0	1	1	1
73	1440	48	0	1	0	0	1	0	0	0

Lampiran 1. Tabel Tegangan Input ADC  
(Lanjutan)

74	1460	49	0	1	0	0	1	0	0	1
75	1480	4A	0	1	0	0	1	0	1	0
76	1500	4B	0	1	0	0	1	0	1	1
77	1520	4C	0	1	0	0	1	1	0	0
78	1540	4D	0	1	0	0	1	1	0	1
79	1560	4E	0	1	0	0	1	1	1	0
80	1580	4F	0	1	0	0	1	1	1	1
81	1600	50	0	1	0	1	0	0	0	0
82	1620	51	0	1	0	1	0	0	0	1
83	1640	52	0	1	0	1	0	0	1	0
84	1660	53	0	1	0	1	0	0	1	1
85	1680	54	0	1	0	1	0	1	0	0
86	1700	55	0	1	0	1	0	1	0	1
87	1720	56	0	1	0	1	0	1	1	0
88	1740	57	0	1	0	1	0	1	1	1
89	1760	58	0	1	0	1	1	0	0	0
90	1780	59	0	1	0	1	1	0	0	1
91	1800	5A	0	1	0	1	1	0	1	0
92	1820	5B	0	1	0	1	1	0	1	1
93	1840	5C	0	1	0	1	1	1	0	0
94	1860	5D	0	1	0	1	1	1	0	1
95	1880	5E	0	1	0	1	1	1	1	0
96	1900	5F	0	1	0	1	1	1	1	1
97	1920	60	0	1	1	0	0	0	0	0
98	1940	61	0	1	1	0	0	0	0	1
99	1960	62	0	1	1	0	0	0	1	0
100	1980	63	0	1	1	0	0	0	1	1
101	2000	64	0	1	1	0	0	1	0	0
102	2020	65	0	1	1	0	0	1	0	1
103	2040	66	0	1	1	0	0	1	1	0
104	2060	67	0	1	1	0	0	1	1	1
105	2080	68	0	1	1	0	1	0	0	0
106	2100	69	0	1	1	0	1	0	0	1
107	2120	6A	0	1	1	0	1	0	1	0
108	2140	6B	0	1	1	0	1	0	1	1
109	2160	6C	0	1	1	0	1	1	0	0
110	2180	6D	0	1	1	0	1	1	0	1
111	2200	6E	0	1	1	0	1	1	1	0

Lampiran 1. Tabel Tegangan Input ADC  
(Lanjutan)

112	2220	6F	0	1	1	0	1	1	1	1
113	2240	70	0	1	1	1	0	0	0	0
114	2260	71	0	1	1	1	0	0	0	1
115	2280	72	0	1	1	1	0	0	1	0
116	2300	73	0	1	1	1	0	0	1	1
117	2320	74	0	1	1	1	0	1	0	0
118	2340	75	0	1	1	1	0	1	0	1
119	2360	76	0	1	1	1	0	1	1	0
120	2380	77	0	1	1	1	0	1	1	1
121	2400	78	0	1	1	1	1	0	0	0
122	2420	79	0	1	1	1	1	0	0	1
123	2440	7A	0	1	1	1	1	0	1	0
124	2460	7B	0	1	1	1	1	0	1	1
125	2480	7C	0	1	1	1	1	1	0	0
126	2500	7D	0	1	1	1	1	1	0	1
127	2520	7E	0	1	1	1	1	1	1	0
128	2540	7F	0	1	1	1	1	1	1	1
129	2560	80	1	0	0	0	0	0	0	0
130	2580	81	1	0	0	0	0	0	0	1
131	2600	82	1	0	0	0	0	0	1	0
132	2620	83	1	0	0	0	0	0	1	1
133	2640	84	1	0	0	0	0	1	0	0
134	2660	85	1	0	0	0	0	1	0	1
135	2680	86	1	0	0	0	0	1	1	0
136	2700	87	1	0	0	0	0	1	1	1
137	2720	88	1	0	0	0	1	0	0	0
138	2740	89	1	0	0	0	1	0	0	1
139	2760	8A	1	0	0	0	1	0	1	0
140	2780	8B	1	0	0	0	1	0	1	1
141	2800	8C	1	0	0	0	1	1	0	0
142	2820	8D	1	0	0	0	1	1	0	1
143	2840	8E	1	0	0	0	1	1	1	0
144	2860	8F	1	0	0	0	1	1	1	1
145	2880	90	1	0	0	1	0	0	0	0
146	2900	91	1	0	0	1	0	0	0	1
147	2920	92	1	0	0	1	0	0	1	0
148	2940	93	1	0	0	1	0	0	1	1
149	2960	94	1	0	0	1	0	1	0	0

Lampiran 1. Tabel Tegangan Input ADC  
(Lanjutan)

150	2980	95	1	0	0	1	0	1	0	1
151	3000	96	1	0	0	1	0	1	1	0
152	3020	97	1	0	0	1	0	1	1	1
153	3040	98	1	0	0	1	1	0	0	0
154	3060	99	1	0	0	1	1	0	0	1
155	3080	9A	1	0	0	1	1	0	1	0
156	3100	9B	1	0	0	1	1	0	1	1
157	3120	9C	1	0	0	1	1	1	0	0
158	3140	9D	1	0	0	1	1	1	0	1
159	3160	9E	1	0	0	1	1	1	1	0
160	3180	9F	1	0	0	1	1	1	1	1
161	3200	A0	1	0	1	0	0	0	0	0
162	3220	A1	1	0	1	0	0	0	0	1
163	3240	A2	1	0	1	0	0	0	1	0
164	3260	A3	1	0	1	0	0	0	1	1
165	3280	A4	1	0	1	0	0	1	0	0
166	3300	A5	1	0	1	0	0	1	0	1
167	3320	A6	1	0	1	0	0	1	1	0
168	3340	A7	1	0	1	0	0	1	1	1
169	3360	A8	1	0	1	0	1	0	0	0
170	3380	A9	1	0	1	0	1	0	0	1
171	3400	AA	1	0	1	0	1	0	1	0
172	3420	AB	1	0	1	0	1	0	1	1
173	3440	AC	1	0	1	0	1	1	0	0
174	3460	AD	1	0	1	0	1	1	0	1
175	3480	AE	1	0	1	0	1	1	1	0
176	3500	AF	1	0	1	0	1	1	1	1
177	3520	B0	1	0	1	1	0	0	0	0
178	3540	B1	1	0	1	1	0	0	0	1
179	3560	B2	1	0	1	1	0	0	1	0
180	3580	B3	1	0	1	1	0	0	1	1
181	3600	B4	1	0	1	1	0	1	0	0
182	3620	B5	1	0	1	1	0	1	0	1
183	3640	B6	1	0	1	1	0	1	1	0
184	3660	B7	1	0	1	1	0	1	1	1
185	3680	B8	1	0	1	1	1	0	0	0
186	3700	B9	1	0	1	1	1	0	0	1
187	3720	BA	1	0	1	1	1	0	1	0

Lampiran 1. Tabel Tegangan Input ADC  
(Lanjutan)

188	3740	BB	1	0	1	1	1	0	1	1
189	3760	BC	1	0	1	1	1	1	0	0
190	3780	BD	1	0	1	1	1	1	0	1
191	3800	BE	1	0	1	1	1	1	1	0
192	3820	BF	1	0	1	1	1	1	1	1
193	3840	C0	1	1	0	0	0	0	0	0
194	3860	C1	1	1	0	0	0	0	0	1
195	3880	C2	1	1	0	0	0	0	1	0
196	3900	C3	1	1	0	0	0	0	1	1
197	3920	C4	1	1	0	0	0	1	0	0
198	3940	C5	1	1	0	0	0	1	0	1
199	3960	C6	1	1	0	0	0	1	1	0
200	3980	C7	1	1	0	0	0	1	1	1
201	4000	C8	1	1	0	0	1	0	0	0
202	4020	C9	1	1	0	0	1	0	0	1
203	4040	CA	1	1	0	0	1	0	1	0
204	4060	CB	1	1	0	0	1	0	1	1
205	4080	CC	1	1	0	0	1	1	0	0
206	4100	CD	1	1	0	0	1	1	0	1
207	4120	CE	1	1	0	0	1	1	1	0
208	4140	CF	1	1	0	0	1	1	1	1
209	4160	D0	1	1	0	1	0	0	0	0
210	4180	D1	1	1	0	1	0	0	0	1
211	4200	D2	1	1	0	1	0	0	1	0
212	4220	D3	1	1	0	1	0	0	1	1
213	4240	D4	1	1	0	1	0	1	0	0
214	4260	D5	1	1	0	1	0	1	0	1
215	4280	D6	1	1	0	1	0	1	1	0
216	4300	D7	1	1	0	1	0	1	1	1
217	4320	D8	1	1	0	1	1	0	0	0
218	4340	D9	1	1	0	1	1	0	0	1
219	4360	DA	1	1	0	1	1	0	1	0
220	4380	DB	1	1	0	1	1	0	1	1
221	4400	DC	1	1	0	1	1	1	0	0
222	4420	DD	1	1	0	1	1	1	0	1
223	4440	DE	1	1	0	1	1	1	1	0
224	4460	DF	1	1	0	1	1	1	1	1
225	4480	E0	1	1	1	0	0	0	0	0

Lampiran 1. Tabel Tegangan Input ADC  
(Lanjutan)

226	4500	E1	1	1	1	0	0	0	0	1
227	4520	E2	1	1	1	0	0	0	1	0
228	4540	E3	1	1	1	0	0	0	1	1
229	4560	E4	1	1	1	0	0	1	0	0
230	4580	E5	1	1	1	0	0	1	0	1
231	4600	E6	1	1	1	0	0	1	1	0
232	4620	E7	1	1	1	0	0	1	1	1
233	4640	E8	1	1	1	0	1	0	0	0
234	4660	E9	1	1	1	0	1	0	0	1
235	4680	EA	1	1	1	0	1	0	1	0
236	4700	EB	1	1	1	0	1	0	1	1
237	4720	EC	1	1	1	0	1	1	0	0
238	4740	ED	1	1	1	0	1	1	0	1
239	4760	EE	1	1	1	0	1	1	1	0
240	4780	EF	1	1	1	0	1	1	1	1
241	4800	F0	1	1	1	1	0	0	0	0
242	4820	F1	1	1	1	1	0	0	0	1
243	4840	F2	1	1	1	1	0	0	1	0
244	4860	F3	1	1	1	1	0	0	1	1
245	4880	F4	1	1	1	1	0	1	0	0
246	4900	F5	1	1	1	1	0	1	0	1
247	4920	F6	1	1	1	1	0	1	1	0
248	4940	F7	1	1	1	1	0	1	1	1
249	4960	F8	1	1	1	1	1	0	0	0
250	4980	F9	1	1	1	1	1	0	0	1
251	5000	FA	1	1	1	1	1	0	1	0
252	5020	FB	1	1	1	1	1	0	1	1
253	5040	FC	1	1	1	1	1	1	0	0
254	5060	FD	1	1	1	1	1	1	0	1
255	5080	FE	1	1	1	1	1	1	1	0
256	5100	FF	1	1	1	1	1	1	1	1

Lampiran 2. Tabel Tegangan Output Sensor Temperatur LM35

Temperature (°C)	Vout Sensor (mV)	No.	Temperature (°C)	Vout Sensor (mV)	No.	Temperature (°C)	Vout Sensor (mV)
-55	-550	21	-35	-350	41	-15	-150
-54	-540	22	-34	-340	42	-14	-140
-53	-530	23	-33	-330	43	-13	-130
-52	-520	24	-32	-320	44	-12	-120
-51	-510	25	-31	-310	45	-11	-110
-50	-500	26	-30	-300	46	-10	-100
-49	-490	27	-29	-290	47	-9	-90
-48	-480	28	-28	-280	48	-8	-80
-47	-470	29	-27	-270	49	-7	-70
-46	-460	30	-26	-260	50	-6	-60
-45	-450	31	-25	-250	51	-5	-50
-44	-440	32	-24	-240	52	-4	-40
-43	-430	33	-23	-230	53	-3	-30
-42	-420	34	-22	-220	54	-2	-20
-41	-410	35	-21	-210	55	-1	-10
-40	-400	36	-20	-200	56	0	0
-39	-390	37	-19	-190	57	1	10
-38	-380	38	-18	-180	58	2	20
-37	-370	39	-17	-170	59	3	30
-36	-360	40	-16	-160	60	4	40

Temperature (°C)	Vout Sensor (mV)	No.	Temperature (°C)	Vout Sensor (mV)	No.	Temperature (°C)	Vout Sensor (mV)
5	50	81	25	250	101	45	450
6	60	82	26	260	102	46	460
7	70	83	27	270	103	47	470
8	80	84	28	280	104	48	480
9	90	85	29	290	105	49	490
10	100	86	30	300	106	50	500
11	110	87	31	310	107	51	510
12	120	88	32	320	108	52	520
13	130	89	33	330	109	53	530
14	140	90	34	340	110	54	540
15	150	91	35	350	111	55	550
16	160	92	36	360	112	56	560
17	170	93	37	370	113	57	570
18	180	94	38	380	114	58	580
19	190	95	39	390	115	59	590
20	200	96	40	400	116	60	600
21	210	97	41	410	117	61	610
22	220	98	42	420	118	62	620
23	230	99	43	430	119	63	630
24	240	100	44	440	120	64	640

Lampiran 2. Tabel Tegangan Output Sensor Temperatur LM35 (Lanjutan)

Temperature (°C)	Vout Sensor (mV)	No.	Temperature (°C)	Vout Sensor (mV)	No.	Temperature (°C)	Vout Sensor (mV)
65	650	141	85	850	161	105	1050
66	660	142	86	860	162	106	1060
67	670	143	87	870	163	107	1070
68	680	144	88	880	164	108	1080
69	690	145	89	890	165	109	1090
70	700	146	90	900	166	110	1100
71	710	147	91	910	167	111	1110
72	720	148	92	920	168	112	1120
73	730	149	93	930	169	113	1130
74	740	150	94	940	170	114	1140
75	750	151	95	950	171	115	1150
76	760	152	96	960	172	116	1160
77	770	153	97	970	173	117	1170
78	780	154	98	980	174	118	1180
79	790	155	99	990	175	119	1190
80	800	156	100	1000	176	120	1200
81	810	157	101	1010	177	121	1210
82	820	158	102	1020	178	122	1220
83	830	159	103	1030	179	123	1230
84	840	160	104	1040	180	124	1240

Temperature (°C)	Vout Sensor (mV)	No.	Temperature (°C)	Vout Sensor (mV)
125	1250	201	145	1450
126	1260	202	146	1460
127	1270	203	147	1470
128	1280	204	148	1480
129	1290	205	149	1490
130	1300	206	150	1500
131	1310			
132	1320			
133	1330			
134	1340			
135	1350			
136	1360			
137	1370			
138	1380			
139	1390			
140	1400			
141	1410			
142	1420			
143	1430			
144	1440			



## LISTING PROGRAM

“PENGENDALIAN KECEPATAN PUTARAN MOTOR DC  
TERHADAP PERUBAHAN TEMPERATUR LINGKUNGAN  
DENGAN SISTEM MODULASI LEBAR PULSA”

```

-----
                ORG    00H

SR0    EQU    30H
SR1    EQU    31H
SR2    EQU    32H
SR3    EQU    33H
SR4    EQU    34H
SR5    EQU    35H
SR6    EQU    36H
SR7    EQU    37H

START:                LCALL  INITLCD
                    LCALL  DISPLAY_MAIN
                    SETB   P2.2
K_START:              JB     P2.1,START
                    JMP    AKTIF_ADC
AKTIF_ADC:            SETB   P2.0
                    SETB   P2.2
                    LCALL  DELAY
                    CLR    P2.0
                    LCALL  DISPLAY_WAIT
                    LCALL  DELAY
                    LCALL  DELAY
                    SJMP   S_TEMP

S_TEMP:              MOV    A,P1
TEMP16:              CJNE   A,#08H,TEMP18
                    LCALL  DISPLAY_16
                    MOV    R0,#20H
SP1:                 CLR    P2.2
RD1:                 JNB   P2.7,$

```

Lampiran 3. Listing Program (Lanjutan)

```
DJNZ R0,SP1
LCALL DELAY1MS
LCALL DELAY1MS
LCALL DELAY1MS
LCALL DELAY1MS
LCALL DELAY1MS
LCALL DELAY1MS
LJMP N

TEMP18: CJNE A,#09H,TEMP20
LCALL DISPLAY_18
MOV R0,#20H
SP2: CLR P2.2
RD2: JNB P2.7,$
DJNZ R0,SP2
LCALL DELAY1MS
LCALL DELAY1MS
LCALL DELAY1MS
LCALL DELAY1MS
LCALL DELAY1MS
LJMP N

TEMP20: CJNE A,#0AH,TEMP22
LCALL DISPLAY_20
MOV R0,#20H
SP3: CLR P2.2
RD3: JNB P2.7,$
DJNZ R0,SP3
LCALL DELAY1MS
LCALL DELAY1MS
LCALL DELAY1MS
LCALL DELAY1MS
LCALL DELAY1MS
LJMP N

TEMP22: CJNE A,#0BH,TEMP24
LCALL DISPLAY_22
MOV R0,#22H
```

Lampiran 3. Listing Program (Lanjutan)

```
SP4:          CLR  P2.2
RD4:          JNB  P2.7,$
              DJNZ R0,SP4
              LCALL DELAY1MS
              LCALL DELAY1MS
              LCALL DELAY1MS
              LCALL DELAY1MS
              Ljmp  N

TEMP24:       CJNE A,#0CH,TEMP26
              LCALL DISPLAY_24
              MOV  R0,#22H

SP5:          CLR  P2.2
RD5:          JNB  P2.7,$
              DJNZ R0,SP5
              LCALL DELAY1MS
              LCALL DELAY1MS
              LCALL DELAY1MS
              LCALL DELAY1MS
              Ljmp  N

TEMP26:       CJNE A,#0DH,TEMP28
              LCALL DISPLAY_26
              MOV  R0,#22H

SP6:          CLR  P2.2
RD6:          JNB  P2.7,$
              DJNZ R0,SP6
              LCALL DELAY1MS
              LCALL DELAY1MS
              LCALL DELAY1MS
              LCALL DELAY1MS
              Ljmp  N

TEMP28:       CJNE A,#0EH,TEMP30
              LCALL DISPLAY_28
              MOV  R0,#24H

SP7:          CLR  P2.2
```

Lampiran 3. Listing Program (Lanjutan)

```
RD7:          JNB  P2.7,$
              DJNZ R0,SP7
              LCALL DELAY1MS
              LCALL DELAY1MS
              Ljmp  N

TEMP30:       CJNE A,#0FH,TEMP32
              LCALL DISPLAY_30
              MOV  R0,#24H

SP8:          CLR  P2.2
RD8:          JNB  P2.7,$
              DJNZ R0,SP8
              LCALL DELAY1MS
              LCALL DELAY1MS
              Ljmp  N

TEMP32:       CJNE A,#10H,TEMP34
              LCALL DISPLAY_32
              MOV  R0,#24H

SP9:          CLR  P2.2
RD9:          JNB  P2.7,$
              DJNZ R0,SP9
              LCALL DELAY1MS
              LCALL DELAY1MS
              Ljmp  N

TEMP34:       CJNE A,#11H,TEMP36
              LCALL DISPLAY_34
              MOV  R0,#25H

SP10:         CLR  P2.2
RD10:         JNB  P2.7,$
              DJNZ R0,SP10
              LCALL DELAY1MS
              Ljmp  N

TEMP36:       CJNE A,#12H,TEMP38
              LCALL DISPLAY_36
```

Lampiran 3. Listing Program (Lanjutan)

```

MOV R0,#25H
SP11: CLR P2.2
RD11: JNB P2.7,$
      DJNZ R0,SP11
      LCALL DELAY1MS
      LJMPL N

TEMP38: CJNE A,#13H,TEMP40
        LCALL DISPLAY_38
        MOV R0,#25H
SP12: CLR P2.2
RD12: JNB P2.7,$
      DJNZ R0,SP12
      LCALL DELAY1MS
      LJMPL N

TEMP40: CJNE A,#14H,TEMP42
        LCALL DISPLAY_40
        CLR P2.2
        LJMPL N

TEMP42: CJNE A,#14H,TEMP44
        LCALL DISPLAY_42
        CLR P2.2
        LJMPL N

TEMP44: CJNE A,#15H,TEMP18
        LCALL DISPLAY_16
        CLR P2.2
        LJMPL N

W: LCALL DISPLAY_DATAW
   CLR P2.2
   LJMPL N

N: JB P2.1,JMP
   SETB P2.2
   LJMPL START

```

Lampiran 3. Listing Program (Lanjutan)

```
JMP:                LJMPL    S_TEMP

DISPLAY_MAIN:      LCALL    DSP_FIRST
WRITE_LCD_MAIN:    LCALL    WRITE_LCD
INLINE_MAIN:       LCALL    MAIN_DISPLAY
                   RET
DSP_FIRST:         LCALL    CLRDSP
DSP1_FIRST:        LCALL    DISPLAY
                   MOV     DPTR,#DSP_L1_F
                   RET
MAIN_DISPLAY:      MOV     DPTR,#DSP_L2_M
                   LCALL    WRITELCD
                   LCALL    DELAY1S
                   LCALL    DELAY1S
                   RET

DISPLAY_WAIT:      LCALL    DSP_WAIT
WRITE_LCD_WAIT:    LCALL    WRITE_LCD
INLINE_WAIT:       LCALL    WAIT_DISPLAY
                   RET
DSP_WAIT:          LCALL    CLRDSP
DSP1_WAIT:         LCALL    DISPLAY
                   MOV     DPTR,#DSP_L1_WAWIT
                   RET
WAIT_DISPLAY:      MOV     DPTR,#DSP_L2_WAIT
                   LCALL    WRITELCD
                   LCALL    DELAY1S
                   LCALL    DELAY1S
                   RET

DISPLAY_16:        LCALL    L1D1
WRITE_D1_1:        LCALL    WRITE_LCD
INLINE_D1_1:       LCALL    L2D1
                   RET
L1D1:              LCALL    CLRDSP
DSP1_D1:           LCALL    DISPLAY
```

Lampiran 3. Listing Program (Lanjutan)

```

MOV DPTR,#DSP_L1_1
RET
L2D1: MOV DPTR,#DSP_L2_1
LCALL WRITELCD
LCALL DELAY1S
LCALL DELAY1S
RET
DISPLAY_18: LCALL L1D2
WRITE_D1_2: LCALL WRITE_LCD
INLINE_D1_2: LCALL L2D2
RET
L1D2: LCALL CLRDSP
DSP1_D2: LCALL DISPLAY
MOV DPTR,#DSP_L1_2
RET
L2D2: MOV DPTR,#DSP_L2_1
LCALL WRITELCD
LCALL DELAY1S
LCALL DELAY1S
RET
DISPLAY_20: LCALL L1D3
WRITE_D1_3: LCALL WRITE_LCD
INLINE_D1_3: LCALL L2D3
RET
L1D3: LCALL CLRDSP
DSP1_D3: LCALL DISPLAY
MOV DPTR,#DSP_L1_3
RET
L2D3: MOV DPTR,#DSP_L2_1
LCALL WRITELCD
LCALL DELAY1S
LCALL DELAY1S
RET
DISPLAY_22: LCALL L1D4

```

Lampiran 3. Listing Program (Lanjutan)

```
WRITE_D1_4:      LCALL WRITE_LCD
INLINE_D1_4:    LCALL L2D4
                RET
L1D4:          LCALL CLRDSP
DSP1_D4:       LCALL DISPLAY
                MOV DPTR,#DSP_L1_4
                RET
L2D4:          MOV DPTR,#DSP_L2_2
                LCALL WRITELCD
                LCALL DELAY1S
                LCALL DELAY1S
                RET

DISPLAY_24:    LCALL L1D5
WRITE_D1_5:    LCALL WRITE_LCD
INLINE_D1_5:  LCALL L2D5
                RET
L1D5:          LCALL CLRDSP
DSP1_D5:       LCALL DISPLAY
                MOV DPTR,#DSP_L1_5
                RET
L2D5:          MOV DPTR,#DSP_L2_2
                LCALL WRITELCD
                LCALL DELAY1S
                LCALL DELAY1S
                RET

DISPLAY_26:    LCALL L1D6
WRITE_D1_6:    LCALL WRITE_LCD
INLINE_D1_6:  LCALL L2D6
                RET
L1D6:          LCALL CLRDSP
DSP1_D6:       LCALL DISPLAY
                MOV DPTR,#DSP_L1_6
                RET
L2D6:          MOV DPTR,#DSP_L2_2
                LCALL WRITELCD
```



Lampiran 3. Listing Program (Lanjutan)

```
LCALL DELAY1S
LCALL DELAY1S
RET

DISPLAY_28:    LCALL L1D7
WRITE_D1_7:   LCALL WRITE_LCD
INLINE_D1_7:  LCALL L2D7
              RET
L1D7:         LCALL CLRDSP
DSP1_D7:      LCALL DISPLAY
              MOV DPTR,#DSP_L1_7
              RET
L2D7:         MOV DPTR,#DSP_L2_3
              LCALL WRITELCD
              LCALL DELAY1S
              LCALL DELAY1S
              RET

DISPLAY_30:   LCALL L1D8
WRITE_D1_8:   LCALL WRITE_LCD
INLINE_D1_8:  LCALL L2D8
              RET
L1D8:         LCALL CLRDSP
DSP1_D8:      LCALL DISPLAY
              MOV DPTR,#DSP_L1_8
              RET
L2D8:         MOV DPTR,#DSP_L2_3
              LCALL WRITELCD
              LCALL DELAY1S
              LCALL DELAY1S
              RET

DISPLAY_32:   LCALL L1D9
WRITE_D1_9:   LCALL WRITE_LCD
INLINE_D1_9:  LCALL L2D9
              RET
L1D9:         LCALL CLRDSP
```

Lampiran 3. Listing Program (Lanjutan)

```
DSP1_D9:      LCALL  DISPLAY
              MOV   DPTR,#DSP_L1_9
              RET

L2D9:        MOV   DPTR,#DSP_L2_3
              LCALL WRITELCD
              LCALL DELAY1S
              LCALL DELAY1S
              RET

DISPLAY_34:   LCALL  L1D10
WRITE_D1_10:  LCALL  WRITE_LCD
INLINE_D1_10: LCALL  L2D10
              RET

L1D10:       LCALL  CLRDSP
DSP1_D10:    LCALL  DISPLAY
              MOV   DPTR,#DSP_L1_10
              RET

L2D10:       MOV   DPTR,#DSP_L2_4
              LCALL WRITELCD
              LCALL DELAY1S
              LCALL DELAY1S
              RET

DISPLAY_36:   LCALL  L1D11
WRITE_D1_11:  LCALL  WRITE_LCD
INLINE_D1_11: LCALL  L2D11
              RET

L1D11:       LCALL  CLRDSP
DSP1_D11:    LCALL  DISPLAY
              MOV   DPTR,#DSP_L1_11
              RET

L2D11:       MOV   DPTR,#DSP_L2_4
              LCALL WRITELCD
              LCALL DELAY1S
              LCALL DELAY1S
              RET
```

Lampiran 3. Listing Program (Lanjutan)

```
DISPLAY_38:      LCALL  L1D12
WRITE_D1_12:    LCALL  WRITE_LCD
INLINE_D1_12:   LCALL  L2D12
                RET
L1D12:          LCALL  CLRDSP
DSP1_D12:       LCALL  DISPLAY
                MOV   DPTR,#DSP_L1_12
                RET
L2D12:          MOV   DPTR,#DSP_L2_4
                LCALL WRITELCD
                LCALL DELAY1S
                LCALL DELAY1S
                RET

DISPLAY_40:     LCALL  L1D13
WRITE_D1_13:    LCALL  WRITE_LCD
INLINE_D1_13:   LCALL  L2D13
                RET
L1D13:          LCALL  CLRDSP
DSP1_D13:       LCALL  DISPLAY
                MOV   DPTR,#DSP_L1_13
                RET
L2D13:          MOV   DPTR,#DSP_L2_5
                LCALL WRITELCD
                LCALL DELAY1S
                LCALL DELAY1S
                RET

DISPLAY_42:     LCALL  L1D14
WRITE_D1_14:    LCALL  WRITE_LCD
INLINE_D1_14:   LCALL  L2D14
                RET
L1D14:          LCALL  CLRDSP
DSP1_D14:       LCALL  DISPLAY
                MOV   DPTR,#DSP_L1_14
                RET
L2D14:          MOV   DPTR,#DSP_L2_5
```

Lampiran 3. Listing Program (Lanjutan)

```
LCALL WRITELCD
LCALL DELAY1S
LCALL DELAY1S
RET

DISPLAY_44:      LCALL L1D15
WRITE_D1_15:    LCALL WRITE_LCD
INLINE_D1_15:   LCALL L2D15
                RET
L1D15:          LCALL CLRDSP
DSP1_D15:       LCALL DISPLAY
                MOV  DPTR,#DSP_L1_15
                RET
L2D15:          MOV  DPTR,#DSP_L2_5
                LCALL WRITELCD
                LCALL DELAY1S
                LCALL DELAY1S
                RET

DISPLAY_DATAW:  LCALL L1DW
WRITE_D1_W:     LCALL WRITE_LCD
INLINE_D1_W:   LCALL L2DW
                RET
L1DW:           LCALL CLRDSP
DSP1_DW:        LCALL DISPLAY
                MOV  DPTR,#DSP_L1_W
                RET
L2DW:           MOV  DPTR,#DSP_L2_W
                LCALL WRITELCD
                LCALL DELAY1S
                LCALL DELAY1S
                RET

CLR_DSP_LCD:    LCALL CLR_DSP
                RET

INITLCD:        MOV  A,#3FH
```

Lampiran 3. Listing Program (Lanjutan)

```
LCALL WRITEREG
LCALL WRITEREG
LCALL WRITEREG
LCALL DELAY1MS
LCALL DELAY1MS
MOV A,#0EH
LCALL WRITEREG
LCALL DELAY1MS
LCALL DELAY1MS
MOV A,#38H
LCALL WRITEREG
LCALL DELAY1MS
LCALL DELAY1MS
MOV A,#01H
LCALL WRITEREG
LCALL DELAY1MS
LCALL DELAY1MS
MOV A,#06H
LCALL WRITEREG
LCALL DELAY1MS
LCALL DELAY1MS
MOV A,#80H
LCALL WRITEREG
LCALL DELAY1MS
LCALL DELAY1MS
MOV A,#0CH
LCALL WRITEREG
LCALL DELAY1MS
LCALL DELAY1MS
MOV A,#3CH
LCALL WRITEREG
LCALL DELAY1MS
LCALL DELAY1MS
RET
CLRDSF: MOV A,#00H
LCALL WRITEREG
RET
```

Lampiran 3. Listing Program (Lanjutan)

```
BARIS1:      MOV  A,#80H
             LCALL DELAY10MS
             LCALL WRITEREG
             RET

BARIS2;      LCALL A,#0COH
             LCALL DELAY10MS
             LCALL WRITEREG
             RET

WRITEREG:    CLR  P0.0
             CLR  P0.1
             CLR  P0.2
             MOV  P3,A
             SETB P0.2
             LCALL DELAY1MS
             LCALL DELAY1MS
             LCALL DELAY1MS
             CLR  P0.2
             RET

WRITELCD:    DEC  R7
             CLR  A
             MOVC A,@A+DPTR
             CJNE R7,#00H,WRITELCD_1
             RET

WRITELCD_1:  INC  DPTR
             LCALL DISP_LCD
             LJMPC WRITELCD
             RET

DISP_LCD:    CLR  P0.2
             SETB P0.0
             CLR  P0.1
             MOV  P3,A
             SETB P0.2
             LCALL DELAY1MS
             LCALL DELAY1MS
             LCALL DELAY1MS
             CLR  P0.2
             RET
```

Lampiran 3. Listing Program (Lanjutan)

```

CLR_DPS:      SETB  P0.2
              CLR   P0.0
              MOV   P3,#01H
              CLR   P0.2
              RET

DELAY:        DJNZ  SR5,DELAY
              DJNZ  SR6,DELAY
              RET

DELAY10MS:    MOV   SR2,#14H
DELAY10MS_1:  LCALL DELAY10MS_2
              DJNZ  SR2,DELAY10MS_1
              RET

DELAY10MS_2:  MOV   SR3,#0F7H
              DJNZ  SR3,$
              RET

DELAY1MS:     MOV   SR2,#02H
DELAY1MS_1:   LCALL DELAY1MS_2
              DJNZ  SR2,DELAY1MS_1
              RET

DELAY1MS_2:   MOV   SR3,#0F6H
              DJNZ  SR3,$
              RET

DELAY10S:     MOV   SR2,#64H
DELAY10S_1:   LCALL LDELAY10S
              DJNZ  SR2,DELAY10S_1
              RET

LDELAY10S:    MOV   SR3,#0C8H
LDELAY10S_2:  LCALL LDELAY10S_3
              DJNZ  SR3,LDELAY10S_2
              RET

LDELAY10S_3:  MOV   SR4,#0F7H
              DJNZ  SR4,$
              RET

DELAY1S:      MOV   SR2,#0AH
DELAY1S_1:    LCALL LDELAY1S
              DJNZ  SR2,DELAY1S_1

```

Lampiran 3. Listing Program (Lanjutan)

```

RET
LDELAY1S:      MOV   SR3,#0C8H
LDELAY1S_2:    LCALL LDELAY10S_3
               DJNZ  SR3,LDELAY1S_2
               RET
LDELAY1S_3:    MOV   SR4,#0F7H
               DJNZ  SR4,$
               RET
DISPLAY:      LCALL  BARIS1
               MOV   R7,#10H
               CLR   A
               RET
WRITE_LCD:    LCALL  WRITELCD
               LCALL  BARIS2
               MOV   R7,#10H
               CLR   A
               RET
DSP_L1_F:     DB    'M. HAMDANI '
               RET
DSP_L2_M:     DB    '0806 366 094 '
               RET
DSP_L1_WAIT:  DB    'MOHON TUNGGU...'
               RET
DSP_L2_WAIT:  DB    'BACA TEMPERATUR'
               RET
DSP_L1_1:     DB    'TEMP. : 16 degC'
               RET
DSP_L1_2:     DB    'TEMP. : 18 degC'
               RET
DSP_L1_3:     DB    'TEMP. : 20 degC'
               RET
DSP_L1_4:     DB    'TEMP. : 22 degC'
               RET
DSP_L1_5:     DB    'TEMP. : 24 degC'

```



Lampiran 3. Listing Program (Lanjutan)

```
RET
DSP_L1_6:    DB    'TEMP. : 26 degC'
RET
DSP_L1_7:    DB    'TEMP. : 28 degC'
RET
DSP_L1_8:    DB    'TEMP. : 30 degC'
RET
DSP_L1_9:    DB    'TEMP. : 32 degC'
RET
DSP_L1_10:   DB    'TEMP. : 34 degC'
RET
DSP_L1_11:   DB    'TEMP. : 36 degC'
RET
DSP_L1_12:   DB    'TEMP. : 38 degC'
RET
DSP_L1_13:   DB    'TEMP. : 40 degC'
RET
DSP_L1_14:   DB    'TEMP. : 42 degC'
RET
DSP_L1_15:   DB    'TEMP. : 44 degC'
RET
DSP_L1_W:    DB    'OVER TEMPERATUR'
RET

DSP_L2_1:    DB    'SPEED: 2000 rpm'
RET
DSP_L2_2:    DB    'SPEED: 2100 rpm'
RET
DSP_L2_3:    DB    'SPEED: 2200 rpm'
RET
DSP_L2_4:    DB    'SPEED: 2300 rpm'
RET
DSP_L2_5:    DB    'SPEED: 2400 rpm'
RET

END
```