



SIMULATOR SAKLAR OTOMATIS KIPAS ANGIN MENGGUNAKAN SENSOR SUHU

TUGAS AKHIR

**Untuk memperoleh gelar Ahli Mada pada
Program Diploma III Teknik Elektro
Jurusan Teknik – Fakultas Teknik
Universitas Negeri Semarang**

**Oleh
Muhammad Afton Badruzzaman
5350305063**

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG
2010**

PENGESAHAN

Tugas Akhir ini telah dipertahankan dihadapan Panitia Ujian Tugas Akhir Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Semarang pada tanggal 20 Agustus 2010.

Panitia :

Ketua

Sekretaris

Drs. Said Sunardyo, M.T.
NIP:196505121991031003

Drs. Agus Murnomo, M.T.
NIP:195506061986031002

Penguji 1

Penguji II / Pembimbing

Drs Samiyono, M.T.
NIP:194906161975011002

Drs. Usman Nurzaman, M.Pd.
NIP: 194904051975011001

Mengetahui
Dekan Fakultas Teknik

Drs. Abdurrahman, M.Pd.
NIP: 196009031985031002

MOTTO DAN PERSEMBAHAN

Motto :

- ” Tuhan menciptakan segala sesuatu akan indah pada waktu-Nya, jalani hidup ini dengan penuh semangat, kerja keras dan pantang menyerah....”.
- ”If you can't explain it simply, you don't understand it well enough”.
- ” Jika hujan bagaikan kegagalan dan mentari bagaikan kesuksesan maka kita butuh keduanya untuk dapat melihat pelangi ”.

Persembahan :

1. Allah Maha Robb, yang mengantarkan pembelajaran tiada henti.
2. Baktiku selalu untuk Bapak dan Ibu.
3. Yang membuatku selalu tegar, tanteku Ci' An, kakak-kakakku Husnia Aklifa, Hasna Alifa, Hikma Rusyda, Hanna Izza, Iffa Humaida, adik-adikku Naila Zulfa Ittriya dan Arina Hilma Istiya.
4. Keluarga besar UKM Riptek.
5. Sahabat-sahabatku Bowo, Mala, Amin, Agos Cacing, Ahong anak-anak Calem kos, Parahiangan kos, L.A kos dan Vega kos.

KATA PENGANTAR

Puji dan syukur senantiasa kami panjatkan kehadirat Allah SWT, yang telah melimpahkan rahmat, pertolongan dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan tugas akhir ini dengan judul “ **Simulator Saklar Otomatis Kipas Angin Menggunakan Sensor Suhu** “, sebagai syarat menempuh jenjang Diploma III Teknik Elektro Universitas Negeri Semarang.

Penulisan laporan tugas akhir ini tidak lepas dari pemikiran dan bantuan dari berbagai pihak. Oleh karena itu penulis menyampaikan ucapan terima kasih kepada Yth.:

1. Drs. Abdurrahman, M.Pd., Dekan Fakultas Teknik Universitas Negeri Semarang.
2. Drs. Djoko Adi Widodo, M.T., Ketua Jurusan Teknik Elektro Universitas Negeri Semarang.
3. Drs. Agus Murnomo, M.T., Ketua Program Studi Diploma III Teknik Elektro Universitas Negeri Semarang.
4. Drs. Usman Nurzaman, M.Pd., dosen pembimbing yang telah membantu memberikan bimbingan terbaik hingga terselesaikannya tugas akhir ini.
5. Drs. Samiyono, M.T., dosen penguji yang telah mengoreksi seluruhnya menjadi lebih baik.
6. Drs. Said Sunardyo, M.T., sekretaris pengujian tugas akhir.
7. Bapak, Ibu, Kakak, Adik dan Tante/ku tercinta, terima kasih atas doa dan dukungannya.

Harapan penulis semoga karya Tugas Akhir ini bermanfaat bagi para pembaca.

Semarang, Agustus 2010

Penulis



ABSTRAK

Badruzzaman, Muhammad, Afton. 2010. “*Simulator Saklar Otomatis Kipas Angin Menggunakan Sensor Suhu* “. Tugas Akhir, Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang. Drs. Usman Nurzaman, M.Pd.

Kata Kunci : Kipas Angin Otomatis, Mikrokontroller, Sensor Suhu.

Perkembangan ilmu pengetahuan dan teknologi telah maju pesat terutama dalam bidang elektronika digital. Sistem digital berkembang dengan adanya sistem mikrokontroller, yang mengarah ke sebuah desain produk, teknik ini yang dinamakan *Embedded engineering*. Kipas angin adalah alat untuk menyetel kondisi ruangan.

Penyusunan laporan tugas akhir ini berdasarkan pada metode observasi, metode interview, metode dokumentasi dan metode studi pustaka. Fungsi kerja Otomatisasi Kipas Angin Menggunakan Sensor Suhu, yaitu memutar kipas AC dengan deteksi sensor suhu pada suhu ruangan.

sensor suhu LM35 mendeteksi suhu dalam ruangan yang besarnya ditampilkan oleh *Liquid Crystal Display* (LCD), ATmega8535 memberikan perintah adanya kenaikan suhu oleh sensor suhu LM35 yang sudah diatur sebesar lebih dari 33°C pada *driver* sensor suhu LM35. Ketika suhu ruangan mencapai lebih dari 33°C baling-baling kipas angin berputar.

Kesimpulan yang dapat diambil dari perakitan dan pembuatan Otomatisasi Kipas Angin Menggunakan Sensor Suhu ini adalah Sensor suhu dapat mendeteksi suhu dalam ruangan, jika suhu ruangan lebih dari 33°C kipas berputar dan jika kurang dari 33°C kipas berhenti berputar. Alat Otomatisasi Kipas Angin Menggunakan Sensor Suhu ini dapat digunakan di dalam ruangan sebagai penyetel kondisi ruangan yang bekerja secara otomatis.

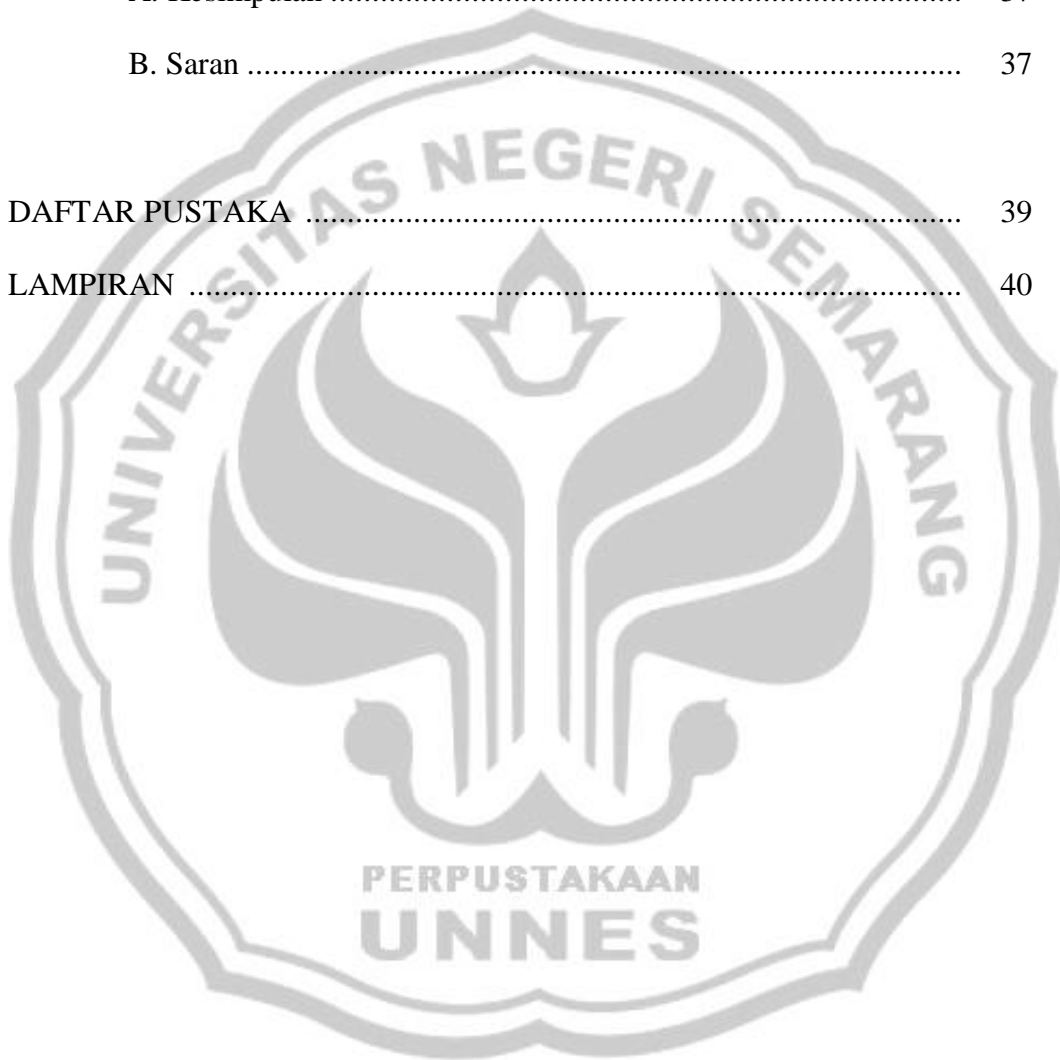
PERPUSTAKAAN
UNNES

DAFTAR ISI

Halaman Judul	i
Lembar Pengesahan	ii
Motto dan Persembahan	iii
Kata Pengantar	iv
Abstrak	vi
Daftar Isi	vii
Daftar Tabel	x
Daftar Gambar	xi
Daftar Lampiran.....	xii
BAB 1 PENDAHULUAN	
A. Latar Belakang	1
B. Tujuan	2
C. Manfaat	2
D. Rumusan Masalah	3
E. Pembatasan Masalah	3
BAB 2 PEMBAHASAN	
A. Landasan Teori	
1. Sensor suhu LM35	4
2. Mikrokontroler	7

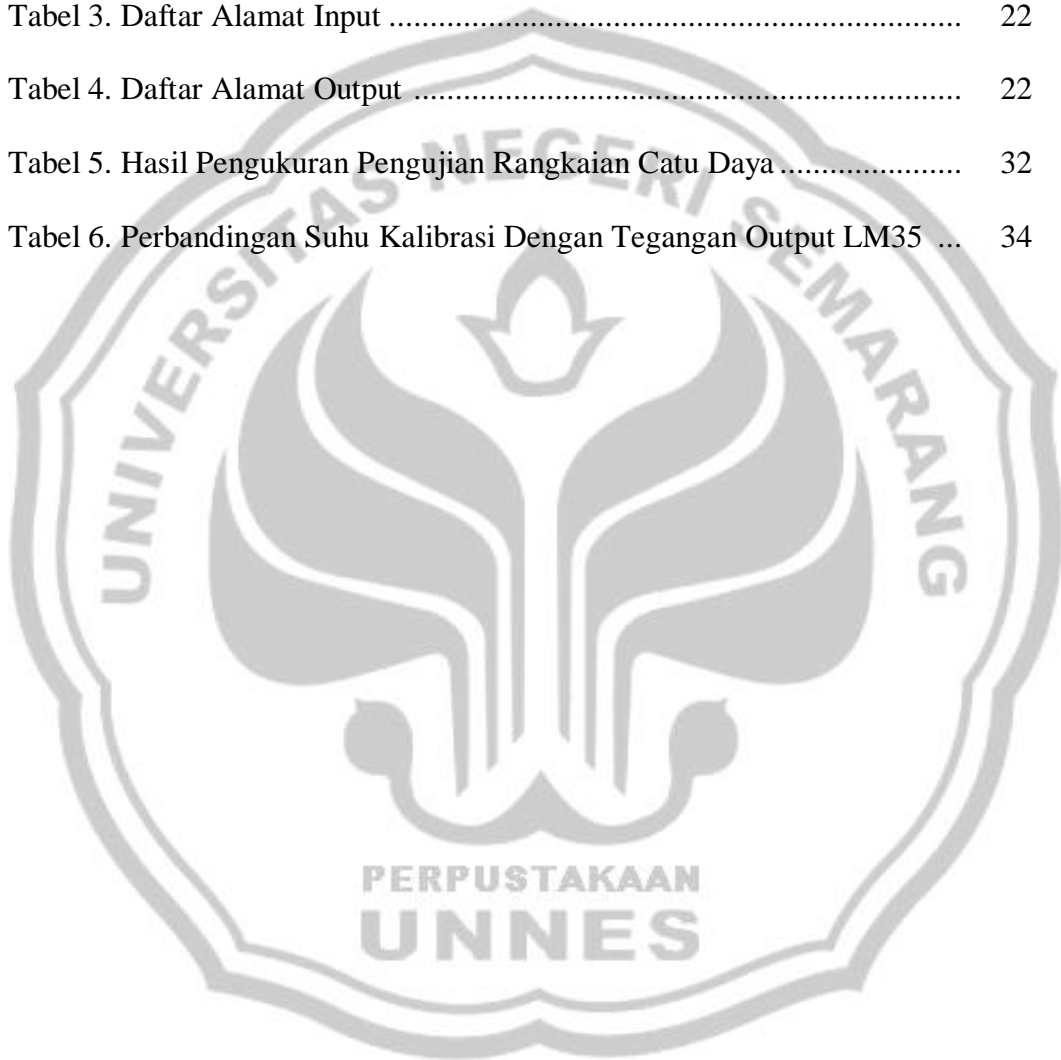
a. Pengertian Mikrokontroler	7
b. Mikrokontroler ATmega8535	9
1. Fitur ATmega8535.....	10
2. Deskripsi Pin	11
3. Arsitektur ATmega8535	13
4. Organisasi Memori	15
a. Memori Program	15
b. Memori Data	16
c. Memori EEPROM	17
4. <i>Liquid Crystal Display</i> (LCD M1632)	17
B. Perakitan	19
1. Metode Laboratoris.....	19
3. Instrumen	20
4. Teknik Analisis Data	20
C. Pembuatan Alat	20
1. Pembuatan Perangkat Lunak (<i>Software</i>).....	20
2. Pembuatan Perangkat Keras (<i>Hardware</i>)	25
3. Pembuatan Pesawat Simulasi	29
D. Hasil Pengujian	32
1. Pengujian Rangkaian Catu Daya	32
2. Pengujian Rangkaian LCD.....	33
3. Pengujian Kerja Alat.....	33
E. Pembahasan	34

1. Analisis Perangkat Lunak	34
2. Analisis Kerja Alat	34
F. Deskripsi kerja Alat.....	35
BAB 3 PENUTUP	
A. Kesimpulan	37
B. Saran	37
DAFTAR PUSTAKA	39
LAMPIRAN	40



DAFTAR TABEL

Tabel 1. Deskripsi Pin ATmega8535	12
Tabel 2. Koneksi Antara Modul LCD dengan Mikrokontroler	18
Tabel 3. Daftar Alamat Input	22
Tabel 4. Daftar Alamat Output	22
Tabel 5. Hasil Pengukuran Pengujian Rangkaian Catu Daya	32
Tabel 6. Perbandingan Suhu Kalibrasi Dengan Tegangan Output LM35 ...	34



DAFTAR GAMBAR

Gambar 1. Jenis-jenis IC LM35 Berdasarkan Bentuk Package	6
Gambar 2. Konfigurasi Pin ATmega8535	11
Gambar 3. Arsitektur ATmega8535	13
Gambar 4. Organisasi Memori ATmega8535	15
Gambar 5. Register I/O sebagai Memori Data dan Sebagai I/O	17
Gambar 6. Rangkaian Koneksi LCD dengan Mikrokontroler	19
Gambar 7. <i>Flowchart</i> Program	22
Gambar 8. ISP Programmer interface Type Kanda System STK 200+/300	26
Gambar 9. Rangkaian Catu daya	27
Gambar 10. Koneksi LM35.....	21
Gambar 11. Lay Out PCB	31
Gambar 12. Tata Letak Komponen Rangkaian Sistem Minimum ATmega8535, Driver Motor Stepper, Catu Daya	32
Gambar 13. Rangkaian Otomatisasi Kipas Angin Menggunakan Sensor Suhu	40
Gambar 14. Rangkaian Otomatisasi Menggunakan Sensor Suhu	79
Gambar 15. Rangkaian Otomatisasi Sensor Suhu Dengan Kipas Angin.....	79
Gambar 16. Mikrokontroler ATmega8535.....	80
Gambar 17. Box(Panel) Alat Otomatisasi Kipas Angin Menggunakan Sensor Suhu	80

DAFTAR LAMPIRAN

Lampiran 1. Gambar Rangkaian.....	40
Lampiran 2. listing Program Kontrol Suhu.....	42
Lampiran 2. Datasheet LM35	66
Lampiran 3. Foto Alat	79



BAB 1

PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi elektronika saat ini sangat pesat, hal ini dapat dibuktikan banyak alat-alat yang diciptakan manusia untuk mempermudah kehidupannya. Teknologi telah mengubah cara-cara mengendalikan alat listrik yang bersifat manual menjadi otomatis. Salah satu alat listrik yang dikendalikan secara manual yaitu kipas angin. Kipas angin dapat dihidupkan dengan menekan tombol on/off, kemudian menekan tombol angka baru baling-baling kipas berputar. Alat listrik yang bersifat otomatis contohnya rice cooker. Hanya dengan memasukkan beras dan air secukupnya ke dalam ricee cooker kemudian tekan tombol cooke, tunggu hingga matang.

Di dalam perkembangan teknologi elektronik pengendali manual dapat diubah menjadi pengendali otomatis dengan sedikit kreatifitas manusia. Kreatifitas manusia dapat berupa otomatisasi pengatur kerja pada benda elektronik untuk menggantikan saklar off/on(tekan). Hal ini dikarenakan pada pemikiran studi pendahuluan yang di lakukan di dalam ruangan dapat digunakan berbagai kegiatan. Suhu ruangan yang tiba-tiba panas karena pengaruh cuaca panas dan banyak orang yang melakukan kegiatan ada di dalam satu ruangan dapat menghambat produktifitas suatu kegiatan, dibutuhkan kipas angin untuk mendinginkan suhu ruangan untuk menyelaraskan kondisi udara. Suhu kamar (Bahasa Inggris: *room temperature*)

atau suhu ruangan, dalam penggunaan [ilmiah](#), dianggap kurang lebih antara 20 sampai 25 derajat [Celsius](#) ($^{\circ}\text{C}$) (68 sampai 77 derajat [Fahrenheit](#) ($^{\circ}\text{F}$), 528 sampai 537 derajat [Rankine](#) ($^{\circ}\text{R}$), atau 293 sampai 298 [Kelvin](#) (K)), walaupun nilai tersebut bukanlah suatu nilai yang ditentukan dengan persis. Untuk kemudahan penghitungan, sering digunakan angka 20 $^{\circ}\text{C}$ atau 300 K. Rentang suhu dan [kelembapan](#) relatif dapat diterima di Indonesia karena pengaruh iklim yang panas. (<http://id.wikipedia.org>).Memperhatikan kondisi ini peneliti ingin membantu mengatasi masalah.

Caranya dengan mengubah saklar tekan menjadi otomatis, saklar kipas angin digantikan dengan menggunakan sensor suhu LM35 yang secara singkat prinsip kerja sensor suhu sebagai berikut ; sensor suhu LM35 mendeteksi suhu dalam ruangan yang besarnya ditampilkan oleh *Liquid Crystal Display* (LCD), ATmega8535 memberikan perintah adanya kenaikan suhu oleh sensor suhu LM35 yang sudah diatur sebesar lebih dari 33°C pada *driver* sensor suhu LM35. Ketika suhu ruangan mencapai lebih dari 33°C baling-baling kipas angin berputar.

B. Tujuan

Dalam pembuatan tugas akhir tujuan yang ingin dicapai adalah dapat:

1. Merangkai pengatur kerja kipas angin didasarkan pada suhu lingkungan.
2. Membuat alat pengendali kipas angin yang bekerja otomatis.

C. Manfaat

Pembuatan Tugas Akhir ini bermanfaat untuk :

1. Mempermudah dalam pemanfaatan kipas angin sebagai penyelemas kondisi ruangan karena dapat bekerja secara otomatis.
2. Membuat kondisi ruangan menjadi lebih segar.

D. Rumusan Masalah

Merumuskan masalah mulai dari :

1. Merancang pengatur kerja kipas angin dengan menggunakan sensor suhu.
2. Mengetahui cara kerja alat otomatisasi kipas angin menggunakan sensor suhu.

E. Pembatasan Masalah

Adapun pembatasan masalah mengenai tugas akhir ini meliputi :

1. Pusat pengendalian menggunakan mikrokontroler ATmega8535.
2. Sensor yang digunakan sebagai pendeteksi suhu di dalam ruangan adalah sensor suhu LM35.
3. Rangkaian dibuat dalam bentuk alat peraga atau simulasi namun dapat diaplikasikan dengan menggunakan sensor dan alat-alat yang sesungguhnya.

BAB 2

PEMBAHASAN

A. Landasan Teori

1. Sensor Suhu LM35

Sensor suhu LM35 ialah sensor temperatur paling banyak digunakan untuk praktek. Selain harganya cukup murah, sensor suhu LM35 tidak membutuhkan kalibrasi eksternal yang menyediakan akurasi $\pm 1/4^{\circ}\text{C}$ pada temperatur ruangan dan $\pm 3/4^{\circ}\text{C}$ pada kisaran -55°C hingga $+150^{\circ}\text{C}$. Sensor suhu LM35 dimaksudkan untuk beroperasi pada -55° hingga $+150^{\circ}\text{C}$, sedangkan sensor suhu LM35C pada -40°C hingga $+110^{\circ}\text{C}$, dan sensor suhu LM35D pada kisaran 0°C - 100°C . Sensor suhu LM35D juga tersedia pada paket 8 kaki dan paket TO-220. Sensor suhu LM35 umumnya akan naik sebesar 10mV setiap kenaikan 1°C (300mV pada 30°C).

Sensor suhu LM35 mempunyai 3 pin, masing-masing pin mempunyai fungsi yang berbeda, pin 1 berfungsi sebagai sumber tegangan kerja dari sensor suhu LM35, pin 2 atau tengah digunakan sebagai tegangan keluaran atau V_{out} dengan jangkauan kerja dari 0 Volt sampai dengan 1,5Volt dengan tegangan operasi sensor suhu LM35 yang dapat digunakan antar 4Volt sampai 30Volt. Keluaran sensor ini akan naik sebesar 10mV setiap derajat *celcius* sehingga diperoleh persamaan sebagai berikut :

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

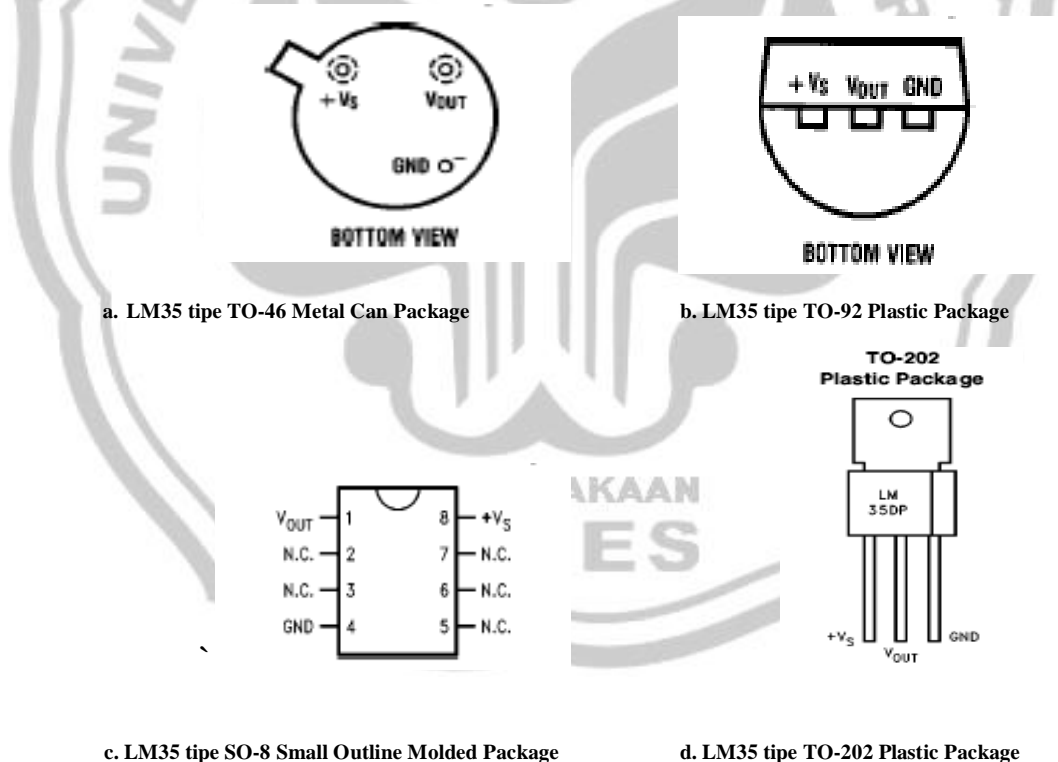
Secara prinsip sensor suhu akan melakukan penginderaan pada saat perubahan suhu setiap suhu 1°C akan menunjukkan tegangan sebesar 10mV . Pada penempatannya sensor suhu LM35 dapat ditempelkan dengan perekat atau dapat pula disemen pada permukaan akan tetapi suhunya akan sedikit berkurang sekitar $0,01^{\circ}\text{C}$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

Jarak yang jauh diperlukan penghubung yang tidak terpengaruh oleh *interferensi* dari luar, dengan demikian digunakan kabel selubung yang ditanahkan sehingga dapat bertindak sebagai suatu antena penerima dan simpangan di dalamnya, juga dapat bertindak sebagai perata arus yang mengoreksi pada kasus yang sedemikian, dengan menggunakan metode *bypass* kapasitor dari V_{in} untuk ditanahkan. Berikut ini adalah karakteristik dari sensor suhu LM35, [alldataseet, www.alldatasheet.com/LM35,2010](http://www.alldatasheet.com/LM35,2010):

- a. Memiliki sensitivitas suhu, dengan faktor skala linier antara tegangan dan suhu $10\text{ mVolt}/^{\circ}\text{C}$, sehingga dapat dikalibrasi langsung dalam *celcius*.
- b. Memiliki ketepatan atau akurasi kalibrasi yaitu $0,5^{\circ}\text{C}$ pada suhu 25°C .
- c. Memiliki jangkauan maksimal operasi suhu antara -55°C sampai $+150^{\circ}\text{C}$.

- d. Bekerja pada tegangan 4 sampai 30 volt.
- e. Memiliki arus rendah yaitu kurang dari $60 \mu\text{A}$.
- f. Memiliki pemanasan sendiri yang rendah (*low-heating*) yaitu kurang dari $0,1 \text{ }^\circ\text{C}$ pada udara diam.
- g. Memiliki impedansi keluaran yang rendah yaitu $0,1 \text{ W}$ untuk beban 1 mA .
- h. Memiliki ketidaklinieran hanya sekitar $\pm \frac{1}{4} \text{ }^\circ\text{C}$.

Berikut diperlihatkan gambar beberapa jenis IC LM35 berdasarkan bentuk *package*:



Gambar 1. Jenis-jenis IC LM35 Berdasarkan Bentuk *Package*.

2. Mikrokontroler

a. Pengertian Mikrokontroler

Mikrokontroler adalah sebuah *single chip computer*, artinya dalam sebuah IC mikrokontroler telah terdapat ROM, RAM, EPROM, serial interface dan parallel interface, timer, counter, interrupt *controller*, converter Analog ke Digital, dan lainnya (sesuai fasilitas dalam mikrokontroler tersebut).(Ardi Winoto, 2008:3)

Atmel merupakan sebuah perusahaan semikonduktor yang didirikan pada tahun 1984 di Amerika Serikat. Perusahaan ini memproduksi berbagai macam komponen elektronik seperti mikrokontroler, pengendali sinyal digital, memori (EPROM, EEPROM), komponen untuk peralatan telekomunikasi, komponen untuk peralatan medis, komponen untuk peralatan militer dan lain-lain

Atmel sebagai salah satu vendor yang mengembangkan dan memasarkan produk elektronika telah menjadi suatu teknologi standar bagi desainer sistem elektronika masa kini. Dengan perkembangan terakhir, yaitu generasi AVR (*Alf and Vegard's Risc processor*). Mikrokontroler AVR memiliki arsitektur RISC 8 bit, dimana semua instruksi dikemas dalam kode 16-bit (16-bit word) dan sebagian besar instruksi dieksekusi dalam 1 siklus clock, berbeda dengan instruksi MCS51 yang membutuhkan 12 siklus clock. Tentu saja itu terjadi karena kedua mikrokontroler tersebut memiliki arsitektur yang berbeda. AVR berteknologi RISC (*Reduced instruction set computing*). Sedangkan seri MSC51 berteknologi CISC (*Complex instruction set*

computing). Secara umum, AVR dapat dikelompokkan menjadi 4 kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, mereka bisa dikatakan hampir sama. Penggunaan mikrokontroler ATmega8535 dikarenakan selain mudah didapatkan dan murah, ATmega8535 juga memiliki fasilitas yang lengkap. (Lingga, 2006:1).

Dalam miniatur ini digunakan mikrokontroler seri ATmega8535. ATmega8535 merupakan salah satu mikrokontroler buatan AVR yang memiliki fasilitas – fasilitas yang cukup lengkap, diantaranya :

1. Flash adalah suatu jenis *Read Only Memory* yang biasanya diisi dengan program hasil buatan manusia yang harus dijalankan oleh mikrokontroler.
2. RAM (*Random Acces Memory*) merupakan memori yang membantu CPU untuk penyimpanan data sementara dan pengolahan data ketika program sedang *running*.
3. EEPROM (*Electrically Erasable Programmable Read Only Memory*) adalah memori untuk penyimpanan data secara permanen oleh program yang sedang *running*.
4. Port I/O adalah kaki untuk jalur keluar atau masuk sinyal sebagai hasil keluaran ataupun masukan bagi program.

5. Timer adalah modul dalam hardware yang bekerja untuk menghitung waktu / pulsa.
6. UART (*Universal Asynchronous Receive Transmit*) adalah jalur komunikasi data khusus secara serial *asynchronous*.
7. PWM (*Pulse Width Modulation*) adalah fasilitas untuk membuat modulasi pulsa.
8. ADC (*Analog to Digital Converter*) adalah fasilitas untuk dapat menerima sinyal analog dalam range tertentu untuk kemudian dikonversi menjadi suatu nilai digital dalam range tertentu.
9. SPI (*Serial Peripheral Interface*) adalah jalur komunikasi data khusus secara serial *synchronous*.
10. ISP (*In System Programming*) adalah kemampuan khusus mikrokontroler untuk dapat diprogram langsung dalam sistem rangkaiannya dengan membutuhkan jumlah pin yang minimal. Oleh karena itu dipilihlah Mikrokontroler ATmega8535 sebagai pengendali bidang solar sel.

b. Mikrokontroler ATmega8535

Mikrokontroler ATmega8535 memiliki fitur yang cukup lengkap mulai dari kapasitas memori program dan memori data yang cukup besar, interupsi, timer / counter, PWM, USART, TWI, *analog comparator*, EEPROM internal dan juga ADC internal.

Selain itu kemampuan kecepatan eksekusi yang lebih tinggi yaitu maksimal 16MHz menjadi alasan bagi banyak orang untuk beralih dan lebih memilih menggunakan mikrokontroler jenis AVR ketimbang mikrokontroler pendahulunya yaitu keluarga MSC-51 yang hanya 12MHz. Untuk menggunakan mikrokontroler keluarga AVR, dengan ATmega8535 perlu diperhatikan:

1. Fitur ATmega 8535
2. Deskripsi PIN
3. Arsitektur ATmega 8535
4. Organisasi Memori

Keterangan:

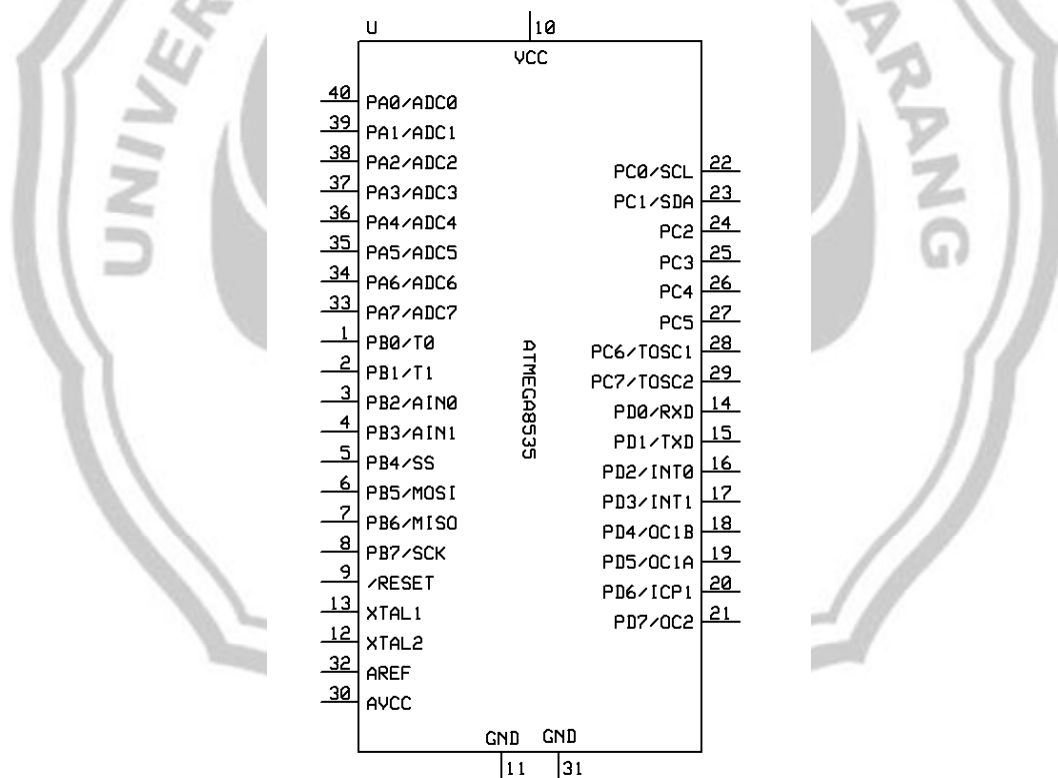
- 1) Fitur ATmega8535

Berikut ini adalah fitur – fitur yang dimiliki oleh ATmega8535 :

- a. 130 macam intruksi, yang hampir semuanya dieksekusi dalam satu siklus clock.
- b. 32 x 8-bit register serba guna.
- c. Kecepatan mencapai 16 MIPS dengan clock 16 MHz.
- d. 8Kbyte Flash memori, yang memiliki fasilitas *In-System Programming*.
- e. 512 Byte internal EEPROM.
- f. 512 Byte SRAM.
- g. Programming Lock, fasilitas untuk mengamankan kode program.

- h. Counter 8-bit dan 1 buah timer / counter 16-bit / 2 buah timer.
- i. Output PWM 4channel.
- j. ADC 10-bit 8 channel.
- k. Serial USART.
- l. Master / Slave SPI serial interface.
- m. Serial TWI atau I2C.
- n. On-Chip Analog Comparator

2) Deskripsi PIN



Gambar 2. Konfigurasi Pin ATmega8535

www.datasheetcatalog.com/ATmega8535

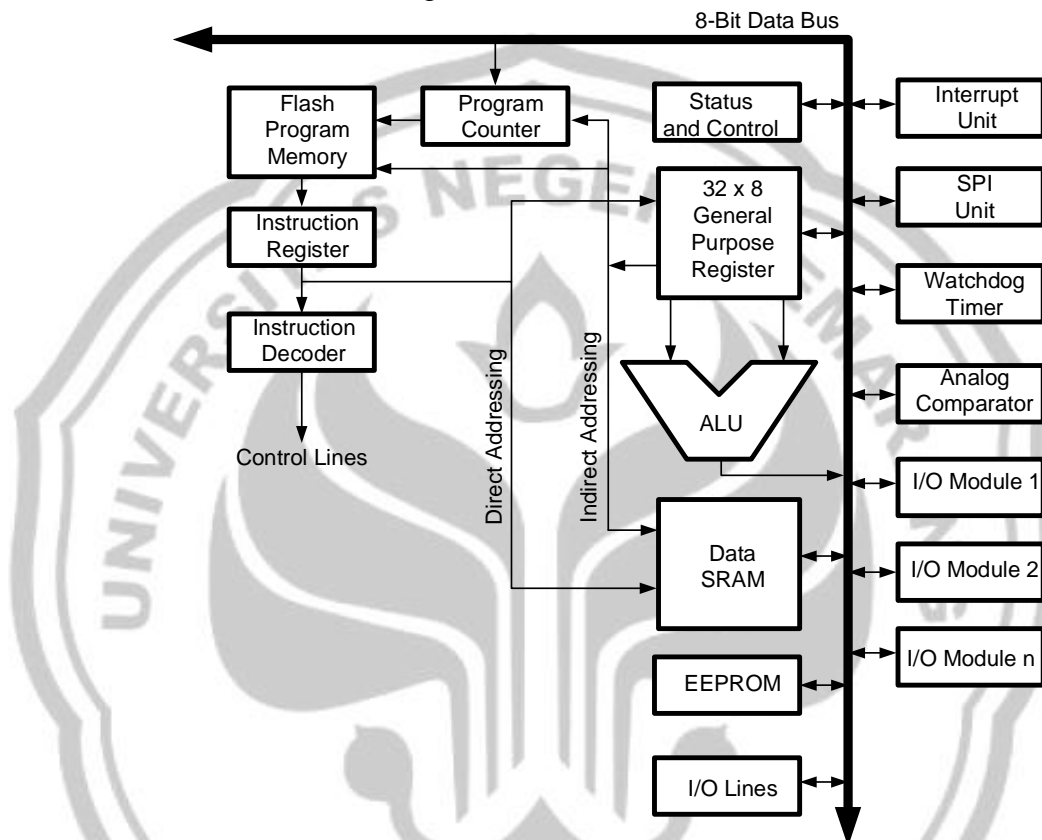
Konfigurasi Pin ATmega8535 gambar 2, secara singkat dapat diterangkan tiap Pin dalam bentuk tabel 1.

Tabel 1. Deskripsi Pin ATmega8535

PIN	KETERANGAN
1-8	<p>Port B, merupakan Port I/O 8-bit dua arah (<i>bit-directional</i>) dengan resistor <i>pull-up</i> internal. Selain sebagai Port I/O 8-bit Port B juga dapat difungsikan secara individu sebagai berikut :</p> <ul style="list-style-type: none"> • PB7 : SCK (<i>SPI Bus Serial Clock</i>) • PB6 : MISO (<i>SPI Bus Master Input / Slave Output</i>) • PB5 : MOSI (<i>SPI Bus Master Output / Slave Input</i>) • PB4 : SS (<i>SPI Slave Select Input</i>) • PB3 : AIN1 (<i>Analog Comparator Negatif Input</i>) OC0 (<i>Output Compare Timer / Counter 0</i>) • PB2 : AIN0 (<i>Analog Comparator Positif Input</i>) INT2 (<i>External Interrupt 2 Input</i>) • PB1 : T1 (<i>Timer / counter 1 External Counter Input</i>) • PB0 : T0 (<i>USART External Clock Input / Output</i>)
9	RESET, merupakan pin reset yang akan bekerja bila diberi pulsa rendah (<i>aktif low</i>) selama minimal 1.5 μ s
10	VCC, catu daya digital
11	GND, Ground untuk catu daya digital
12	XTAL2, merupakan output dari penguat oscillator pembalik
13	XTAL1, merupakan input ke penguat oscillator pembalik dan input ke internal clock.
14-21	<p>Port D, merupakan Port I/O 8-bit dua arah (<i>bi-directional</i>) dengan resistor <i>pull-up</i> internal. Selain sebagai Port I/O 8-bit Port D juga dapat difungsikan secara individu sebagai berikut :</p> <ul style="list-style-type: none"> • PD7 : OC2 (<i>Output Compare Timer / Counter 2</i>) • PD6 : ICP1 (<i>Timer / Counter 1 input Capture</i>) • PD5 : OC1A (<i>Output Compare A Timer / Counter 1</i>) • PD4 : OC1B (<i>Output Compare B Timer / Counter 1</i>) • PD3 : INT1 (<i>External Interrupt 1 Input</i>) • PD2 : INT0 (<i>External Interrupt 0 Input</i>) • PD1 : TXD (<i>USART transmit</i>) • PD0 : RXD (<i>USART receive</i>)
22-29	<p>Port C, merupakan Port I/O 8-BIT dua arah (<i>bi-directional</i>) dengan resistor <i>pull-up</i> internal. Selain sebagai Port I/O 8-bit 4bit Port C juga dapat difungsikan secara individu sebagai berikut :</p> <ul style="list-style-type: none"> • PC7 : TOC2 (<i>Timer Ocillator 2</i>) • PC6 : TOC1 (<i>Timer Ocillator 1</i>) • PC1 : SDA (<i>Serial Data Input / Output, I²C</i>) • PC0 :SCL (<i>Serial Clock, I²C</i>)
30	AVCC, merupakan catu daya yang digunakan untuk masukan analog ADC yang terhubung ke Port A
31	GND, Ground untuk catu daya analog

32	AREF, merupakan tegangan referensi analog untuk ADC
33-40	Port A, merupakan Port I/O 8-bit dua arah (<i>bi-directional</i>) dengan resistor <i>pull-up</i> internal. Selain sebagai Port I/O 8-bit Port A juga dapat berfungsi sebagai masukan 8 channel ADC

3) Arsitektur ATmega8535



Gambar 3. Arsitektur ATmega8535

www.datasheetcatalog.com/ATmega8535

Gambar 3 di atas artinya, Mikrokontroler ATmega8535 memiliki arsitektur Harvard, yaitu memisahkan memori untuk kode program dan memori untuk data sehingga dapat memaksimalkan unjuk kerja. Instruksi-instruksi dalam memori program dieksekusi dalam satu alur tunggal, pada saat satu instruksi dikerjakan

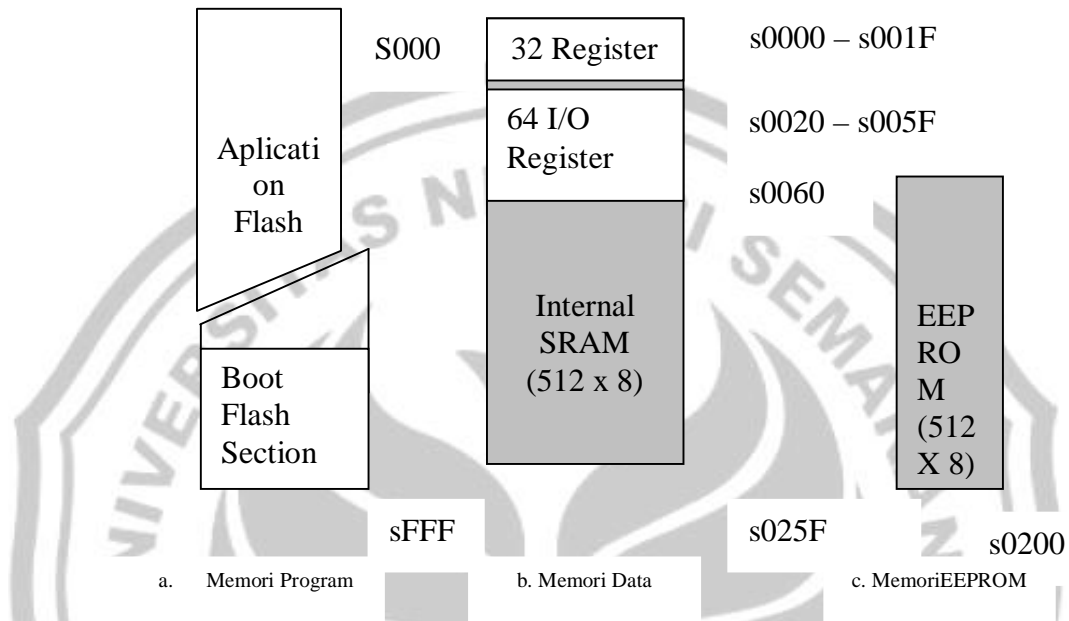
instruksi berikutnya sudah diambil (*pre – fetched*) dari memori program. Konsep inilah yang memungkinkan instruksi – instruksi dapat dieksekusi dalam setiap satu siklus clock.

32 x 8-bit register serba guna digunakan untuk mendukung operasi pada *Aritmetic Logic Unit* (ALU) yang dapat dilakukan dalam satu siklus. 6 dari register serba guna ini dapat digunakan sebagai 3 buah register pointer 16-bit pada mode pengalamatan tak langsung untuk mengambil data pada ruang memori data. Ketiga register pointer 16-bit ini disebut dengan register X (gabungan R26 dan R27), register Y (gabungan R28 dan R29), dan register Z (gabungan R30 dan R31).

Hampir semua instruksi AVR memiliki format 16-bit (*word*). Setiap alamat memori program terdiri dari instruksi 16-bit atau 32-bit. Selain register serba guna diatas, terdapat register lain yang terpetakan dengan teknik *memory mapped I/O* selebar 64 byte. Beberapa register ini digunakan untuk fungsi khusus antara lain sebagai register kontrol Timer / Counter, Interupsi, ADC, USART, SPI, EEPROM dan fungsi I/O lainnya. Register – register ini menempati memori pada alamat 0x20h – 0x5Fh.

4) Organisasi Memori

Mikrokontroler ATmega8535 memiliki 3 jenis memori yaitu memori program, memori data, dan memori EEPROM. Ketiganya memiliki ruang sendiri dan terpisah seperti terlihat pada gambar 4:



Gambar 4. Organisasi Memori ATmega8535

Pada Gambar 4, berikut adalah keterangan dari masing-masing bagian:

1) Memori Program

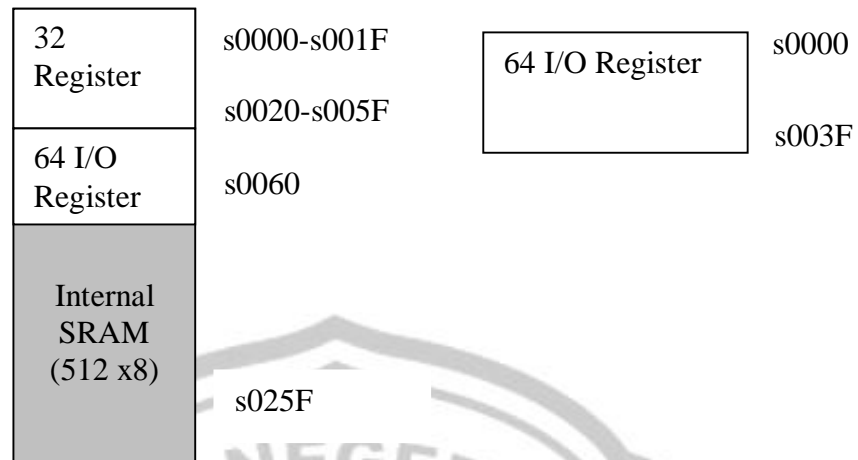
ATmega8535 memiliki kapasitas memori program sebesar 8 Kbyte yang terpetakan dari alamat 0000h – 0FFFh dimana masing – masing alamat memiliki lebar data 16 bit. Sehingga organisasi memori program seperti ini sering dituliskan dengan 4K x 16 bit. Memori program ini terbagi menjadi 2 bagian yaitu bagian program *boot* dan bagian program aplikasi. Jika kita tidak menggunakan fitur *Boot*

Loader Flash maka semua kapasitas memori program diatas dapat digunakan untuk program aplikasi. Tetapi jika kita menggunakan fitur *Boot Loader Flash* maka pembagian ukuran kedua bagian ini ditentukan oleh *BOOTSZ fuse*.

2) Memori Data

ATMega8535 memiliki kapasitas memori data sebesar 608 Byte yang terbagi menjadi 3 bagian yaitu register serba guna, register I/O dan SRAM seperti terlihat pada gambar dibawah ini. 32 byte alamat terendah digunakan untuk register serba guna yaitu R0 – R31. 64 byte berikutnya digunakan untuk register I/O yang digunakan untuk mengatur fasilitas seperti timer / counter, interupsi, ADC, USART, SPI, EEPROM dan port I/O seperti Port A, Port B, Port C, Port D. Selanjutnya 512 Byte diatasnya digunakan untuk memori data SRAM.

Jika register – register I/O diatas diakses seperti mengakses data pada memori (menggunakan instruksi LD atau ST) maka register I/O diatas menepati alamat 0020 – 005F seperti terlihat pada gambar dibawah ini. Tetapi jika register – register I/O diatas diakses seperti mengakses I/O pada umumnya (menggunakan instruksi IN atau OUT) maka register I/O diatas menepati alamat memori 0000h – 003Fh seperti terlihat pada gambar 5.



Gambar 5. Register I/O sebagai Memori Data dan Sebagai I/O

3) Memori EEPROM

ATMega8535 memiliki memori EEPROM sebesar 512 Byte yang terpisah dari memori program maupun memori data. Memori EEPROM ini hanya dapat diakses dengan menggunakan register – register I/O yaitu register EEPROM Address (EEARH-EEARL), register EEPROM Data (EEDR) dan register EEPROM Control (EERC). Untuk mengakses data eksternal sehingga waktu eksekusinya relatif lebih lama bila dibandingkan dengan mengakses data dari SRAM.

3. Liquid Crystal Display (LCD M1632)

LCD (*Liquid Crystal Display*) merupakan perangkat display yang paling umum dipasangkan ke mikrokontroler, mengingat ukurannya yang kecil dan kemampuan menampilkan karakter atau grafik yang lebih baik

dibandingkan display *7 segment* ataupun *alphanumeric*. Pada pengembangan sistem *embedded*, LCD mutlak diperlukan sebagai sumber pemberi informasi utama, misalnya alat pengukur kadar gula darah, penampil jam, penampil *counter* putaran motor industri dan lainnya.

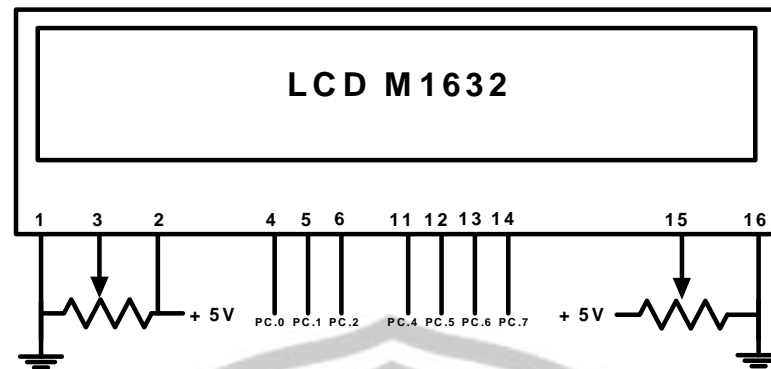
LCD adalah modul penampil yang banyak digunakan karena tampilannya menarik. LCD yang paling banyak digunakan saat ini adalah LCD M1632 karena harganya cukup murah. LCD M1632 merupakan modul LCD dengan tampilan (2 baris x 16 kolom) dengan konsumsi daya rendah (Widodo Budiharto 2007: 43 – 44).

Untuk menghubungkan LCD M1632 dengan Mikrokontroler ATmega8535 dapat mengikuti tabel 2 dan mengikuti rangkaian pada gambar 6.

Tabel 2. Koneksi Antara Modul LCD dengan Mikrokontroler

Pin LCD	Keterangan	Pin Mikrokontroler	Keterangan
1	GND	11	GND
2	+5 V	10	VCC
4	RS	22	Port C.0
5	RD	23	Port C.1
6	EN	24	Port C.2
11	D4	26	Port C.4
12	D5	27	Port C.5
13	D6	28	Port C.6
14	D7	29	Port C.7

Code Vision AVR telah menyediakan fungsi pustaka yang khusus menangani akses modul LCD yaitu *lcd.h*.



Gambar 6. Rangkaian Koneksi LCD dengan Mikrokontroler

B. Perakitan

Sesuai dengan tujuan perencanaan ini adalah merancang alat pengendali kipas angin yang bekerja otomatis dan mengganti kipas angin dengan sistem kerja manual dengan menggunakan sensor suhu berbasis mikrokontroler. Cara yang digunakan adalah metode eksperimen laboratoris. Langkah – langkah tersebut meliputi perancangan , pembuatan, pengujian alat dan analisis kerja alat.

1. Metode Laboratoris

Metode eksperimen yang digunakan dalam penelitian ini adalah eksperimen sekali tembak atau sering disebut *one shot case study*. Metode eksperimen ini mempunyai pola X,O dan X adalah perlakuan dan O adalah tes akhir. Di bawah ini adalah bentuk metode eksperimen. (Arikunto, Suharsimi 2002: 77)

X	O
Perencanaan dan pembuatan	Tingkat keberhasilan

2. Instrumen

Instrumen adalah alat ukur yang digunakan untuk pengukuran dalam eksperimen dan harus mempunyai tingkat validitas yang tinggi, artinya sebuah instrumen dikatakan valid apabila mampu mengukur secara tepat atau mendekati harga sesungguhnya. Selain valid, sebuah instrumen juga harus mempunyai tingkat realibilitas yang baik. Instrumen hanya dapat dipercaya bila data yang diperoleh sesuai dengan kenyataan. Dalam tugas akhir ini digunakan alat ukur Digital Multimeter dengan spesifikasi SANWA Type CD721.

3. Teknik Analisis Data

Pengukuran unjuk kerja alat ini bertujuan untuk mengetahui kualitas kerja apakah alat ini dapat bekerja sesuai dengan harapan dalam perancangan atau tidak.

Teknik analisis data disini menggunakan metode analisis diskriptif yaitu membandingkan antara perhitungan perencanaan dengan pengukuran atau pengamatan hasil eksperimen. Apabila terjadi penyimpangan dilakukan identifikasi dari penyimpangan tersebut.

C. Pembuatan Alat

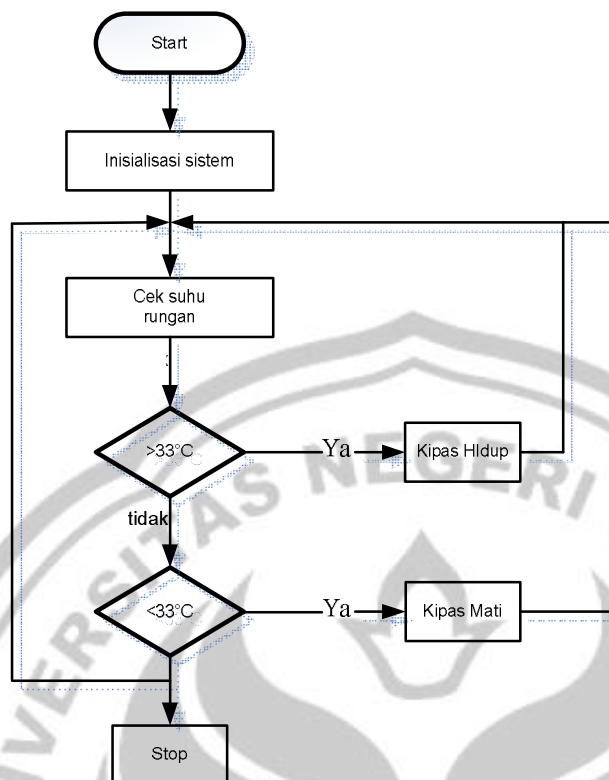
1. Pembuatan Perangkat Lunak (*Software*)

Sebelum membuat program dibutuhkan pemahaman urutan kerja dari alat yang akan dibuat, sehingga program yang dibuat

berurutan sesuai dengan *flowchart* yang dibuat. Berikut adalah perencanaan pembuatan perangkat lunak yang harus dikerjakan:

- Pertama-tama mikrokontroler menginisialisasi port-port yang akan digunakan untuk keperluan pembacaan sensor dan port untuk menampilkan ke LCD.
- Setelah selesai inisialisasi maka sensor LM35 sudah dapat mengirimkan data ke mikrokontroler.
- Data output LM35 yang berupa tegangan akan dikirimkan ke ADC internal yang dimiliki oleh mikrokontroler ATmega8535.
- Data yang telah diterima mikrokontroler melalui ADC akan diolah dengan perumusan tertentu agar nilainya dapat dikonversi menjadi satuan derajat Celcius.
- Akan dilakukan beberapa syarat yang dilakukan berdasarkan suhu yang diperoleh.
- Hasil yang berupa derajat Celcius ini akan ditampilkan ke LCD (*Liquid Crystal Display*).

Berdasarkan pembuatan perangkat lunak di atas, gambar 7 berikut adalah *flowchart* programnya;



Gambar 7. Flowchart Program

a. Membuat Daftar Input dan Output Mikrokontroler

Alamat input dan output mikrokontroler yang digunakan ditunjukkan pada tabel berikut ini.

Tabel 3. Daftar Alamat Input

No	Alamat Input	Keterangan
1.	Timer 0	225 detik

Tabel 4. Daftar Alamat Output

No	Alamat Output	Keterangan
1.	Port D 4, Port D 5, Port D 6, Port D 7	Motor Stepper
2.	Port C 0 – Port C 7	LCD M1632

b. Pembuatan Program Menggunakan Program AVR.

Menerjemahkan *flowchart* ke dalam program bahasa C disesuaikan dengan daftar I/O yang telah dibuat. *Listing* Program kendali bidang solar sel dapat dilihat pada halaman lampiran.

c. Memasukan Program ke Mikrokontroler.

Memasukan program ke dalam mikrokontroler dilakukan menggunakan *Personal Computer*(PC) melalui program *CodeVision AVR Compiler* . Sebelum memasukan program ke mikrokontroler, pastikan mikrokontroler tersebut mendapatkan catu daya yang memadai. Karena jika mikrokontroler tidak mendapatkan catu daya yang memadai maka program tidak mengisi ke mikrokontroler.

Langkah – langkah memasukan program ke mikrokontroler adalah sebagai berikut :

- a. Buka program *CodeVision AVR* dengan cara **Start Menu → All Program → CodeVision AVR → CodeVision AVR C Compiler**
- b. Untuk membuat project baru gunakan **File → New** maka akan muncul 2 pilihan yaitu source atau project. Karena kita akan membuat project dulu maka → pilih **Project**.
- c. Setelah itu akan muncul konfirmasi apakah kita akan menggunakan *CodeWizard AVR* atau tidak. Pilih *Yes* jika ingin menggunakan fasilitas *CodeWizard AVR* atau *No* jika tidak. Fasilitas *CodeWizard AVR* digunakan untuk membangkitkan kode program secara

otomatis sesuai dengan menu –menu yang telah tersedia. Untuk project ini kode program akan ditulis sendiri sehingga pilih *No*.

- d. Tuliskan dan simpan nama file project. Misal **Stepper.prj**
- e. Pilih Tab **C Compiler** kemudian pilih chip sesuai dengan IC mikrokontroller yang dipakai serta atur clock sesuai dengan frekuensi kristal yang digunakan.
- f. Langkah selanjutnya adalah membuat file baru untuk menuliskan *listing* programnya yaitu dengan cara : **File → New → pilih Source**. Setelah itu akan muncul jendela baru yang dapat digunakan untuk menulis program. Tuliskan program kemudian simpan dengan cara **File → save** dan beri nama, misalnya **Half.c**
- g. Selanjutnya file program ini (*Half.c*) harus dimasukkan ke dalam file project yang telah dibuat sebelumnya (*Stepper.prj*) dengan cara **Project → Configure → pilih Tab Files → Add** pilih file **Half.c → Open**
Sekarang file program *Half.c* telah masuk dalam project.
- h. Langkah berikutnya adalah proses kompilasi. Untuk mengkompilasi program yang telah dituliskan tadi dilakukan dengan cara :
Project → Make
Apabila dalam proses mengkompilasi ini terdapat error maka cari sumber error tersebut lalu betulkan sampai tidak terdapat error lagi.
- i. Proses selanjutnya adalah mendownload program ke mikrokontroller. Caranya adalah :

Buka program *CodeVision AVR Chip Programmer*. Caranya adalah **Tools | Chip Programmer**. Jangan lupa pastikan settingan programmernya pada modul STK200. Untuk mensetting programmernya caranya adalah **Settings | Programmers** kemudian pilih AVR Chip Programmer Type : Kanda System STK200+/300.

j. Buka file program yang akan didownload ke mikrokontroler caranya adalah **File | Load FLASH** kemudian pilih nama file **Half.hex**.

k. Download program ke mikrokontroler. Caranya adalah **Program | All**

2. Pembuatan Perangkat Keras (*Hardware*)

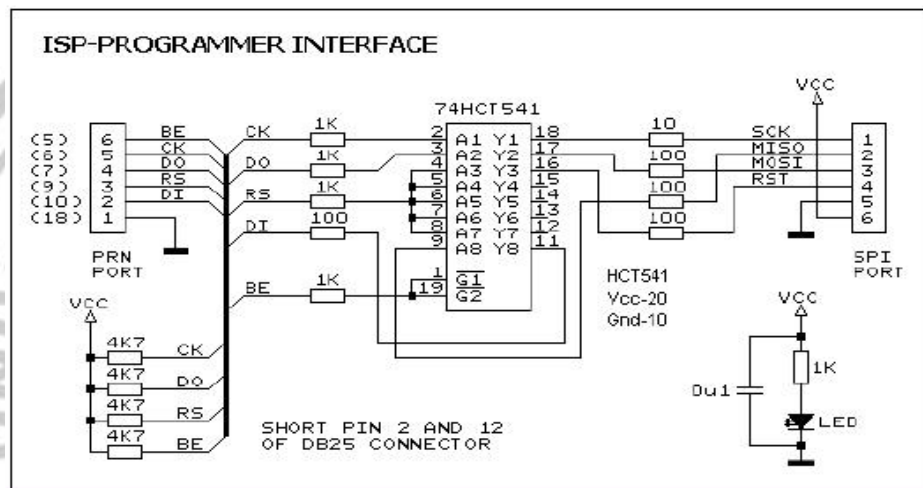
a. Sistem Minimum ATmega8535

Sistem minimum ATmega8535 menggunakan kristal sebagai pembangkit clock yang dibutuhkan untuk detak mikrokontroler ATmega8535. Kristal yang digunakan adalah 8 MHz. Selain kristal ada juga komponen-komponen pendukung lainnya misalnya kapasitor keramik 15 pF, Elco 1000 uF / 25 V, Induktor 10 mH, dan lainnya.

b. Pembuatan Downloader

Downloader disini digunakan untuk memasukan program yang telah dibuat ke mikrokontroler. Banyak sekali IC downloader yang bisa digunakan misalnya 74LS541, 74HC125, 74LS125 dan 74HCT541. Sebenarnya bisa saja tidak menggunakan IC Buffer diatas

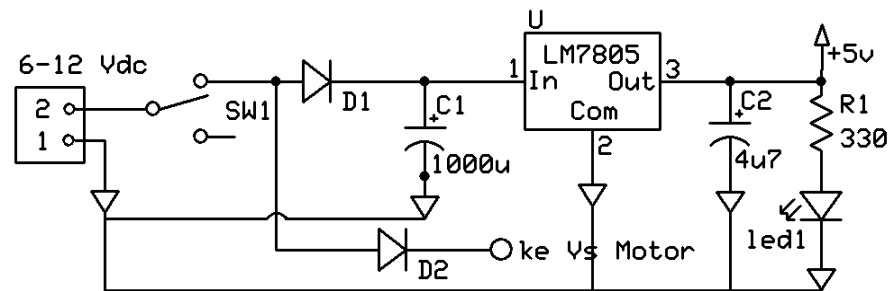
tetapi mengandung resiko yang merepotkan, misal terjadi hubung singkat pada rangkaian mikrokontroller maka bisa saja merembet ke komputer sehingga kemungkinan akan menyebabkan port paralel komputer yang digunakan menjadi rusak. Dalam pembuatan tugas akhir ini digunakan downloader dengan menggunakan IC 74HCT541 dengan rangkaian gambar 8.



Gambar 8. ISP Programmer interface Type Kanda System STK 200+/300

c. Rangkaian Catu Daya

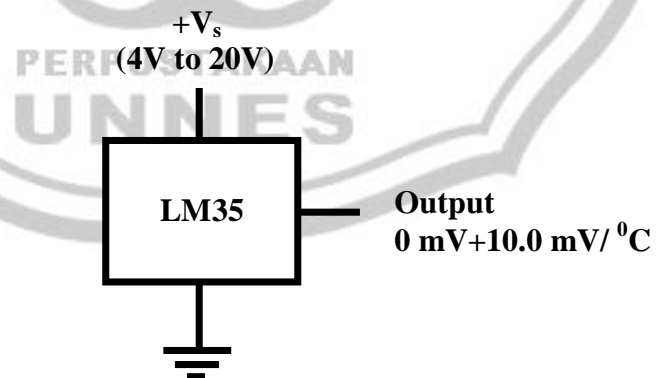
Catu daya 5 V menggunakan transformator step down 2A tanpa CT (*center tap*). Untuk penyearah digunakan penyearah gelombang penuh dengan menggunakan model dioda *bridge* dengan kapasitor sebagai filter. Agar tegangan keluaran menjadi 5 V maka digunakan IC regulator 7805, untuk membatasi tegangan dari 12 VDC menjadi 5 VDC.



Gambar 9. Rangkaian Catu daya

d. Perakitan Sensor Suhu LM35

Sensor LM35 memiliki tegangan kerja 5 Volt namun outputnya hanya antara 0,01V sampai 1,00V mengingat LM35 yang digunakan adalah dari seri DZ sehingga *range* pengukuran hanya berkisar antara 0-100 °C dengan perubahan sebesar 10mV per 1C. Dengan ketelitian yang dimiliki maka sensor tersebut dapat diterapkan langsung dengan mikrokontroler ATmega8535 yang memiliki ADC internal 10 bit.



Gambar 10. Koneksi LM35

e. Relay

Relay adalah suatu rangkaian *switch magnetic* yang bekerja bila mendapat catu dan suatu rangkaian *trigger*. Relay memiliki tegangan dan arus nominal yang harus dipenuhi output rangkaian pendriver atau pengemudinya. Arus yang digunakan pada rangkaian adalah arus DC (*Direct Current*).

Relay terdiri dari lilitan kawat (*coil*) yang dililitkan pada inti besi lunak. Jika kawat mendapatkan medan magnet dan menarik *switch* kontak. *Switch* kontak mengalami gaya listrik magnet sehingga berpindah posisi ke kutub lain atau terlepas dari kutub asalnya. Keadaan ini akan bertahan selama arus mengalir pada kumparan relay. Dan relay akan kembali ke posisi semula yaitu *normaly ON* atau *normaly OFF*, bila tiada arus yang mengalir maka posisi normal relay tergantung pada jenis relay yang termakan. Dan pemakaian jenis relay tergantung pada keadaan yang diinginkan pada suatu rangkaian. Menurut kerjanya relay dapat dibedakan menjadi :

- a. *Normaly Open* (NO), saklar akan terbuka bila dialiri arus.
- b. *Normaly Close* (NC), saklar akan tertutup bila dialiri arus.
- c. *Change Over* (CO), relay ini mempunyai saklar tunggal yang normalnya tertutup lam, bila kumparan satu dialiri arus maka saklar akan terhubung ke terminal A, sebaliknya bila kumparan dua dialiri arus maka saklar terhubung ke terminal B.

Relay yang digunakan adalah basis transistor yang dialiri oleh arus dari kolektor ke emiter yang mengakibatkan relay terhubung. Fungsi dioda pada rangkaian adalah untuk melindungi transistor dari tegangan induksi berlebih dimana tegangan ini dapat merusak transistor.

Jika transistor pada basis tidak ada arus maju maka transistor terbuka sehingga arus tidak mengalir dari kolektor ke emiter, relay tidak bekerja karena tidak ada arus yang mengalir pada gulungan kawat.

3. Pembuatan Pesawat Simulasi

Secara umum, pembuatan alat ini melalui beberapa tahap yaitu: Pembuatan layout PCB (*printed circuit board/papan rangkaian tercetak*) dan pemasangan komponen.

a. Pembuatan Layout PCB

Langkah – langkah dalam pembuatan PCB adalah sebagai berikut :

- a. Buatlah desain layout rangkaian pada kertas.
- b. Menentukan tata letak kaki komponen yang akan digambar.
- c. Menggambar jalur layout pada komputer dengan menggunakan *software Exspress PCB*.
- d. Mengeprint hasil layout

b. Teknik Sablon PCB

1. Setelah layout jadi dan diprint, layout di fotocopy ke dalam mika transparansi dengan ketebalan perbandingan 100 : 1.
2. Sebelum disetrika alangkah baiknya bila papan CCB(*Copper Cload Board*) yang akan digunakan dicuci menggunakan

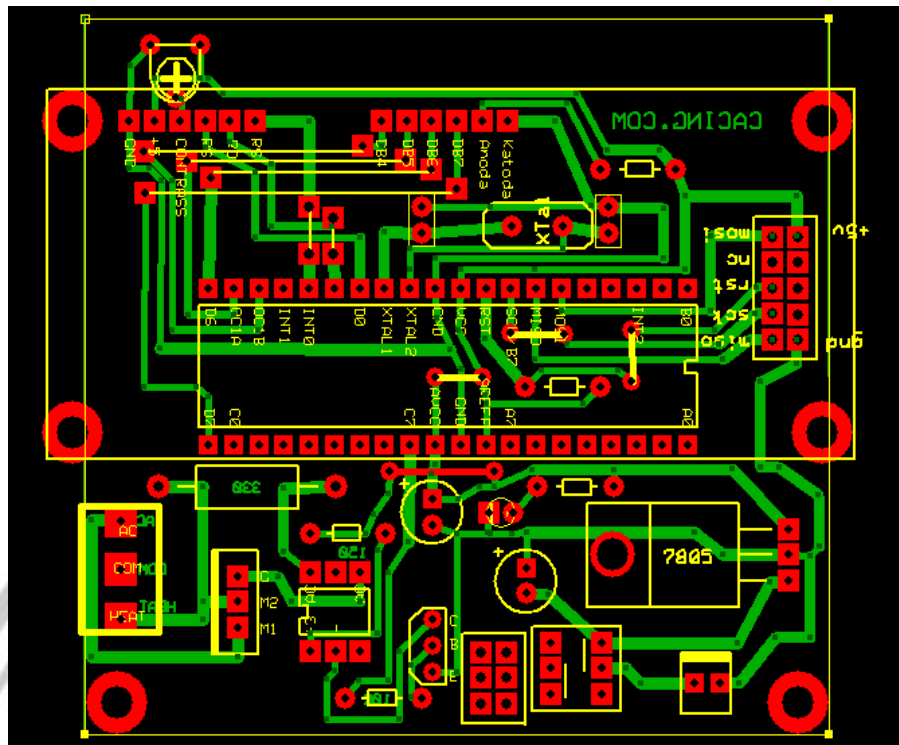
bensin untuk menghilangkan debu – debu yang masih menempel pada lapisan tembaga sehingga tinta fotocopy bisa menempel saat disetrika.

3. Menyetrika hasil fotocopy ke papan CCB selama kurang lebih 30 menit.
4. Setelah disetrika, biarkan dingin terlebih dahulu selama kurang lebih 1 jam untuk mendapatkan hasil yang bagus.
5. Setelah dingin, lepas mika, jika ada jalur yang menempel pada mika, jangan disetrika lagi karena jalur pada PCB akan kembali ke mika. Untuk itu tebalkan jalur yang belum tersetrika menggunakan spidol *water proof*.

c. Pelarutan dan Pengeboran

1. Mempersiapkan larutan *ferry chloride* ($FeCl_3$).
2. Memasukan papan CCB yang telah disablon ke dalam larutan *ferry chlorida* ($FeCl_3$).
3. Setelah lapisan tembaga yang tidak tersablon larut, PCB diangkat dari larutan dan bersihkan dengan menggunakan bensin.
4. Menitik *pads* dengan paku lancip agar sewaktu proses pengeboran tidak meleset.
5. Bor *pads* yang telah dititik.

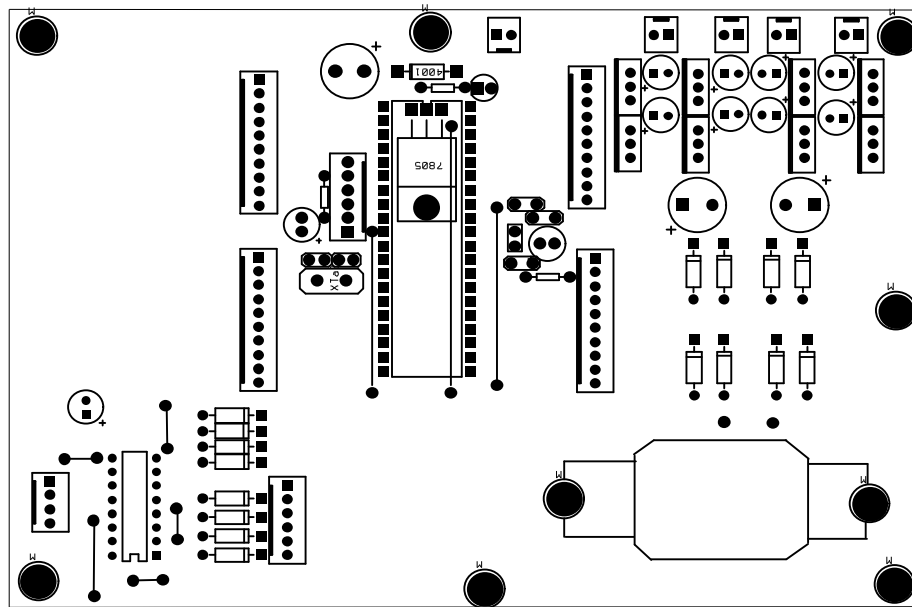
Gambar 11 memperlihatkan gambar lay out PCB yang sudah dibuat.



Gambar 11. Lay Out PCB

d. Pemasangan Tata Letak Komponen

Semua komponen dipasang pada PCB yang telah dibuat dengan memperhatikan tata letak komponennya. Pemasangan komponen dilakukan secara bertahap yaitu dipasang perbagian sehingga dapat langsung dilakukan pengecekan. Teknik penyolderan komponen pada PCB harus benar sehingga komponen dapat terpasang dengan sempurna. Usahakan dalam penyolderan jangan terlalu lama karena komponen yang disolder bisa mati atau rusak. Hasil penyolderan yang baik adalah tenol lancip dan tidak berlubang pada kaki komponen yang disolder serta tidak terlalu banyak tenol yang terpakai.



Gambar 12. Tata Letak Komponen Rangkaian Sistem Minimum ATmega8535,
Driver Motor Stepper, Catu Daya

D. Hasil Pengujian

1. Pengujian Rangkaian Catu Daya

Pengujian rangkaian catu daya ini bertujuan untuk mengetahui tegangan yang dikeluarkan oleh rangkaian. Tabel 5 merupakan hasil pengukuran tegangan keluaran yang dihasilkan rangkaian catu daya yang dilakukan sampai 4X agar didapatkan hasil yang tepat. Pengujian dilakukan di satu titik tegangan dengan waktu yang berbeda.

Tabel 5. Hasil Pengukuran Pengujian Rangkaian Catu Daya

Tegangan		
Urutan	Yang diharapkan (V)	Yang dihasilkan (V)
1	5	4,99
2	5	5,01
3	5	5,04
4	5	4,98
Rata - rata		5,005

Tidak sesuainya tegangan keluaran seperti yang diharapkan tersebut dikarenakan beberapa faktor diantaranya adalah kualitas bahan dari tiap – tiap komponen yang digunakan sudah tidak murni lagi atau adanya nilai toleransi dari komponen tersebut.

2. Pengujian Rangkaian LCD

Pengujian LCD ini untuk mengetahui apakah LCD yang dipakai masih bagus atau sudah rusak. Pengujian pertama yang dilakukan adalah dengan memberi tegangan pada kaki catu (kaki 2 dan 15), namun jika menyala, pengujian selanjutnya adalah memeberikan program ke sistem minimum ATmega8535 yang akan ditampilkan oleh LCD melalui port C sebagai contoh menampilkan Tulisan “ELEKTRO UNNES”.

```
#include <mega8535.h>
#include <lcd.h>
#asm.equ_lcd_port=0x15;Port C
#endasm

Main()
{
  lcd_init(16);
  Lcd_clear();
  Lcd_gotoxy(0,0);
  Lcd_putsf(“ELEKTRO UNNES”);
}
```

3. Pengujian Kerja Alat

Dalam pengujian kerja dari rangkaian LM35 yang diaplikasikan pada kipas angin dengan tampilan digital ini, bahwa suhu pada saat kalibrasi hampir dikatakan sangat baik karena misalnya ketika suhunya

30⁰C, keluaran dari rangkaian LM35 adalah sebesar 0,3 V. Hasil perbandingan pengukuran tersebut dapat ditunjukkan pada tabel 6 di bawah ini :

Tabel 6. Perbandingan Suhu Kalibrasi Dengan Tegangan Output LM35

Suhu Kalibrasi (Celcius)	Output Rangkaian LM35 (V)
28	28,64
29	29,00
30	30,04
31	31,00
32	31,96
33	33,00

Waktu pengukuran suhu kalibrasi menggunakan thermometer air raksa 0-50 °C.

E. Pembahasan

1. Analisis Perangkat Lunak

Perangkat lunak mikrokontroler dalam bentuk program menggunakan bahasa C dapat berfungsi sesuai dengan yang direncanakan. Kondisi dari fungsi – fungsi logika dan timer program yang digunakan dapat dilihat pada *listing* program bagian lampiran.

2. Analisis Kerja Alat.

Dalam perencanaan pembuatan rangkaian LM35 yang diaplikasikan ke kipas angin dengan tampilan digital menggunakan LCD dikatakan sangat baik. Karena misalnya ketika suhu pada saat kalibrasi dengan termometer alkohol sebesar 30 ⁰C keluaran dari rangkaian LM35 adalah

sebesar 0,3 V, kemudian pada saat suhu 100 °C keluaran dari rangkaian LM35 adalah sebesar 1 °C, dan juga pada saat suhu 110 °C keluaran dari rangkaian LM35 adalah 1,1 °C.

Keakuratan pengukuran dari pembuatan alat rangkaian LM35 dapat dirumuskan sebagai berikut :

$$\frac{\text{Suhu kalibrasi}}{\text{Suhu dari output LM35}} \times 100 \%$$

Misal :

$$\frac{30}{30} \times 100 \%$$

$$= 100 \%$$

Jadi dari uraian di atas, rangkaian sensor LM35 tidak memiliki nilai keerroran.

F. Deskripsi Kerja Alat

Pengoperasian Alat Otomatisasi Kipas Angin dengan menggunakan sensor LM35 berbasis mikrokontroler ATmega8535 ini, akan mengukur suhu udara di dalam ruangan dengan menggunakan sensor LM35 untuk mengukur suhu dalam satuan derajat *Celcius*. Ketika sensor telah mendapatkan suhu yaitu dengan menerjemahkan sifat fisis suhu menjadi sinyal listrik yaitu perubahan tegangan *output* sensor, maka kemudian output ini dibaca oleh ADC internal dari mikrokontroler ATmega8535 dan kemudian data

dikalkulasikan dengan rumusan tertentu sehingga pada tahap berikutnya sistem dapat menentukan apakah suhu dalam ruangan sudah sesuai atau belum. Jika suhu ruangan sudah mencapai suhu yang sudah ditentukan pada sistem program maka kipas angin akan hidup sendiri. Nilai yang sedang dipantau oleh sensor akan ditampilkan ke *display* LCD.



BAB 3

PENUTUP

A. Kesimpulan

Berdasarkan hasil merakit Simulator Saklar Otomatis Kipas Angin Menggunakan Sensor Suhu dan pembahasan dapat disimpulkan:

1. Sensor suhu LM35 mendeteksi suhu dalam ruangan, setelah lebih dari 33°C kipas berputar dan jika kurang dari 33°C kipas berhenti berputar.
2. Alat Otomatisasi Kipas Angin Menggunakan Sensor Suhu dapat digunakan di dalam ruangan sebagai penyetel kondisi ruangan yang bekerja secara otomatis.

B. Saran

Dalam perjalanan pengerjaan proyek akhir ini tentunya tidak lepas dari berbagai macam kekurangan dan kelemahan. Oleh karena itu demi kesempurnaan tugas akhir ini, saran yang dianjurkan dalam proyek tugas akhir ini adalah, apabila ingin membuat alat yang sama, baiknya sensor suhu diatur sebesar 27°C kipas angin hidup, dan 25°C kipas angin mati. Suhu ruangan dalam penggunaan ilmiah, dianggap kurang lebih antara 20°C sampai 25°C.

DAFTAR PUSTAKA

Agus, Bejo. 2008. **C&AVR Rahasia Kemudahan Bahasa C dalam Mikokontroler ATmega8535**. Yogyakarta : GRAHA ILMU.

Lingga, Wardana. 2006. **Belajar Sendiri Mikrokontroller AVR Seri ATmega16, Simulasi Hardware dan Aplikasi**, Yogyakarta: Andi.

Widodo, Budiharto. 2007. **Panduan Praktikum Mikrokontroler AVR ATmega16**. Jakarta : PT. Elex Media Komputindo.

Winoto, A. 2008. **Mikrokontroler AVR ATmega8/32/16/8535 dan Pemrogramannya dengan Bahasa C pada WinAVR**. Bandung: Informatika

<http://go2.wordpress.com> (Diakses 13 Juli 2010).

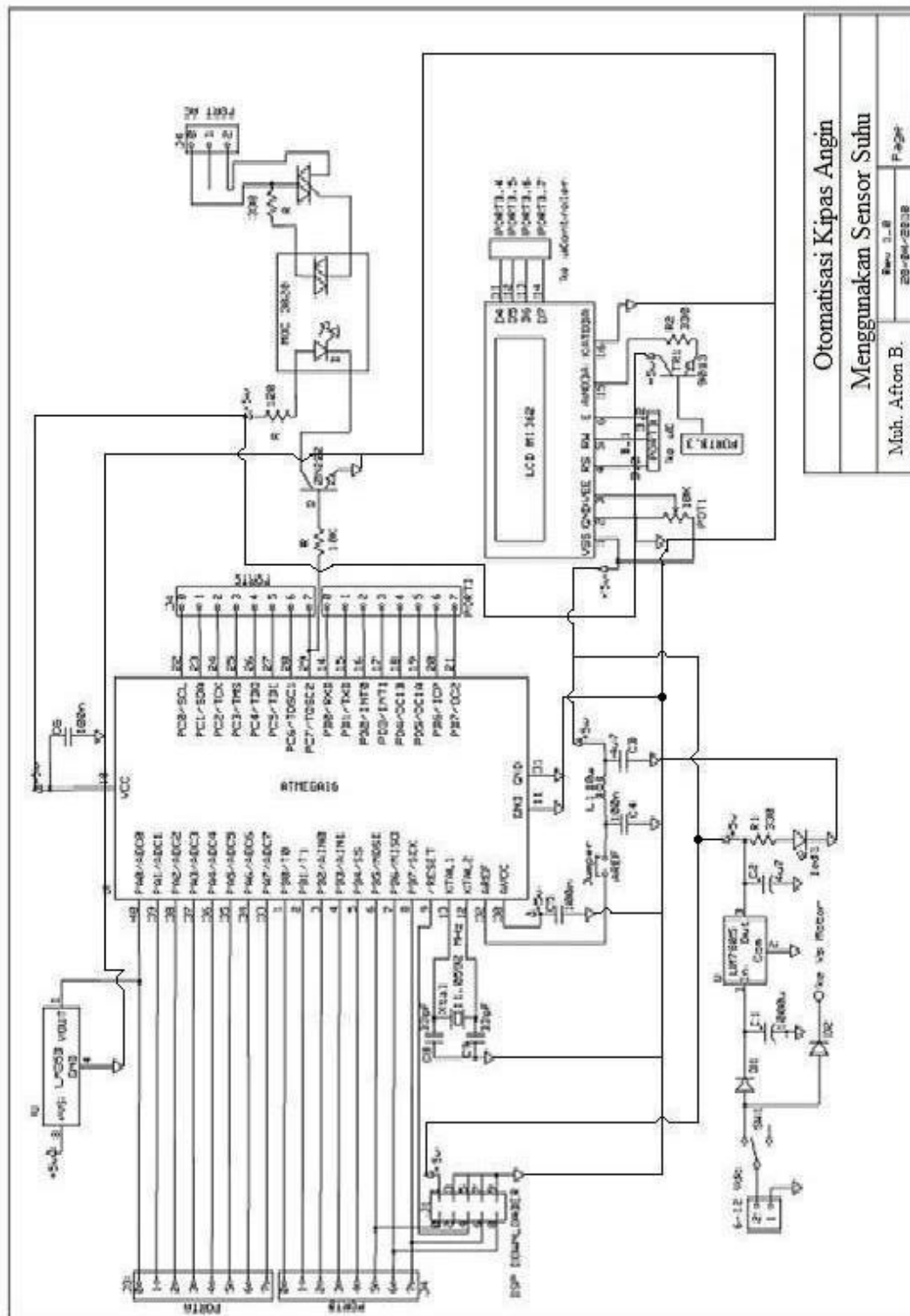
<http://sensorindo.com> (Diakses 13 Juli 2010).

<http://sensorindo.itrademarket.com> (Diakses 13 Juli 2010).

www.juraganelektro.com (Diakses 13 Juli 2010).

www.lain-lain.iklanmax.com (Diakses 13 Juli 2010).

PERPUSTAKAAN
UNNES



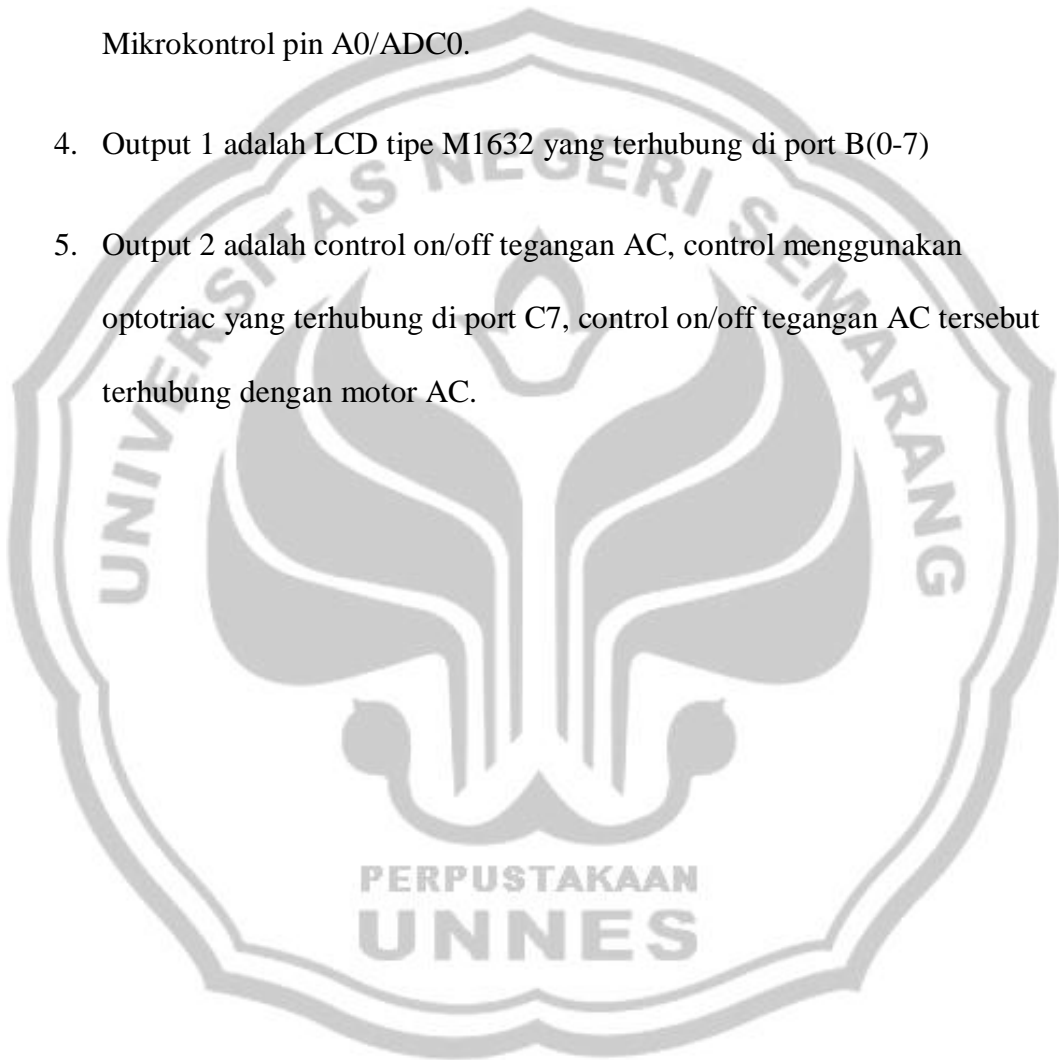
Otomatisasi Kipas Angin
Menggunakan Sensor Suhu
 Moh. Afton B. Revisi 3.08
28-04-2018 Page

Gambar 13. Rangkaian Otomatisasi Kipas Angin Menggunakan Sensor Suhu

Keterangan gambar rangkaian Otomatisasi Kipas Angin Menggunakan

Sensor Suhu di atas adalah;

1. Rangkaian catu daya, adaptor 12V 4A membatasi tegangan dari 12V di outputkan menjadi 5V.
2. Mikrokontroler menggunakan tipe ATmega16.
3. Input sensor suhu menggunakan sensor suhu LM35D dan terhubung di Mikrokontrol pin A0/ADC0.
4. Output 1 adalah LCD tipe M1632 yang terhubung di port B(0-7)
5. Output 2 adalah control on/off tegangan AC, control menggunakan optotriac yang terhubung di port C7, control on/off tegangan AC tersebut terhubung dengan motor AC.



Listing Program Kontrol Suhu

/*****

This program was produced by the
 CodeWizardAVR V1.24.0 Professional
 Automatic Program Generator
 © Copyright 1998-2003 HP InfoTech s.r.l.
<http://www.hpinfotech.ro>
 e-mail:office@hpinfotech.ro

Project :
 Version :
 Date : 7/20/2010
 Author : F4CG
 Company : F4CG
 Comments:

Chip type : ATmega8535
 Program type : Application
 Clock frequency : 11.059200 MHz
 Memory model : Small
 External SRAM size : 0
 Data Stack size : 128

*****/

```
#include <mega8535.h>
#include <delay.h>
#include <stdio.h>
float hasil;
unsigned char lcd_bufferx[33];
float temp;
typedef unsigned char byte;

unsigned char led=0;
unsigned char leda=0;
int data;
int jeda;
unsigned char kata[16];
```

```
unsigned char buff[30];
unsigned long milidetik=0;
unsigned int detik=0;
unsigned int menit=0;
unsigned int jam=0;
```

```
unsigned char xdata[30];
```

```

unsigned int i;

// Alphanumeric LCD Module functions
#asm
    .equ __lcd_port=0x12
#endasm
#include <lcd.h>

#define ADC_VREF_TYPE 0x00
// Read the AD conversion result
unsigned int read_adc(unsigned int adc_input)
{
    ADMUX=adc_input|ADC_VREF_TYPE;
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}

// Declare your global variables here
////////////////////////////////////

/* table for the user defined character
   arrow that points to the top right corner */
flash byte char0[8]={
    0b1100000,
    0b0011000,
    0b0000110,
    0b1111111,
    0b1111111,
    0b0000110,
    0b0011000,
    0b1100000};

char lcd_buffer[33];

/* function used to define user characters */
void define_char(byte flash *pc,byte char_code)
{
    byte i,a;
    a=(char_code<<3) | 0x40;
    for (i=0; i<8; i++) lcd_write_byte(a++,*pc++);
}

```

```

////////////////////////////////////
void tampil(unsigned char dat)
{
    unsigned char data;

    data = dat / 100;
    data+=0x30;
    lcd_putchar(data);

    dat%=100;
    data = dat / 10;
    data+=0x30;
    lcd_putchar(data);

    dat%=10;
    data = dat + 0x30;
    lcd_putchar(data);
}
////////////////////////////////////
////////////////////////////////////
void tampilx(unsigned int datx)
{
    unsigned int datax;

    datax = datx / 1000;
    datax+=0x30;
    lcd_putchar(datax);

    datx%=1000;
    datax = datx / 100;
    datax+=0x30;
    lcd_putchar(datax);

    datx%=100;
    datax = datx / 10;
    datax+=0x30;
    lcd_putchar(datax);

    datx%=10;
    datax = datx + 0x30;
    lcd_putchar(datax);
}

```



```

char limitPilih = -1;

lcd_clear();
lcd_gotoxy(0,0);
switch (NoMenu) {
case 1: // Run
switch (NoSubMenu) {
case 1: // Run
lcd_putsf("Set Max Suhu : ");
////33333333333333333333333333333333
var_in_eeprom = dtkRun;
break;
}
break;
/*
case 2: // Run
switch (NoSubMenu) {
case 1: // Run
lcd_putsf("Set Timer Stop : ");
////33333333333333333333333333333333
var_in_eeprom = dtkStop;
break;
}
break;
*/
}

while (sw_cancel) {
delay_ms(150);
lcd_gotoxy(0, 1);
tampil(var_in_eeprom);
// char limitPilih = -1;

if (!sw_ok) {
lcd_clear();
tulisKeEEPROM( NoMenu, NoSubMenu,
var_in_eeprom );
goto exitSetByte;
}
if (!sw_down) {

if ( !limitPilih )
var_in_eeprom--;
else {
if ( var_in_eeprom == 0 )
var_in_eeprom = limitPilih;
}
}
}
}

```

```

else
    var_in_eeprom--;
}
}
if (!sw_up) {

    if ( !limitPilih )
        var_in_eeprom++;
    else {
        if ( var_in_eeprom == limitPilih )
            var_in_eeprom = 0;
        else
            var_in_eeprom++;
    }
}
}
exitSetByte:
delay_ms(50);
lcd_clear();
}

/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
byte kursorDetikStop,kursorDetikRun;
void showMenu() { PORTD.6=0;PORTD.7=0;
    lcd_clear(); PORTB=0xff;PORTC.7=0; //on_lcd=1;
menu01:
    delay_ms(250); // bouncing sw
    lcd_gotoxy(0,0);
        // 0123456789abcdef
    lcd_putsf(" Start");          //11111111111111111111
    lcd_gotoxy(0,1);
    lcd_putsf(" Max suhu");
    // kursor awal
    lcd_gotoxy(0,0);
    lcd_putchar(0);

    if (!sw_ok) { delay_ms(250);
        lcd_clear();
        // kursorDetikStop = 1;
        goto mulai;
    }
    if (!sw_down) {
        goto menu02;
    }
}

```

```

if (!sw_up) {
    // lcd_clear();
    goto menu01;
}

goto menu01;
menu02:
delay_ms(50); // bouncing sw
lcd_gotoxy(0,0);
    // 0123456789abcdef
lcd_putsf(" Start"); ///////////////11111111111111111111
lcd_gotoxy(0,1);
lcd_putsf(" Max suhu");
// kursor awal
lcd_gotoxy(0,1);
lcd_putchar(0);
if (!sw_ok) {
    lcd_clear();
    kursorDetikRun = 1;
    goto setDetikRun;
}

if (!sw_down) {
    goto menu02;
}
if (!sw_up) {
    // lcd_clear();
    goto menu01;
}

goto menu02;

setDetikRun:
delay_ms(50);
lcd_gotoxy(0,0);
    // 0123456789ABCDEF
lcd_putsf(" Set Max Suhu=");
/////////22222222222222222222222222222222
// lcd_putsf(" 250 200 300 ");
lcd_putchar(' '); lcd_putchar(' ');
tampil(dtKRun); lcd_putchar(' '); lcd_putchar(' ');
//tampil(Ki); lcd_putchar(' '); lcd_putchar(' ');
// tampil(Kd); lcd_putchar(' '); lcd_putchar(' ');

switch (kursorDetikRun) {
    case 1:

```



```

    }

    goto setDetikStop;
    */
    mulai:

}
/////////////////////////////////////////////////////////////////

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In
Func6=In Func7=In
// State0=T State1=T State2=T State3=T State4=T State5=T State6=T
State7=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In
Func6=In Func7=In
// State0=T State1=T State2=T State3=T State4=T State5=T State6=T
State7=T
PORTB=0xff;
DDRB=0xff;
/*
// Port C initialization
// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In
Func6=In Func7=In
// State0=T State1=T State2=T State3=T State4=T State5=T State6=T
State7=T
PORTC=0x80;
DDRC=0x80;
*/
// Port C initialization
// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In
Func6=In Func7=Out
// State0=P State1=P State2=P State3=P State4=P State5=P State6=P
State7=0
PORTC=0x7F;

```

```

DDRC=0x80;

// Port D initialization
// Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In
Func6=In Func7=In
// State0=T State1=T State2=T State3=T State4=T State5=T State6=T
State7=T
PORTD=0x00;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

```

```

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
// Analog Comparator Output: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 172.800 kHz
// ADC Voltage Reference: AREF pin
// ADC High Speed Mode: Off
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE;
ADCSRA=0x86;
SFIOR&=0xEF;

// LCD module initialization
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf("  Aplikasi  ");
lcd_gotoxy(0,1);
lcd_putsf("Trmmeter Digital");
delay_ms(1000);
define_char(char0,0); //biar ngga error

//showMenu();
jeda=dtkRun;
lcd_clear();
delay_ms(500);
while (1)
{
    // Place your code here

    //opamp harus dikasih tegangan 12v agar range teg output
    bisa max sekitar 9vdc
    lcd_gotoxy(0,0);

```

```

    tampilx(read_adc(0));
    //4.8= nilai vreff adc/10 bit
    //temp=((read_adc(0)*4.8)/100)+8; //penguatan 10x
    temp=(read_adc(0)*4.8)/20; //penguatan 2x pake 180 ohm
n 180 ohm
    //temp=(read_adc(0)*4.8)/10;
    lcd_gotoxy(0,1);
    sprintf(lcd_bufferx,"drjt: %0.2f C ",temp);
    lcd_puts(lcd_bufferx);

    PORTB=0xff;
    jeda=dtkRun;
    while (!sw_ok)
    {
        showMenu();
    }

    if(temp<=jeda)
    {
        while (!sw_ok)
        {
            showMenu();
        }
        PORTC.7=1; lcd_gotoxy(10,0);lcd_putsf("hidup");
        PORTB=0xff;
    }

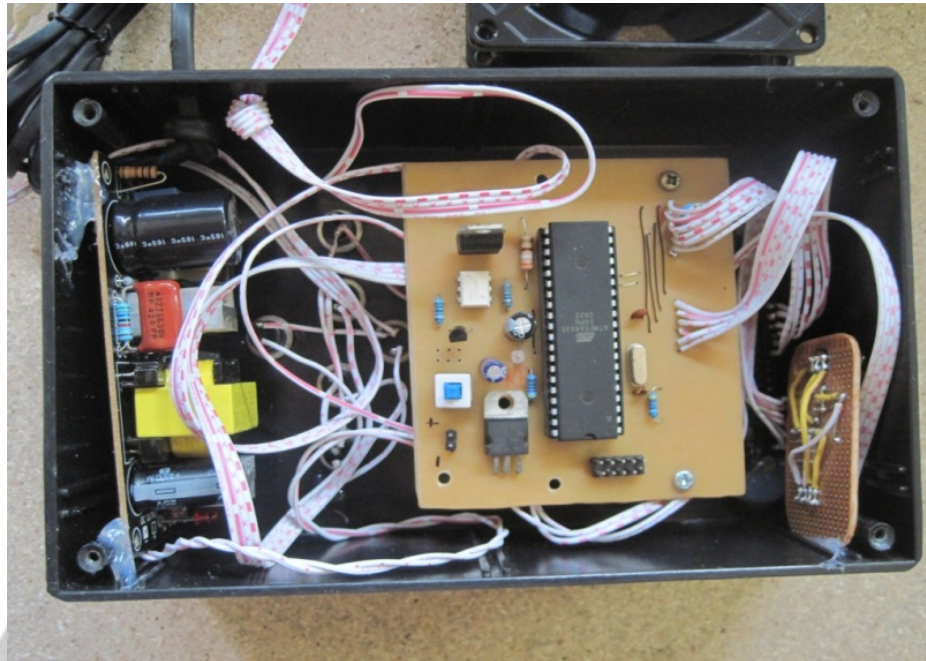
    else if(temp>=jeda)
    {
        while (!sw_ok)
        {
            showMenu();
        }
        PORTC.7=0; lcd_gotoxy(10,0);lcd_putsf("mati ");
        // PORTB=0xff;
        //buzzer
        //unsigned char i;
        for(i=0;i<=2000;i++)
        {

            PORTC.7=0;
            while (!sw_ok)
            {
                showMenu();
            }
        }
    }

```



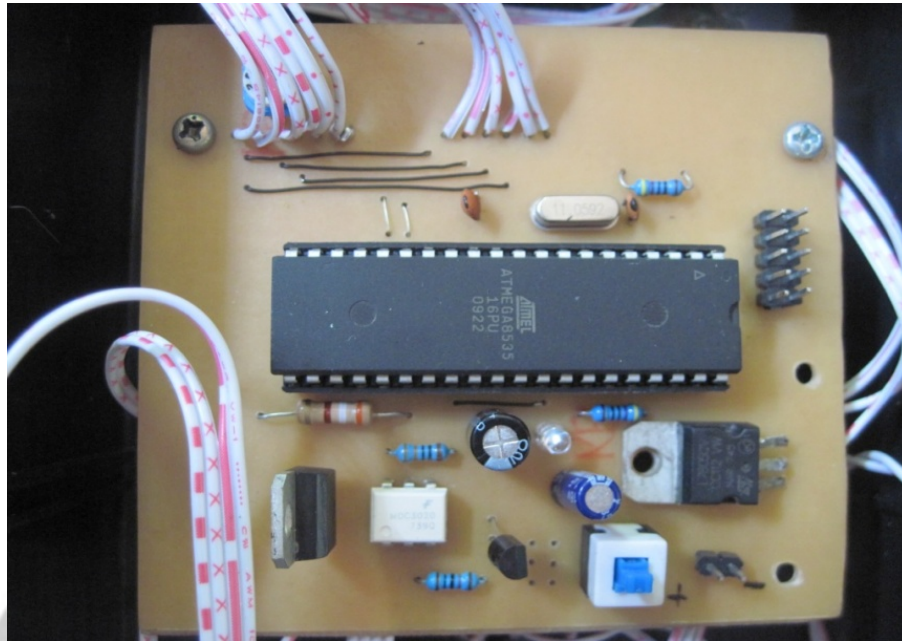
```
    }  
    delay_us(125);PORTB=0xff;//buzzer=1;  
    delay_us(125);PORTB=0x00;//buzzer=0;  
  
    if(i>=400)  
    {  
        while (!sw_ok)  
        {  
            showMenu();  
        }  
        PORTC.7=0;  
        PORTB=0xff;//delay_ms(3000);  
    }  
    // delay_ms(3000);  
    // buzzer=1;  
}  
  
while (!sw_ok)  
{  
showMenu();  
}  
};  
}
```



Gambar 14. Rangkaian Otomatisasi Menggunakan Sensor Suhu



Gambar 15. Rangkaian Otomatisasi Sensor Suhu Dengan Kipas Angin



Gambar 16. Mikrokontroler ATmega8535



Gambar 17. Box(Panel) Alat Otomatisasi Kipas Angin Menggunakan Sensor Suhu