



**PENGATURAN KIPAS BERBASIS MIKROKONTROLER
DENGAN MENGGUNAKAN SENSOR SUHU**

TUGAS AKHIR

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

**oleh
Ari Wibowo
5350306007**

**PERPUSTAKAAN
UNNES**

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG
2011**

PENGESAHAN

Tugas Akhir ini telah dipertahankan dihadapan Panitia Ujian Tugas Akhir Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Semarang pada tanggal Juli 2011

Panitia :

Ketua

Sekretaris

Drs. Said Sunardiyo, M.T.

Drs. Said Sunardiyo, M.T.

NIP 196505121991031003

NIP 196505121991031003

Penguji I

Penguji II /Pembimbing

Drs. Sugeng Purbawanto, M.T.

Drs. R. Kartono, M.Pd.

NIP 195703281984031001

NIP 195504211985031003

Mengetahui

Dekan Fakultas Teknik

Drs. Abdurrahman, M. Pd.
NIP 196009031985031002

MOTTO DAN PERSEMBAHAN

1. “ memberi semua yang kita punya bukanlah hal yang terbaik, tapi memberi yang dibutuhkan adalah yang terbaik ”.
2. “ percayalah akan hal yang dianggap penting, karena hal yang dipercaya adalah modal keberhasilan ”.

Laporan ini kupersembahkan untuk :

1. ALLAH SWT yang selalu memberikan karunia kepadaku.
2. Kedua orang tuaku yang selalu mendoakanku dan memberi motivasi.
3. Teknik Elektro UNNES tercinta
4. Teman-temanku seperjuangan.

PERPUSTAKAAN
UNNES

KATA PENGANTAR

Puji syukur saya panjatkan kepada ALLAH SWT yang telah melimpahkan rahmat dan hidayahnya, sehingga laporan tugas akhir dengan judul **“Pengaturan Kipas Berbasis Mikrokontroler Dengan Menggunakan Sensor Suhu”**, dapat diselesaikan.

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

1. Drs. Abdurrahman, M.Pd selaku Dekan Fakultas Teknik Universitas Negeri Semarang.
2. Drs. Djoko Adi Widodo, M.T selaku Ketua Jurusan Teknik Elektro Universitas Negeri Semarang.
3. Drs. Agus Murnomo, M.T selaku Ketua Program Studi D3 Teknik Elektro Universitas Negeri Semarang.
4. Drs. R. Kartono, M.Pd selaku dosen pembimbing yang telah membantu memberikan bimbingan terbaik.
5. Kedua orang tuaku yang selalu mendukung dan mendoakan.
6. Semua pihak yang telah membantu menyelesaikan Tugas Akhir yang tidak dapat penulis sebutkan satu-persatu.

Semarang, Juli 2011

Penulis

ABSTRAK

Wibowo, Ari, 2011. “**Pengaturan Kipas Berbasis Mikrokontroler Dengan Menggunakan Sensor Suhu** “. Tugas Akhir, Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang. Drs R. Kartono, Mpd

Seiring perkembangan ilmu pengetahuan dan teknologi, saat sekarang banyak muncul gagasan-gagasan bidang elektronika digital. Sistem digital berkembang diaplikasikan pada teknologi mikrokontroler. Sistem ini menyederhanakan sistem dari konvensional menjadi otomatis. Tugas akhir ini dimaksudkan mengkaji pemanfaatan mikrokontroler pengaturan kipas. Umumnya kipas angin di dalam rumah masih diatur oleh saklar, sehingga pemakai menghidupkan dan mematikan kipas serta mengatur speed kipas secara manual. Adanya rangkaian pengontrol perangkat listrik, pemakai dapat mengontrol kipas menghidupkan dan mematikan serta memindah tingkatan speed kipas secara otomatis. Proses otomatisasi tersebut dikontrol menggunakan program mikrokontroler sehingga dapat mengerakkan relai dan mengatur kecepatan putar kipas.

Penyusunan laporan tugas akhir ini memerlukan metode *prototipe*, yaitu metode membuat suatu alat yang belum pernah dibuat sebelumnya oleh orang lain, dirancang dan dikembangkan sehingga dapat tercipta sebuah alat baru. *Prototipe* memberikan masukan pada tingkat otomatisasi pesawat kipas angin berbasis mikrokontroler dengan menggunakan sensor suhu.

Rangkaian ini dapat mengatur kecepatan kipas berdasarkan suhu ruangan. Dari hasil uji coba yang telah dilakukan menunjukkan bahwa rangkaian ini bekerja dengan baik.

Hasil dari pengujian alat yang dilakukan percobaan sebanyak 5 kali tersebut terjadi ketidakcocokan antara hasil percobaan saat suhu naik dan saat suhu turun. Dikarenakan sensor DS 1820 pembacaan suhu ruangnya $\pm 0,5$ derajat C pada saat pembacaan suhu naik maupun saat turun. Sehingga terjadi selisih 0,5 derajat antara hasil pengujian dengan hasil yang diinginkan

Kesimpulan perencanaan dan pembuatan pesawat simulasi ini adalah bahwa Mikrokontroler *ATMega8535* bisa digunakan sebagai pengaturan kipas angin dengan menggunakan *relay* sebagai saklar kipas angin, sehingga dapat mengatur kecepatan kipas angin secara otomatis. Keuntungan dari penggunaan mikrokontroler sebagai pengendalinya adalah mudah dalam pemrograman, mengubah dan koreksi kesalahan programnya. Mudah dalam pemeliharaan dan perbaikan.

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN.....	ii
MOTTO DAN PERSEMBAHAN.....	iii
KATA PENGANTAR	iv
ABSTRAK	v
DAFTAR ISI	vi
DAFTAR GAMBAR.....	viii
DAFTAR TABEL	ix
DAFTAR LAMPIRAN.....	x
BAB I PENDAHULUAN	
A. Latar Belakang	1
B. Rumusan Masalah	2
C. Batasan Masalah.....	2
D. Tujuan	2
E. Manfaat	2
BAB II PEMBUATAN DAN PEMBAHASAN	
A. Landasan Teori.....	3
1. Mikrokontroler	3
a. Pengertian Mikrokontroler	3
b. Mikrokontroler Atmega8535.....	5

2. Sensor Suhu DS1820.....	10
3. Relai	11
4. Catu Daya.....	12
B. Metode Pembuatan Alat.....	13
1. Perencanaan Pembuatan Perangkat Lunak (<i>software</i>)	13
2. Perencanaan Pembuatan Perangkat Keras (<i>Hardware</i>)	15
a. Perancangan Pembuatan Perangkat Mikrokontroler.....	15
b. Perancangan Pembuatan <i>Downloader</i>	17
c. Perencanaan Pembuatan Rangkaian Relai	18
d. Perencanaan Pembuatan Rangkaian Catu Daya	19
e. Prosedur Pembuatan Alat.....	21
C. Pengujian.....	23
1. Pengujian Catu Daya	23
2. Pengujian Alat	24
D. Pembahasan.....	26
E. Keterbatasan Alat.....	26
BAB III PENUTUP	
A. Kesimpulan	27
B. Saran	27
DAFTAR PUSTAKA	28

DAFTAR GAMBAR

Gambar 01. Konfigurasi PIN ATmega8535	6
Gambar 02. Arsitektur ATmega8535	9
Gambar 03. Rangkaian Suhu DS 1820	10
Gambar 04. Gambar Relai	12
Gambar 05. Skema Kerja Catu Daya	12
Gambar 06. Diagram Blok Alat.....	13
Gambar 07. <i>Flow chart</i> Dasar Kerja Alat	14
Gambar 08. Skema Rangkaian Sistem Minimum Mikroontroler.....	15
Gambar 09. a. Jalur Pengawatan Sistem Minimum.....	16
b. Tata letak komponen.....	16
Gambar 10. Skema rangkaian Downloader.....	17
Gambar 11. a. Jalur Pengawatan Downloader.....	18
b. Tata letak komponen	18
Gambar 12. a. Jalur Pengawatan Relai	19
b. Gambar Rangkaian Relai.....	19
Gambar 13. Rangkaian Catu Daya	20

DAFTAR TABEL

Tabel 1. Deskripsi PIN ATMega8535	7
Tabel 2. Alat Yang Digunakan Untuk Proses Pembuatan Alat.....	21
Tabel 3. Pengamatan Tegangan Keluaran Catu Daya	23
Tabel 4. Tabel Berdasarkan Seting Alat	24
Tabel 5. Hasil Pengujian	25



DAFTAR LAMPIRAN

Lampiran I. Mikrokontroler ATMega8535	29
Lampiran II. Progam.....	32
Lampiran III. Rangkain Lengkap	43
Lampiran IV. Foto Alat.....	44



BAB I

PENDAHULUAN

A. Latar Belakang

Umumnya perangkat listrik elektronika sederhana masih menggunakan sistem operasi manual. Perangkat elektronika tersebut masih menggunakan saklar manual untuk menyalakan dan mematikannya. Seiring berkembangnya zaman, saat ini ditemukan teknologi mikrokontroler. Sistem ini menyederhanakan sistem yang masih konvensional menjadi otomatis dan lebih ringkas. Dengan adanya teknologi ini maka pengendalian perangkat elektronik menjadi lebih mudah dan efisien.

Pada saat ini di pasaran AC (*air conditioner*) masih menjadi pilihan utama sebagai penyejuk udara. Dikarenakan sistem operasinya yang sudah otomatis dan fitur-fitur tambahan lainnya sehingga pemilik tidak lagi disibukkan dalam pengoperasiannya. Dengan harga AC yang sedikit mahal tidak semua lapisan masyarakat dapat memiliki alat penyejuk udara ini, hanya kalangan-kalangan menengah ke atas yang mampu memiliki. Oleh karena itu konsep alat pengaturan kipas menggunakan sensor suhu ini dibuat. Mengubah kipas yang tadinya masih manual ke otomatis. Kipas dapat menyala, mati dan menyesuaikan kecepatan kipas angin berdasarkan suhu pada ruangan. Pembuatan konsep ini bertujuan agar kipas angin walaupun dengan harga yang lebih terjangkau dari AC, tetapi memiliki sistem operasi yang otomatis seperti halnya AC. Sehingga masyarakat dapat menikmati kemudahan sebuah alat kipas angin ini dengan kemampuan yang hampir sama dengan AC tetapi dengan dana yang lebih murah.

B. Rumusan Masalah

Bagaimana merancang, membuat dan menguji coba kipas angin kontrol suhu?

C. Batasan Masalah

Permasalahan pada tugas akhir ini akan dibatasi yaitu pembuatan alat pengatur kipas dengan menggunakan mikrokontroler ATmega 8535, dan sensor suhu DS 1820 untuk mendeteksi suhu pada ruangan.

D. Tujuan

Dapat merancang dan membuat alat pengaturan kipas angin yang putarannya dikontrol oleh mikrokontroler ATmega8535 sesuai suhu ruangan.

E. Manfaat

1. Sebagai informasi pengembangan ilmu pengetahuan dan teknologi di dunia pendidikan.
2. Sebagai sarana pembelajaran otomatisasi dibidang instrumentasi kendali.

BAB II

PEMBAHASAN

A. Landasan Teori

1. Mikrokontroler

a. Pengertian Mikrokontroler

Mikrokontroler dapat dianalogikan dengan sebuah sistem komputer yang dikemas dalam sebuah chip. Artinya bahwa di dalam sebuah IC mikrokontroler sebetulnya sudah terdapat kebutuhan minimal agar mikroprosesor dapat bekerja, yaitu meliputi mikroprosesor, ROM, RAM, I/O dan clock seperti halnya yang dimiliki oleh sebuah komputer PC. Mengingat kemasannya yang hanya berupa sebuah chip dengan ukuran yang relatif kecil tentu saja spesifikasi dan kemampuan yang dimiliki oleh mikrokontroler menjadi lebih rendah bila dibandingkan dengan sistem komputer seperti PC baik dilihat dari segi kecepatan, kapasitas memori maupun fitur-fitur yang dimilikinya.

Seperti umumnya komputer, mikrokontroler sebagai alat yang mengerjakan perintah-perintah yang diberikan oleh manusia. Artinya, bagian terpenting dan utama dari suatu sistem komputerisasi adalah program itu sendiri yang dibuat oleh seorang *programmer*. Program ini memerintahkan komputer untuk melakukan jalinan yang panjang dari aksi-aksi sederhana untuk melakukan tugas yang lebih kompleks yang diinginkan oleh *programmer*.

Sistem dengan mikrokontroler umumnya menggunakan piranti *input* yang jauh lebih kecil seperti saklar atau *keypad* kecil. Hampir semua *input* mikrokontroler hanya dapat memproses sinyal *input digital* dengan tegangan yang sama dengan tegangan logika dari sumber. Tegangan positif sumber umumnya adalah 5 volt. Padahal dalam dunia nyata terdapat banyak sinyal *analog* atau sinyal dengan tegangan level yang bervariasi. Karena itu ada piranti *input* yang mengkonversikan sinyal *analog* menjadi sinyal *digital* sehingga komputer bisa mengerti dan menggunakannya. Ada beberapa mikrokontroler yang dilengkapi dengan piranti konversi ini, yang disebut dengan ADC, dalam satu rangkaian terpadu.

Mikrokontroler 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.

b. Mikrokontroler ATmega8535

Seperti yang telah dijelaskan sebelumnya, ATmega8535 memiliki fasilitas yang cukup lengkap. Dari kapasitas memori program dan memori data yang cukup besar, *interupsi*, *timer / counter*, PWM, USART, TWI, *analog comparator*, EEPROM *internal* dan juga ADC *internal* semuanya ada dalam ATmega8535.

Selain itu kemampuan kecepatan eksekusi yang lebih tinggi menjadi alasan bagi banyak orang untuk lebih memilih menggunakan mikrokontroler jenis AVR daripada mikrokontroler pendahulunya yaitu

keluarga MCS-51. Dengan demikian sangatlah tepat untuk mempelajari mikrokontroler keluarga AVR dengan ATmega8535. (Bejo 2008: 10)

Mikrokontroler ATmega8535 merupakan salah satu mikrokontroler buatan AVR yang memiliki bagian-bagian, diantaranya :

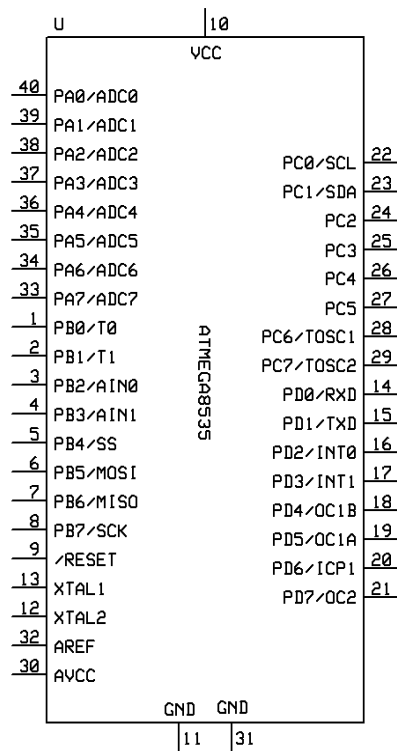
(1) Fitur ATmega8535

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

(2) Konfigurasi PIN ATmega8535

PIN adalah kaki-kaki dari sebuah IC. Gambar 1. menunjukkan konfigurasi PINn ATmega8535. (Sumber: datasheet ATmega8535)



Gambar 1. Konfigurasi PIN ATmega8535

Konfigurasi PIN ATmega 8535 pada Gambar 1 terdapat susunan masing-masing kaki mikrokontroler ATmega8535. Tabel.1 menerangkan tentang susunan dari masing-masing kaki mikrokontroler ATmega8535.

Tabel 1. Susunan kaki 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 internal</i>. 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 Mater 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 <i>digital</i>
11	GND, Ground untuk catu daya <i>digital</i>
12	XTAL2, merupakan <i>output</i> dari penguat oscillator pembalik
13	XTAL1, merupakan <i>input</i> ke penguat oscillator pembalik dan <i>input</i> ke <i>internal clock</i> .
14-21	<p>Port D, merupakan Port I/O 8 bit dua arah (<i>bi-directional</i>) dengan resistor <i>pull-up internal</i>. 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 Interupt 1 Input</i>)

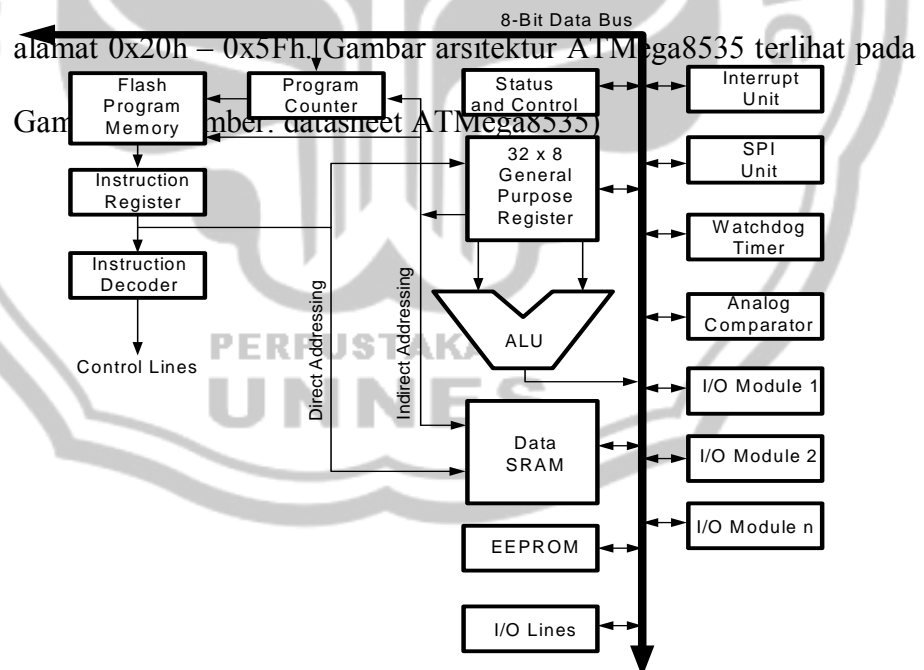
	<ul style="list-style-type: none"> • PD2 : INT0 (<i>External Interrupt 0 Input</i>) • PD1 : TXD (<i>USART transmit</i>) • PD0 : RXD (<i>USART receive</i>)
22-29	<p>Port C, merupakan Port I/O 8 bit dua arah (<i>bi-directional</i>) dengan resistor pull-up <i>internal</i>. Selain sebagai Port I/O 8 bit 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 <i>analog</i> ADC yang terhubung ke Port A
31	GND, Ground untuk catu daya <i>analog</i>
32	AREF, merupakan tegangan referensi <i>analog</i> untuk ADC
33-40	Port A, merupakan Port I/O 8 bit dua arah (<i>bi-directional</i>) dengan resistor pull-up <i>internal</i> . Selain sebagai Port I/O 8 bit Port A juga dapat berfungsi sebagai masukan 8 <i>channel</i> ADC

(3) Arsitektur ATmega8535

Mikrokontroler ATmega8535 memiliki arsitektur Harvard, yaitu memisahkan memori untuk kode program dan memori untuk data sehingga dapat memaksimalkan unjuk kerja dan paralelisme. Instruksi-instruksi dalam memori program dieksekusi dalam satu alur tunggal, dimana 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. Enam dari *register* serba guna dapat digunakan sebagai tiga buah *register pointer* 16 bit pada mode pengalamatan tak langsung untuk mengambil data pada ruang memori data.

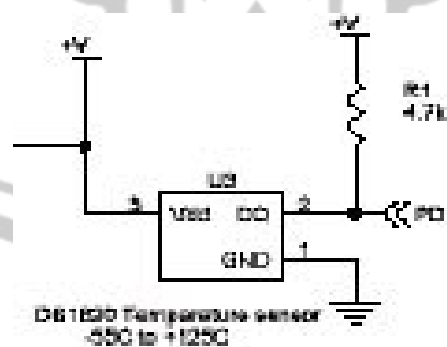
Hampir semua perintah AVR memiliki format 16 bit (*word*). Setiap alamat memori program terdiri dari instruksi 16 bit atau 32 bit. Selain *register* serba guna, terdapat *register* lain yang terpetakan dengan teknik *memory mapped I/O* selebar 64 Byte. Beberapa *register* ini digunakan untuk fungsi khusus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



Gambar 2. Arsitektur ATmega8535

2. Sensor DS 1820

Sensor DS 1820 terbuat dari bahan semikonduktor. Pada semikonduktor, energi bandgap-nya berada pada kisaran 1 eV. Dengan meningkatnya suhu, maka akan lebih banyak ikatan kovalen yang pecah dan lebih banyak pula elektron bebas maupun ruang kosong positif akan terbentuk. Elektron valensi dengan energi thermal tertentu dan berdekatan dengan sebuah ruang kosong dapat berpindah posisi tersebut, sehingga terlihat seperti muatan positif yang bergerak diantara semikonduktor. Partikel bermuatan positif ini disebut hole. Pada semikonduktor, dua jenis partikel bermuatan ini berjasa dalam menghasilkan arus : elektron bebas yang bermuatan negatif, serta hole yang bermuatan positif. Konsentrasi elektron dan hole adalah parameter penting dalam karakteristik dari sebuah unsur semikonduktor, karena mereka berpengaruh langsung terhadap besarnya arus.



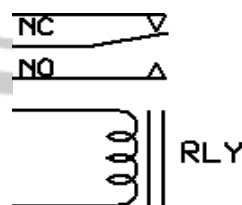
Gambar 3. Rangkaian Suhu DS 1820

DS18S20 menggunakan eksklusif Maxim 1-Wire bus yang mengimplementasikan protokol komunikasi bus menggunakan satu kontrol sinyal. Dalam sistem bus, mikroprosesor (perangkat master) mengidentifikasi dan alamat perangkat pada bus menggunakan setiap perangkat unik. Kode 64-bit. Karena masing-masing perangkat memiliki kode yang unik, jumlah perangkat yang dapat ditangani pada satu bus hampir tak terbatas. 1-Wire bus protokol, termasuk penjelasan rinci tentang perintah dan "Slot waktu," adalah tercakup dalam bagian Sistem 1-Wire Bus. Fitur lain dari DS18S20 adalah kemampuan untuk beroperasi tanpa catu daya eksternal. Power bukan dipasok melalui resistor pullup 1-Wire melalui pin DQ ketika bus yang tinggi. Bus yang tinggi sinyal juga biaya kapasitor internal (CPP), yang kemudian pasokan listrik ke perangkat ketika bus yang rendah. Metode ini berasal daya dari bus 1-Wire disebut sebagai Sebagai "kekuatan parasit" alternatif, DS18S20 juga dapat didukung oleh pasokan eksternal pada Vdd. Sensor DS1820 memiliki kemampuan untuk mengukur suhu pada kisaran -55°C sampai 125°C dan bekerja secara akurat dengan kesalahan $\pm 0,5^{\circ}\text{C}$ pada kisaran -10°C sampai 85°C .

3. Relai

Relai adalah komponen elektronika berupa saklar elektronik yang digerakkan oleh arus listrik. Secara prinsip, relai merupakan tuas saklar dengan lilitan kawat pada batang besi (solenoid) di dekatnya. Ketika solenoid dialiri arus listrik, tuas akan tertarik karena adanya gaya

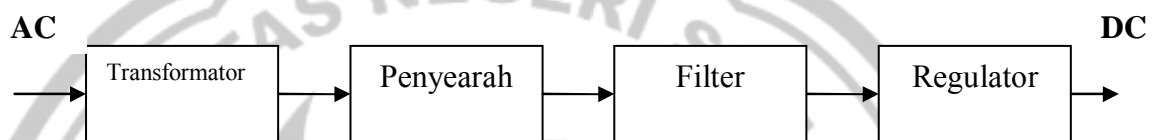
magnet yang terjadi pada solenoid sehingga kontak saklar akan menutup. Pada saat arus dihentikan, gaya magnet akan hilang, tuas akan kembali ke posisi semula dan kontak saklar kembali terbuka. Relai biasanya digunakan untuk menggerakkan arus/tegangan yang besar. Dalam pemakaiannya biasanya relai yang digerakkan dengan arus DC dilengkapi dengan sebuah dioda yang di-paralel dengan lilitannya dan dipasang terbalik yaitu anoda pada tegangan (-) dan katoda pada tegangan (+). Ini bertujuan untuk mengantisipasi sentakan listrik yang terjadi pada saat relai berganti posisi dari on ke off agar tidak merusak komponen di sekitarnya. Penggunaan relai perlu memperhatikan tegangan pengontrolnya serta kekuatan relai men-switch arus/tegangan. Biasanya ukurannya tertera pada bodi relai. Misalnya relai 12VDC/4 A 220V, artinya tegangan yang diperlukan sebagai pengontrolnya adalah 12Volt DC dan mampu men-switch arus listrik (maksimal) sebesar 4 ampere pada tegangan 220 Volt. Sebaiknya relay difungsikan 80% saja dari kemampuan maksimalnya agar aman, lebih rendah lagi lebih aman.



Gambar 4. Relai untuk mengatur kecepatan kipas

4. Catu Daya

Catu daya atau *power supply* adalah suatu rangkaian yang berfungsi sebagai sumber daya untuk mengoperasikan rangkaian yang lain. Jenis rangkaian catu daya cukup banyak tetapi untuk rangkaian yang sederhana biasanya terdiri dari *transformator*, penyearah, *filter*, dan *regulator*. Prinsip dasar untuk memperoleh daya searah jala-jala dapat dijelaskan secara diagram berikut:

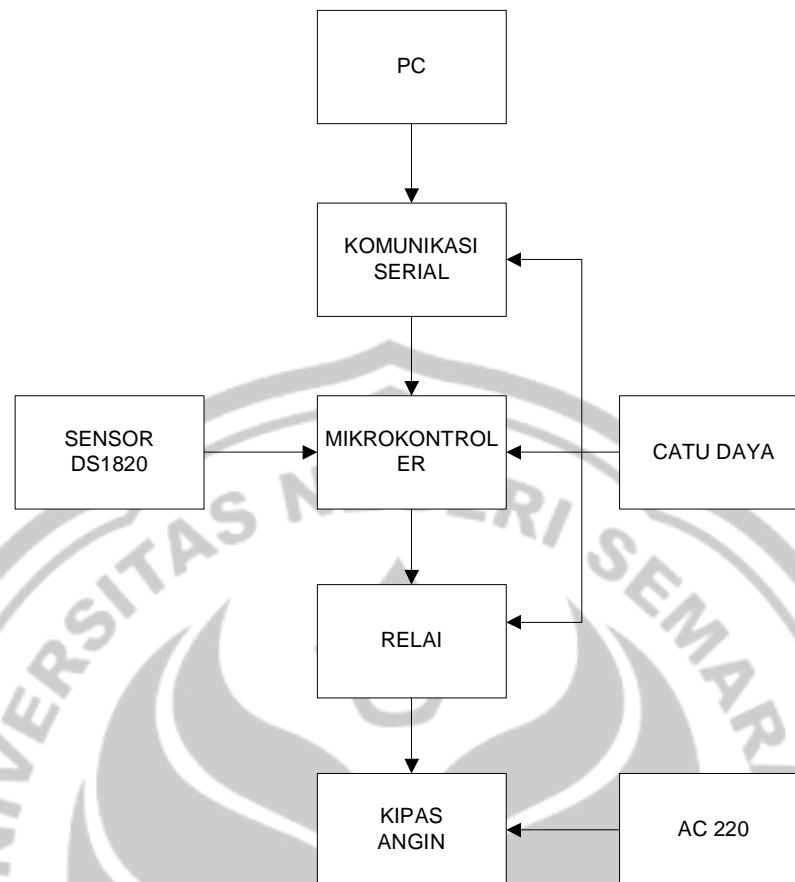


Gambar 5. Gambar Skema Kerja Catu Daya

B. Metode Pembuatan Alat

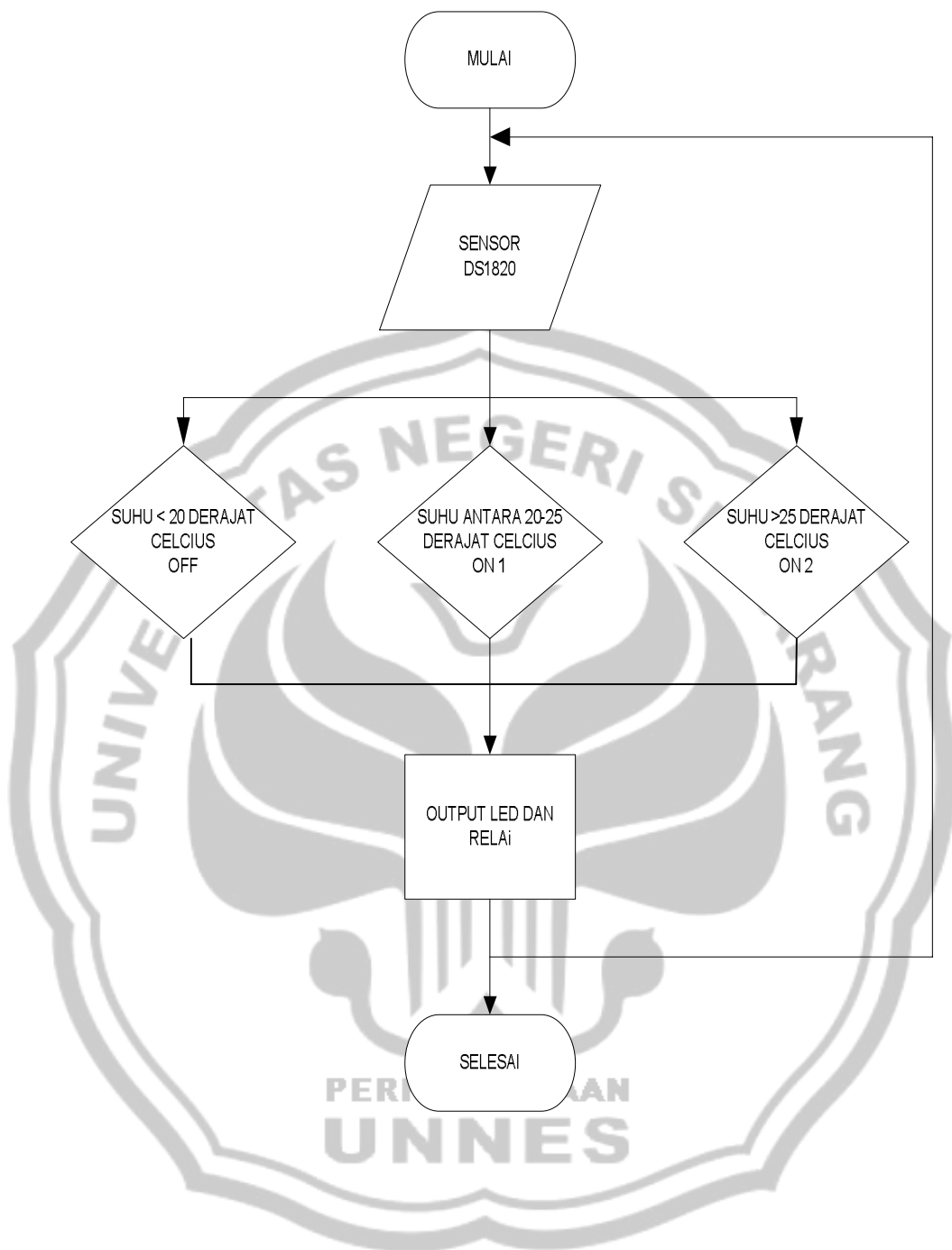
1. Perencanaan Pembuatan Perangkat Lunak (*Software*)

Pembuatan diagram blok dimaksudkan untuk memudahkan pembuatan program, karena dengan diagram blok bisa diketahui alur kerja alat. Gambar 3 dan Gambar 4 menunjukkan gambar diagram dasar kerja alat



Keterangan : PC hanya digunakan pada waktu pemrograman mikrokontroler saja.

Gambar 6. Diagram Blok Pengaturan Kipas Berbasis Mikrokontroler Dengan Menggunakan Sensor Suhu.

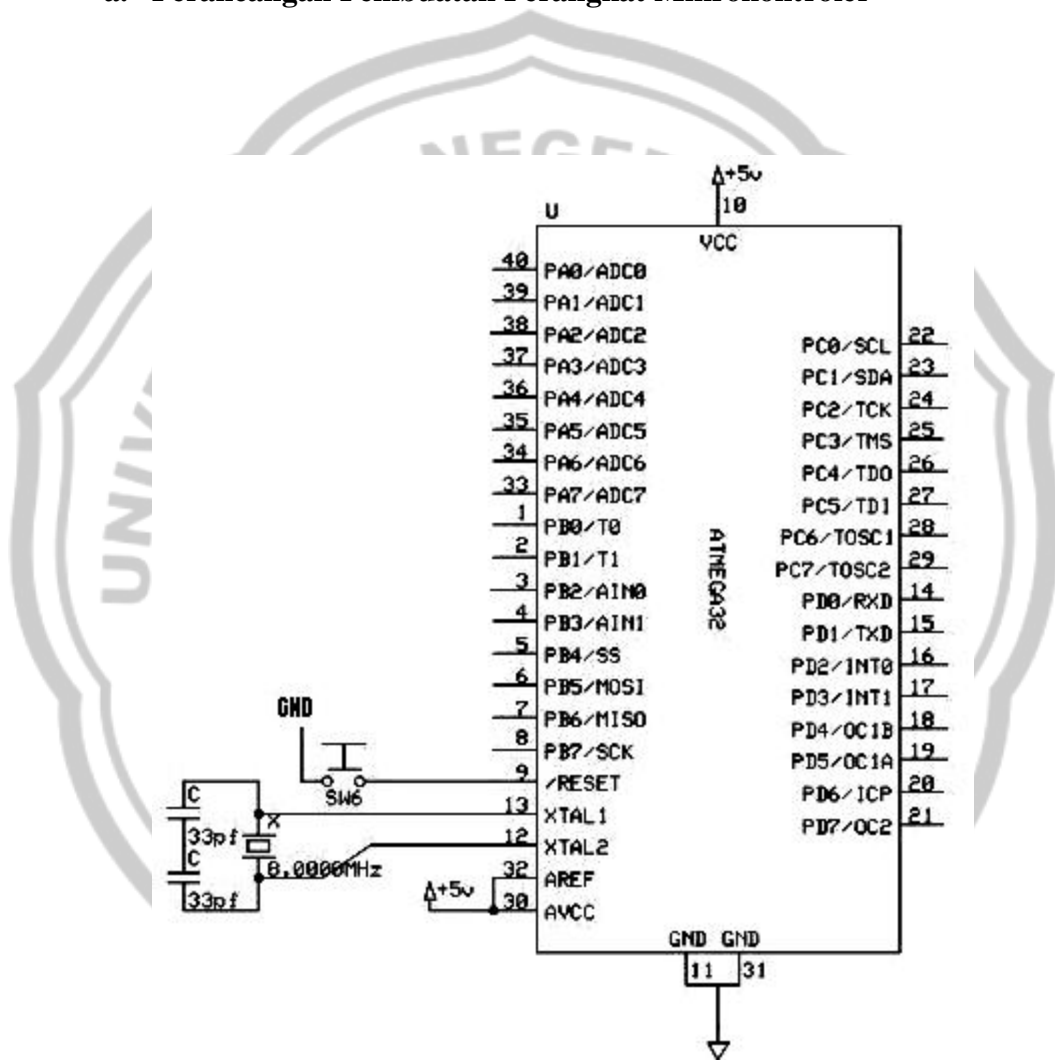


Gambar 7. *Flow chart* Pengaturan Kipas Berbasis Mikrokontroler Dengan Menggunakan Sensor Suhu.

2. Perancangan Pembuatan Perangkat Keras (*hardware*)

Langkah pertama dalam perancangan *hardware* adalah membuat desain skema rangkaian serta menentukan alat dan bahan yang akan digunakan

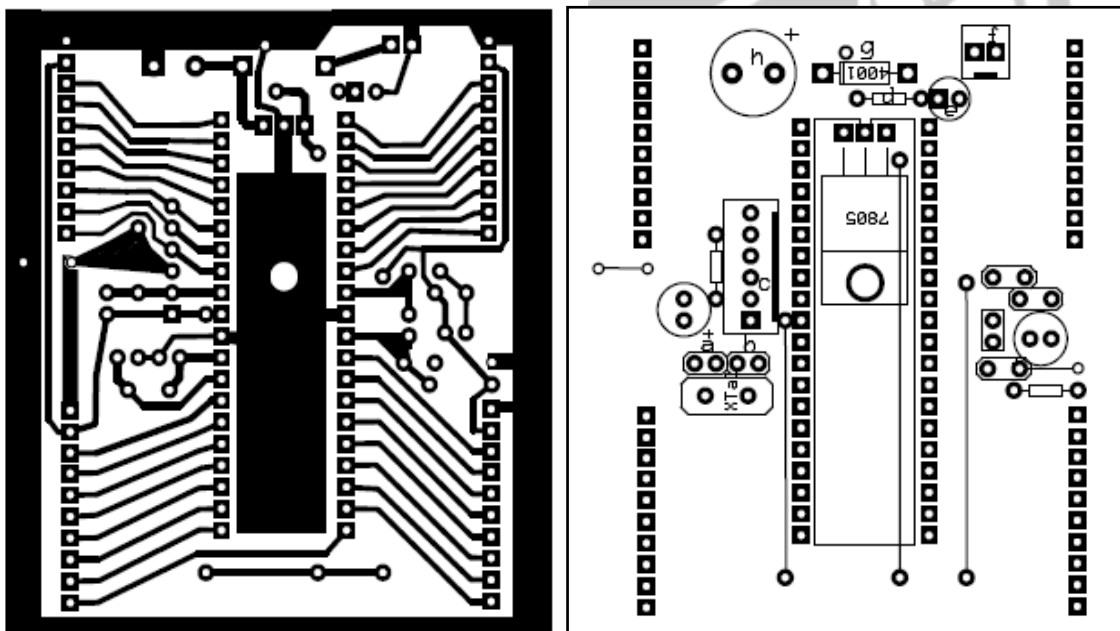
a. Perancangan Pembuatan Perangkat Mikrokontroler



Gambar 8. Skema Rangkaian Sistem Minimum Mikrokontroler

(1) Komponen yang diperlukan:

- (a) Resistor 100Ω $\frac{1}{4}$ W.....8 buah
- (b) Resistor 330Ω $\frac{1}{4}$ W.....1 buah
- (c) IC ATmega 8535.....1 buah
- (d) IC LM 7805.....1 buah
- (e) Kristal 8.00000 Mhz.....1 buah
- (f) Capacitor 1000uf/16v..... 1 buah
- (g) Capacitor 30 pf.....2 buah
- (h) Conektor 2pin.....1 buah
- (i) Conektor 6pin.....1 buah
- (j) Dioda IN 4002.....1 buah
- (k) Transistor BD139.....1 buah



a

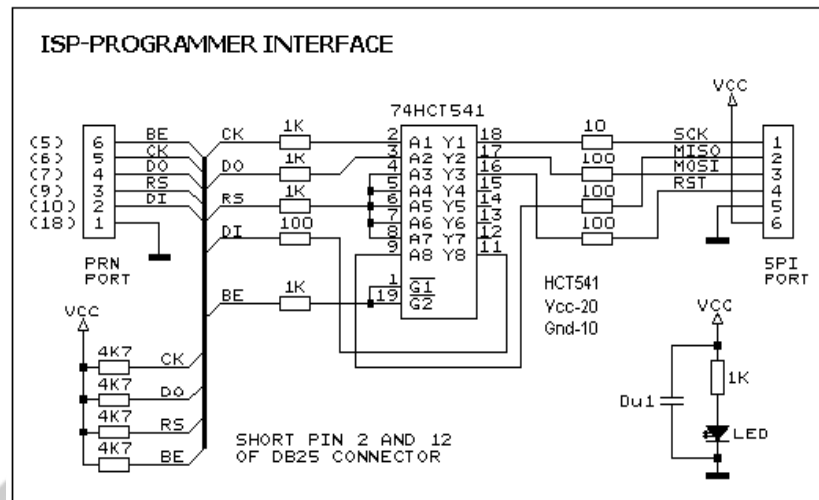
b

Gambar 9. Rangkaian Tercetak Sistem Minimum.

a. Jalur Pengawatan

b. Tata Letak Komponen

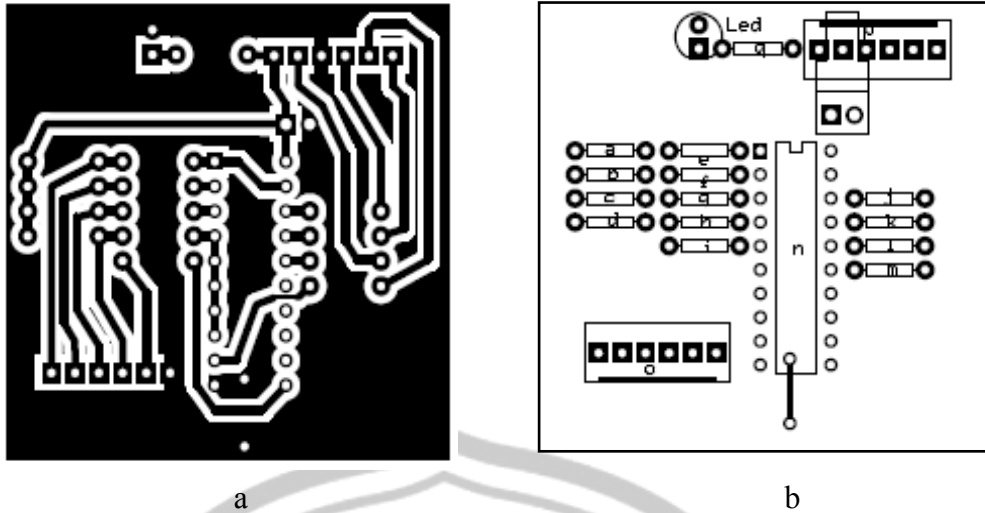
b. Perancangan Pembuatan *Downloader*



Gambar 10. ISP Programmer interface Type Kanda Sistem STK200+/300

(1) Daftar Komponen:

- | | |
|--|--------|
| (a) Resistor 4K7 Ω $\frac{1}{4}$ W..... | 4 buah |
| (b) Resistor 100 Ω $\frac{1}{4}$ W..... | 5 buah |
| (c) Resistor 1K Ω $\frac{1}{4}$ W..... | 4 buah |
| (d) Resistor 330 Ω $\frac{1}{4}$ W..... | 1 buah |
| (e) IC 74LS541 + soket..... | 1 buah |
| (f) LED $\frac{1}{4}$ W..... | 1 buah |
| (g) Conektor DB 25 pin..... | 1 buah |
| (h) Conektor 6 pin..... | 2 buah |



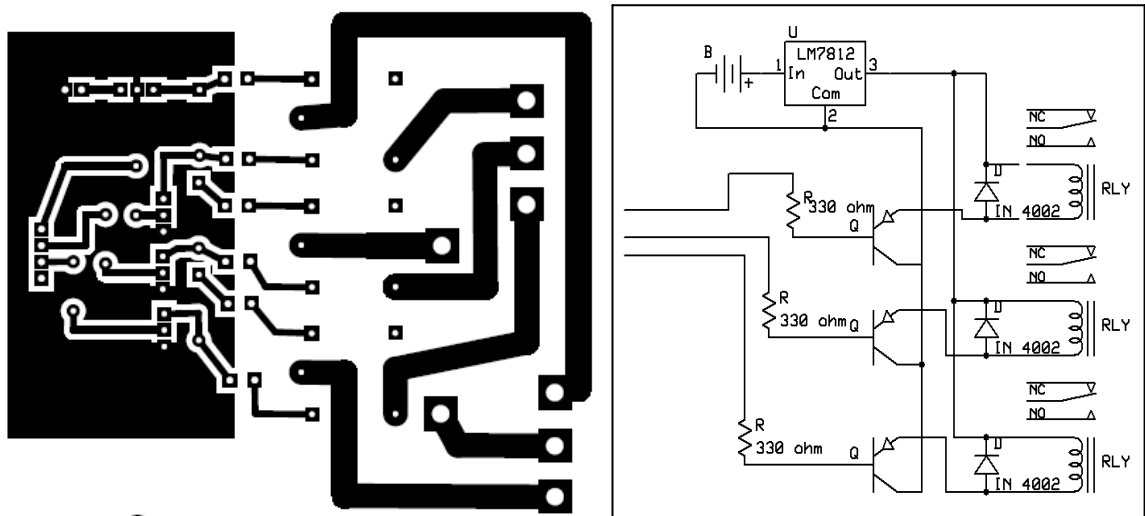
Gambar 11. Rangkaian Tercetak Downloader

- a. Jalur Pengawatan
- b. Tata Letak Komponen

c. Perencanaan Pembuatan Rangkaian Relai

Relai berfungsi untuk menghidupkan kipas, mikrokontroler tidak mampu memberikan arus yang besar, sehingga perlu perangkat yang mampu untuk menghidupkan kipas tersebut.

Saat kaki basis transistor menerima tegangan dari mikrokontroler maka transistor akan bekerja sebagai saklar untuk menghidupkan kipas sehingga arus yang besar dapat dialirkan melalui relai.



a

b

Gambar 12. Rangkaian Tercetak Relai

a. Jalur Pengawatan

b. Gambar Rangkaian

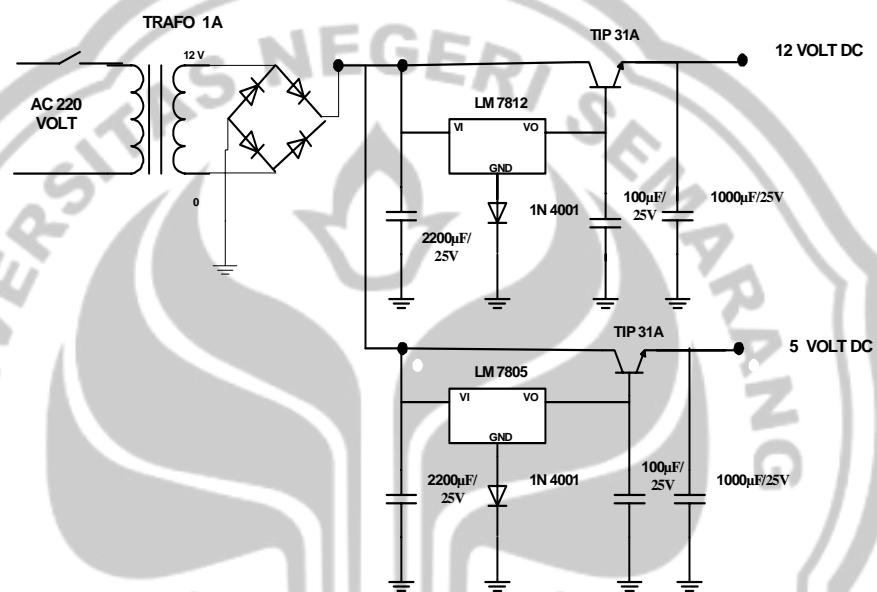
Daftar komponen :

- | | |
|----------------------------|--------|
| (a) Resistor 330Ω..... | 3 buah |
| (b) Transistor 2222 N..... | 3 buah |
| (c) Relay 12V/250V..... | 3 buah |
| (d) Dioda IN 4002..... | 3 buah |
| (e) Konektor 2 pin..... | 1 buah |
| (f) Konektor 4 pin..... | 1 buah |

d. Perencanaan pembuatan rangkaian Catu Daya

Catu daya 5 V menggunakan transformator step down 2A tanpa CT (*center tap*). Penyearah digunakan penyearah gelombang penuh dengan menggunakan model dioda *bridge* dengan kapasitor sebagai filter. Agar tegangan keluaran lebih presisi digunakan IC regulator 7805. Sedangkan untuk pengaman pada input AC atau sisi primer trafo menggunakan *fuse*, *varistor* dan kapasitor non polar

275 V. Fungsi dari pengaman tersebut adalah untuk menghindari *Over Voltage* dari batas yang telah ditentukan yaitu 220 V AC. Cara kerja dari pengaman tersebut adalah jika tegangan jala – jala PLN naik menjadi 230 V AC maka varistor akan putus (kontak) sehingga *varistor short* dengan *fuse*, akibatnya *fuse* menjadi putus dan rangkaian elektronik aman dari bahaya *over voltage*.



Gambar 13. Rangkaian Catu Daya

Daftar komponen :

- | | |
|-------------------------------|--------|
| (a) Trafo 1A..... | 1 buah |
| (b) Dioda Brake 2A..... | 1 buah |
| (c) IC LM7812..... | 1 buah |
| (d) IC LM7805..... | 1 buah |
| (e) Dioda IN 4001..... | 2 buah |
| (f) Kapasitor 2200uF/25v..... | 2 buah |
| (g) Kapasitor 1000uF/25v..... | 2 buah |
| (h) Kapasitor 100uF/25v..... | 2 buah |
| (i) TIP3 1A..... | 2 buah |

e. Prosedur Pembuatan Alat

- (1) Menyiapkan alat-alat yang akan digunakan

Tabel 02. Alat yang digunakan untuk proses pembuatan alat

Nama alat	Spesifikasi	Ukuran	Jumlah
Spidol	Permanent	0,5 mm	1
Bor	Kecil 12V	0,4 mm	1
Tang	Potong	Sedang	1
Penggaris	Besi	30 cm	1
Cutter	-	Besar	1
Solder	-	-	1
Toolset	Lengkap	-	1

- (2) Menggambar jalur

Teknik yang digunakan adalah teknik sablon setrika. Langkah pertama adalah menggambar lay out PCB. Gambar tersebut di foto kopi di kertas transparan (mika) dan selanjutnya gambar disetrika di atas papan PCB sampai gambar tersebut menempel di papan PCB. Pada saat akan melakukan setrika kita harus membersihkan papan PCB terlebih dahulu agar gambar dapat menempel dengan sempurna.

- (3) Pelarutan Papan PCB

Gambar di papan PCB yang telah selesai dibuat kemudian dilarutkan dengan cairan *ferridclorida* (FhCl) untuk menghilangkan lapisan tembaga yang tidak tertutup gambar. Pada

saat melakukan pelarutan, wadah yang digunakan harus selalu di goyang-goyang.

(4) Pelubangan Papan PCB

Proses pelubangan adalah untuk membuat tempat memasang komponen yang akan digunakan.

(5) Memasang Komponen ke Papan PCB

Pemasangan komponen harus sesuai dengan tata letak yang telah ditentukan sebelumnya.

(6) Menyolder

Untuk menghasilkan hasil solderan yang baik sebaiknya pada kaki-kaki komponen dan PCB dibersihkan dahulu agar tenol dapat menempel dengan baik. Saat menyolder gunakanlah mata solder yang bersih dan lancip agar mempermudah dalam penyolderan.

(7) Pemotongan Kaki Komponen

Pemotongan kaki komponen dilakukan untuk meratakan kaki komponen agar menjadi rapi.

C. Pengujian

1. Pengujian Catu Daya

Pengujian rangkaian catu daya ini bertujuan untuk mengetahui tegangan yang dikeluarkan oleh rangkaian. Hasil pengukuran tegangan

keluaran yang dihasilkan oleh rangkaian catu daya dapat dilihat dalam tabel 03:

Tabel 03. Tabel Pengamatan Tegangan Keluaran Catu Daya

No.	Tegangan Keluaran Catu Daya	
	Tegangan Yang Diharapkan	Tegangan Yang Dihasilkan
1.	5 Volt	5,2 Volt
2.	12 Volt	11,85 Volt

Dalam hal ini tegangan keluaran rangkaian Catu Daya banyak yang tidak sesuai dengan yang diharapkan, hal ini disebabkan oleh beberapa faktor antara lain adalah kualitas bahan dari tiap komponen yang digunakan sudah tidak murni lagi atau adanya nilai toleransi dari komponen tersebut.

2. Pengujian Alat

Dari 5 percobaan alat ini masing-masing memiliki data yang digunakan untuk membuktikan suatu proses otomatisasinya. Data tersebut berupa *speed* kipas angin, kipas angin menyala maupun mati. Adapun hasil dari uji coba rangkaian yang telah dilakukan dapat dilihat pada Tabel 4 dan 5.

Tabel 04. Tabel Berdasarkan Seting Alat

No.	Berdasarkan Setting		Keterangan
	Pembacaan Suhu	Kecepatan Kipas Angin (speed)	
1.	Suhu < 20 derajat	Off	Suhu di bawah 20 derajat kipas angin mati sesuai setting
2.	Suhu 20 – 25 derajat	On 1	Suhu antara 20 – 25 derajat kipas angin menyala pada speed 1 sesuai setting.
3.	Suhu > 25 derajat	On 2	Suhu di atas 25 derajat kipas angin menyala pada speed 2 sesuai setting.

Table 05. Tabel Hasil Pengujian

No.	Pembacaan Suhu					Suhu Rata-rata	Speed Kipas
	Percobaan 1	Percobaan 2	Percobaan 3	Percobaan 4	Percobaan 5		
1.	9.5 derajat	20 derajat	19 derajat	19.5 derajat	20 derajat	19,6	Off
2.	20.5 derajat	20.5 derajat	21 derajat	20.5 derajat	20.5 derajat	20,6	On 1
3.	26 derajat	25.5 derajat	25.5 derajat	26,5 derajat	25,5 derajat	25,8	On 2

D. Pembahasan

Hasil dari pengujian alat yang dilakukan percobaan sebanyak 5 kali tersebut terjadi ketidakcocokan suhu antara hasil percobaan dengan setingan. Hal tersebut dikarenakan sensor DS 1820 pembacaan tingkatan suhu ruangnya $\pm 0,5$ derajat Celcius pada saat pembacaan suhu naik maupun saat turun.

Adapun perbedaan antara tingkatan *speed* 1 dengan *speed* 2 sangat kecil karena kondisi kipas yang sudah tidak baik. Dan dalam pembahasan ini juga tidak disinggung mengenai RPM motor kipas, tetapi hanya tingkatan *speed* yang terindikasi dari nyala lampu LED. Dimulai saat kipas dalam kondisi ON pada *speed* 1 maka yang menyala LED warna merah, dan pada *speed* 2 maka yang menyala LED warna biru. Apabila kipas dalam kondisi OFF maka LED mati semuanya beserta kipasnya.

E. Keterbatasan Alat

Kelemahan yang terdapat pada rangkaian pengaturan kipas angin berbasis mikokontroler dengan menggunakan sensor suhu ini adalah penggunaan kipas angin yang kurang baik. Sehingga kecepatan 1 dan 2 pada kipas ini hampir sama.

BAB III

PENUTUP

A. Kesimpulan

Rangkaian ini disimulasikan untuk mengatur tingkat kecepatan kipas angin. Kipas angin dapat menyala, mati dan menyesuaikan tingkat kecepatan kipas angin berdasarkan suhu ruangan secara otomatis sesuai dengan program yang dimasukkan ke dalam mikrokontroler ATMEGA8535. Rangkaian ini juga memiliki kelemahan yaitu perpindahan dari kecepatan 1 ke 2 kurang terlihat perbedaannya, dikarenakan kipas angin yang digunakan kurang baik dan kecepatan 1 dan 2 pada kipas angin ini memang hampir sama kecepatannya.

Adapun perbedaan antara tingkatan speed 1 dengan speed 2 sangat kecil karena kondisi kipas yang sudah tidak baik. Dan dalam pembahasan ini juga tidak disinggung mengenai rpm speed motor kipas, tetapi hanya tingkatan speed yang terindikasi dari nyala lampu LED. Dimulai saat kipas dalam kondisi ON pada speed 1 maka yang menyala LED warna merah, dan pada speed 2 maka yang menyala LED warna biru. Apabila kipas dalam kondisi OFF maka LED mati semuanya beserta kipasnya.

B. Saran

Rangkain ini mampu mengontrol perangkat listrik berupa kipas angin secara otomatis dengan baik, akan tetapi kecepatan 1 dan 2 pada kipas hampir sama kecepatannya. Kipas ini dilengkapi dengan lampu LED indikator dan

dilengkapi LCD untuk pembacaan suhu pada ruangan untuk mempermudah dalam membedakan kecepatan pada kipas.



DAFTAR PUSTAKA

- Arikunto, Suharsimi. 2002. *Dasar – dasar evaluasi pendidikan*. Jakarta : PT. Rineka Cipto.
- Bejo, Agus. 2008. *C&AVR Rahasia Kemudahan Bahasa C dalam Mikokontroler ATmega8535*. Yogyakarta : Graha Ilmu.
- Budiharto, Widodo. 2007. *Panduan Praktikum Mikrokontroler AVR ATmega16*. Jakarta : PT. Elex Media Komputindo.
- Sutrisno. 1987. *Elektronika Teori Dan Penerapannya*. Bandung : ITB, Edisi Dua.

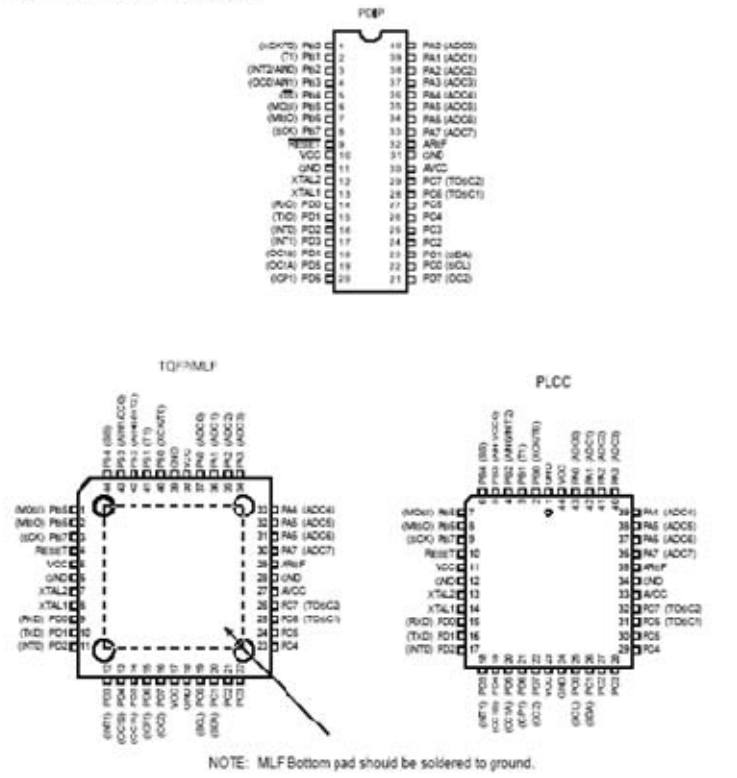


LAMPIRAN I



Pin Configurations

Figure 1. Pinout ATmega8535



NOTE: MLF Bottom pad should be soldered to ground.

Disclaimer

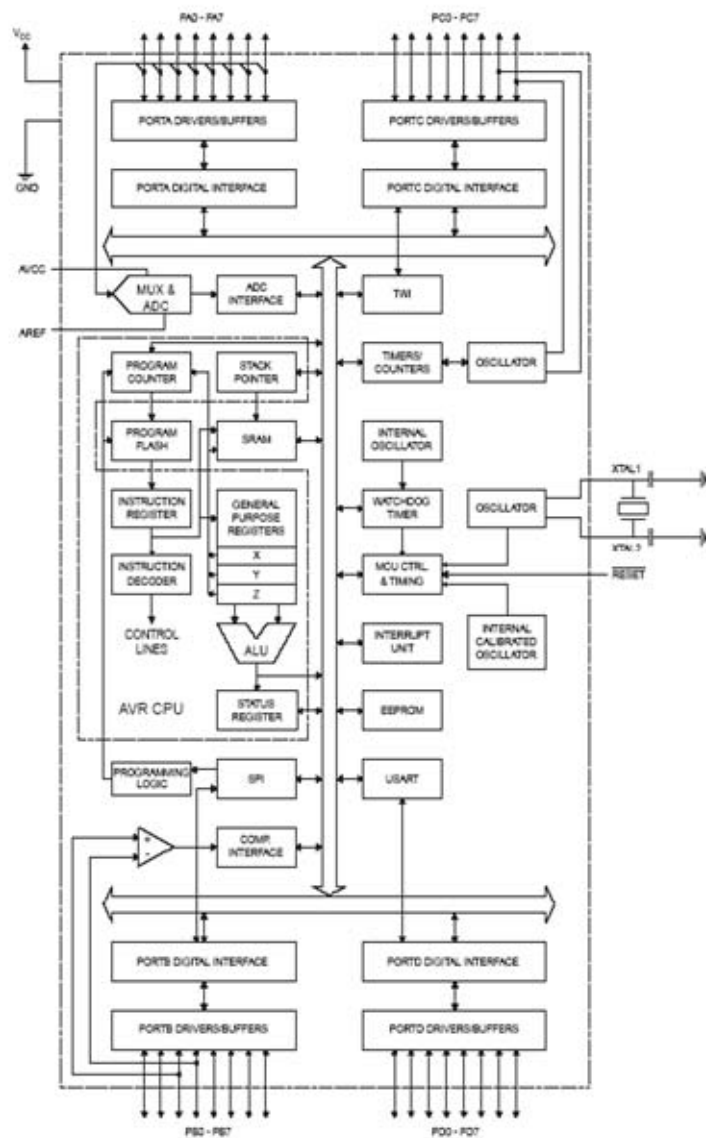
Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Overview

The ATmega8535 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing instructions in a single clock cycle, the ATmega8535 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



Pin Descriptions

V_{CC}	Digital supply voltage.
GND	Ground.
Port A (PA7..PA0)	<p>Port A serves as the analog inputs to the A/D Converter.</p> <p>Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p>
Port B (PB7..PB0)	<p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port B also serves the functions of various special features of the ATmega8535 as listed on page 58.</p>
Port C (PC7..PC0)	<p>Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p>
Port D (PD7..PD0)	<p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega8535 as listed on page 62.</p>
RESET	Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 35. Shorter pulses are not guaranteed to generate a reset.
XTAL1	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
XTAL2	Output from the inverting Oscillator amplifier.
AVCC	AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to V _{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V _{CC} through a low-pass filter.
AREF	AREF is the analog reference pin for the A/D Converter.

LAMPIRAN II

```

/*****

```

```

This program was produced by the
CodeWizardAVR V2.04.2c Evaluation
Automatic Program Generator
© Copyright 1998-2009 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

```

```

Project :
Version :
Date   : 7/14/2010
Author : Freeware, for evaluation and non-commercial use only
Company :
Comments:

```

```

Chip type       : ATmega8535
Program type    : Application
AVR Core Clock frequency: 8.000000 MHz
Memory model    : Small
External RAM size : 0
Data Stack size : 128

```

```

*****/

```

```

#include <mega8535.h>
#include <delay.h>
#include <stdio.h>

```

```

// Alphanumeric LCD Module functions
#asm
    .equ __lcd_port=0x18 ;PORTB
#endasm
#include <lcd.h>

```

```

// 1 Wire Bus functions
#asm
    .equ __w1_port=0x12 ;PORTD
    .equ __w1_bit=2
#endasm

```

```

#define speed1 PORTA.1
#define speed2 PORTA.0

```

```

#include <1wire.h>

```

```

// DS1820 Temperature Sensor functions
#include <ds1820.h>

```

```

// Standard Input/Output functions
#include <math.h>

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

unsigned char devices;
char lcd_buffer[33];
int temp,t1;

void scanning(void);

/* allocate space for ROM codes of the devices hich generate an alarm */
//unsigned char alarm_rom_codes[1][9];

unsigned char rom_codes[1][9];

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

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

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

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

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=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
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=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;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: Off
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 115200
UCSRA=0x00;
UCSRB=0x08;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x05;

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

// LCD module initialization
```

```

lcd_init(16);

// 1 Wire Bus initialization
wl_init();

/* detect how many DS1820/DS18S20 devices
   are connected to the bus and
   store their ROM codes in the rom_codes array */
devices=wl_search(0xf0,rom_codes);

/* display the number */
//printf("%-u DEVICE(S) DETECTED\n\r",devices);

/* if no devices were detected then halt */
if (devices==0) while (1); /* loop forever */

while (1)
{
// Place your code here
// baca sensor suhu
temp=ds1820_temperature_10(&rom_codes[0][0]);

lcd_gotoxy(0,1);

sprintf (lcd_buffer,"suhu:%-i.%-u%",temp/10,abs(temp%10));
lcd_puts(lcd_buffer);

//      lcd_gotoxy(0,1);
//      sprintf (lcd_buffer,"temp:%i",temp);
//      lcd_puts(lcd_buffer);

lcd_gotoxy(0,0);
sprintf (lcd_buffer,"20-25:1,<25:2,");
lcd_puts(lcd_buffer);

//PORTA.1=1;

scaning();

};
}

void scanning(void)
{
//unsigned char suhu;

```

```
//suhu=(t1/10);
PORTA=0;

switch(temp)

{

    case 100:
    speed1=0;
    speed2=0;
    lcd_gotoxy(12,1);
    lcd_putsf("off");
    delay_ms(100);
    break;

    case 105:
    speed1=0;
    speed2=0;
    lcd_gotoxy(12,1);
    lcd_putsf("off");
    delay_ms(100);
    break;

    case 110:
    speed1=0;
    speed2=0;
    lcd_gotoxy(12,1);
    lcd_putsf("off");
    delay_ms(100);
    break;

    case 115:
    speed1=0;
    speed2=0;
    lcd_gotoxy(12,1);
    lcd_putsf("off");
    delay_ms(100);
    break;

    case 120:
    speed1=0;
    speed2=0;
    lcd_gotoxy(12,1);
    lcd_putsf("ff");
    delay_ms(100);
    break;

    case 125:
    speed1=0;
    speed2=0;
    lcd_gotoxy(12,1);
    lcd_putsf("off");
    delay_ms(100);
    break;
```




```
case 130:  
speed1=0;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("off");  
delay_ms(100);  
break;
```

```
case 135:  
speed1=0;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("ff");  
delay_ms(100);  
break;
```

```
case 140:  
speed1=0;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("ff");  
delay_ms(100);  
break;
```

```
case 145:  
speed1=0;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("ff");  
delay_ms(100);  
break;
```

```
case 150:  
speed1=0;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("ff");  
delay_ms(100);  
break;
```

```
case 155:  
speed1=0;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("off");  
delay_ms(100);  
break;
```

```
case 160:  
speed1=0;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("off");  
delay_ms(100);  
break;
```



```
case 165:  
speed1=0;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("off");  
delay_ms(100);  
break;
```

```
case 170:  
speed1=0;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("off");  
delay_ms(100);  
break;
```

```
case 175:  
speed1=0;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("off");  
delay_ms(100);  
break;
```

```
case 180:  
speed1=0;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("off");  
delay_ms(100);  
break;
```

```
case 185:  
speed1=0;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("off");  
delay_ms(100);  
break;
```

```
case 190:  
speed1=0;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("0ff");  
delay_ms(100);  
break;
```

```
case 195:  
speed1=0;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("off");
```



```
delay_ms(100);  
break;
```

```
case 200:  
speed1=1;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("-1-");  
delay_ms(100);  
break;
```

```
case 205:  
speed1=1;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("-1-");  
delay_ms(100);  
break;
```

```
case 210:  
speed1=1;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("-1-");  
delay_ms(100);  
break;
```

```
case 215:  
speed1=1;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("-1-");  
delay_ms(100);  
break;
```

```
case 220:  
speed1=1;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("-1-");  
delay_ms(100);  
break;
```

```
case 225:  
speed1=1;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("-1-");  
delay_ms(100);  
break;
```

```
case 230:  
speed1=1;  
speed2=0;
```



```
lcd_gotoxy(12,1);  
lcd_putsf("-1-");  
delay_ms(100);  
break;
```

```
case 235:  
speed1=1;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("-1-");  
delay_ms(100);  
break;
```

```
case 240:  
speed1=1;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("-1-");  
delay_ms(100);  
break;
```

```
case 245:  
speed1=1;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("-1-");  
delay_ms(100);  
break;
```

```
case 250:  
speed1=1;  
speed2=0;  
lcd_gotoxy(12,1);  
lcd_putsf("-1-");  
delay_ms(100);  
break;
```

```
case 255:  
speed1=0;  
speed2=1;  
lcd_gotoxy(12,1);  
lcd_putsf("-2-");  
delay_ms(100);  
break;
```

```
case 260:  
speed1=0;  
speed2=1;  
lcd_gotoxy(12,1);  
lcd_putsf("-2-");  
delay_ms(100);  
break;
```

```
case 265:
```



```
speed1=0;
speed2=1;
lcd_gotoxy(12,1);
lcd_putsf("-2-");
delay_ms(100);
break;
```

```
case 270:
speed1=0;
speed2=1;
lcd_gotoxy(12,1);
lcd_putsf("-2-");
delay_ms(100);
break;
```

```
case 275:
speed1=0;
speed2=1;
lcd_gotoxy(12,1);
lcd_putsf("-2-");
delay_ms(100);
break;
```

```
case 280:
speed1=0;
speed2=1;
lcd_gotoxy(12,1);
lcd_putsf("-2-");
delay_ms(100);
break;
```

```
case 285:
speed1=0;
speed2=1;
lcd_gotoxy(12,1);
lcd_putsf("-2-");
delay_ms(100);
break;
```

```
case 290:
speed1=0;
speed2=1;
lcd_gotoxy(12,1);
lcd_putsf("-2-");
delay_ms(100);
break;
```

```
case 295:
speed1=0;
speed2=1;
lcd_gotoxy(12,1);
lcd_putsf("-2-");
delay_ms(100);
break;
```

```
case 300:
```



```
speed1=0;
speed2=1;
lcd_gotoxy(12,1);
lcd_putsf("-2-");
delay_ms(100);
break;
```

```
case 305:
speed1=0;
speed2=1;
lcd_gotoxy(12,1);
lcd_putsf("-2-");
delay_ms(100);
break;
```

```
case 310:
speed1=0;
speed2=1;
lcd_gotoxy(12,1);
lcd_putsf("-2-");
delay_ms(100);
break;
```

```
case 315:
speed1=0;
speed2=1;
lcd_gotoxy(12,1);
lcd_putsf("-2-");
delay_ms(100);
break;
```

```
case 320:
speed1=0;
speed2=1;
lcd_gotoxy(12,1);
lcd_putsf("-2-");
delay_ms(100);
break;
```

```
case 325:
speed1=0;
speed2=1;
lcd_gotoxy(12,1);
lcd_putsf("-2-");
delay_ms(100);
break;
```

```
case 330:
speed1=0;
speed2=1;
lcd_gotoxy(12,1);
lcd_putsf("-2-");
delay_ms(100);
break;
```

```
case 335:
```

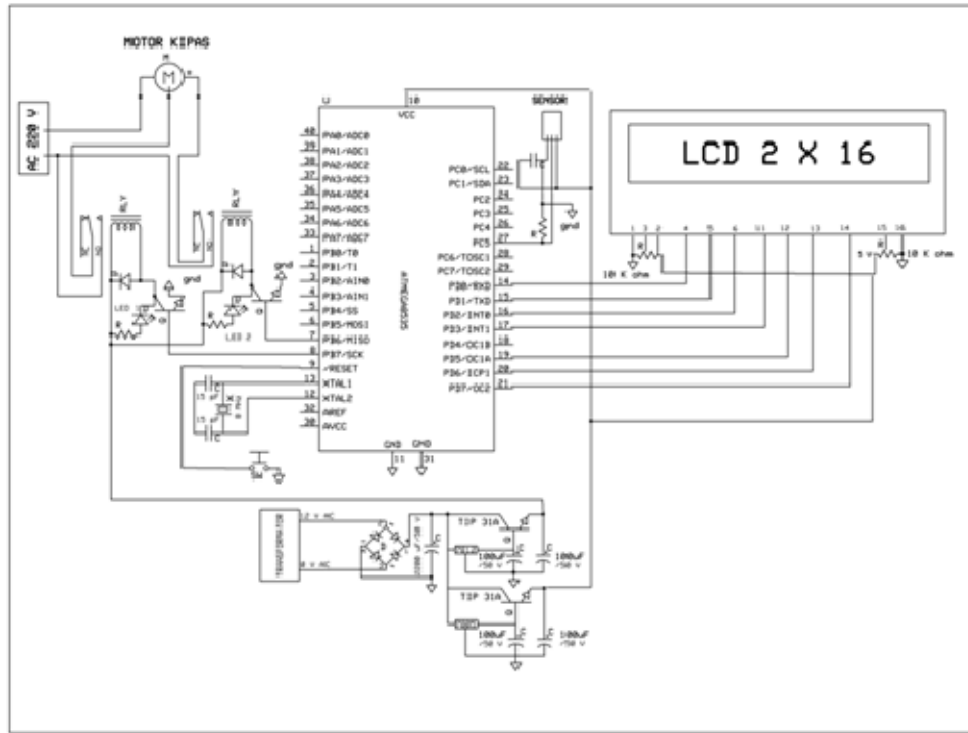


```
speed1=0;  
speed2=1;  
lcd_gotoxy(12,1);  
lcd_putsf("-2-");  
delay_ms(100);  
break;
```

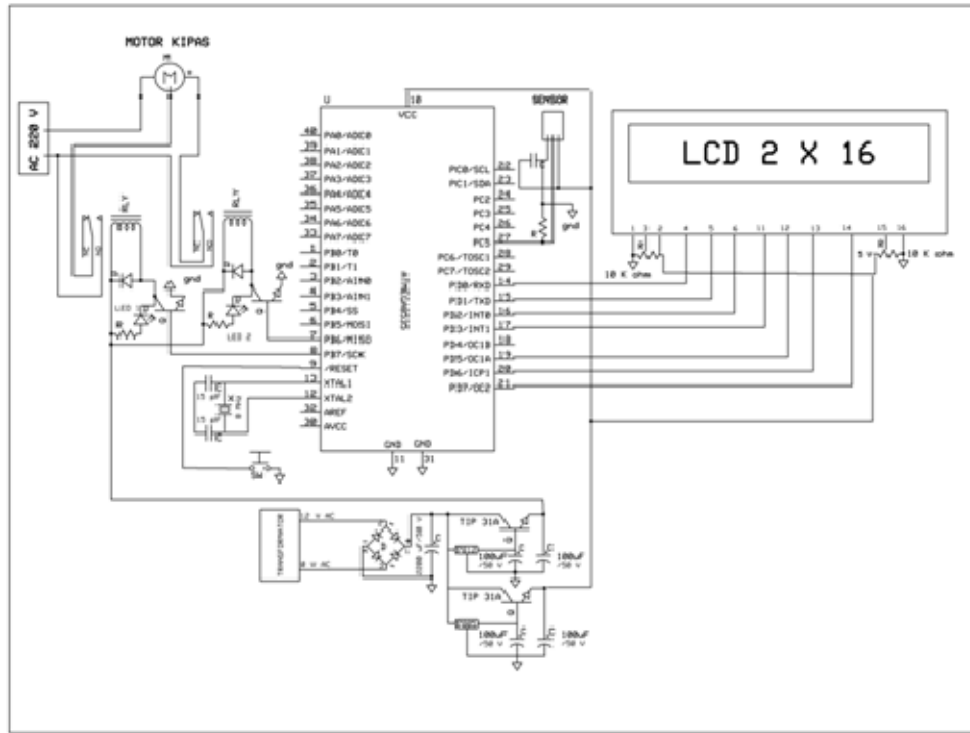
```
case 340:  
speed1=0;  
speed2=1;  
lcd_gotoxy(12,1);  
lcd_putsf("-2-");  
delay_ms(100);  
break;
```

```
case 345:  
speed1=0;  
speed2=1;  
lcd_gotoxy(12,1);  
lcd_putsf("-2-");  
delay_ms(100);  
break;
```





Rangkaian Lengkap Pengaturan Kipas Berbasis Mikrokontroler Dengan Menggunakan Sensor Suhu



LAMPIRAN IV



Alat Pengaturan Kipas Berbasis Mikrokontroler Dengan Menggunakan Sensor Suhu.



Rangkaian Keseluruhan



Rangkaian Relai



Rangkaian Mikrokontroler