



**PENERAPAN SENSOR OPTOCOUPLER PADA ALAT
PENGUKUR KECEPATAN ANGIN BERBASIS
MIKROKONTROLER AVR ATmega8535**

skripsi

disajikan sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains Program

Studi Fisika

oleh

Yudistiro Ardi Nugroho

NIM 4250404032

Jurusan Fisika

Fakultas Matematika Dan Ilmu Pengetahuan Alam

Universitas Negeri Semarang

2011

PERSETUJUAN PEMBIMBING

Skripsi ini yang berjudul “*Penerapan Sensor Optocoupler Pada Alat Pengukur Kecepatan Angin Berbasis Mikrokontroler AVR ATmega8535*”, telah disetujui oleh pembimbing untuk diajukan ke sidang panitia ujian skripsi.

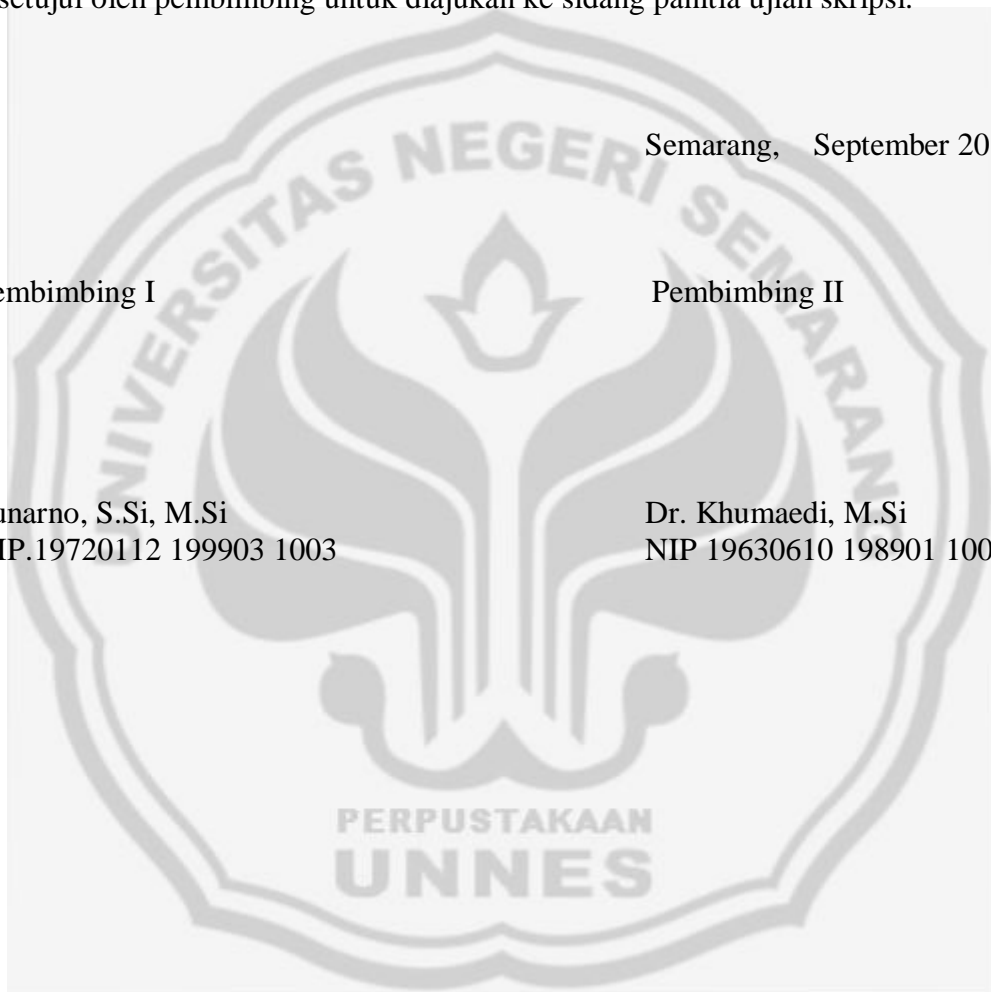
Semarang, September 2011

Pembimbing I

Pembimbing II

Sunarno, S.Si, M.Si
NIP.19720112 199903 1003

Dr. Khumaedi, M.Si
NIP 19630610 198901 1002



PERNYATAAN

Saya menyatakan bahwa skripsi ini bebas plagiat, dan apabila di kemudian hari terbukti terdapat plagiat dalam skripsi ini, maka saya bersedia menerima sanksi sesuai ketentuan peraturan perundang-undangan.

Semarang, 6 September 2011

Penulis

Yudistiro Ardi Nugroho

NIM 4250404032



PENGESAHAN

Skripsi yang berjudul

Penerapan Sensor Optocoupler Pada Alat Pengukur Kecepatan Angin
Berbasis Mikrokontroler AVR Atmega8535

disusun oleh

Yudistiro Ardi Nugroho

4250404032

telah dipertahankan di hadapan sidang Panitia Ujian Skripsi FMIPA UNNES,
pada tanggal: 8 September 2011.

Panitia:

Ketua

Sekretaris

Drs. Kasmadi Imam S, M.S.
NIP. 1951111519791001

Dr. Putut Marwoto, M.S.
NIP. 196308211988031004

Ketua Penguji

Dr. Sulhadi, M.Si.
NIP. 19710816 199802 1 001

Anggota Penguji/
Pembimbing Utama

Anggota Penguji/
Pembimbing Pendamping

Sunarno, S.Si, M.Si.
NIP. 197201121999031003

Dr. Khumaedi, M.Si.
NIP 196306101989011002

PERSEMBAHAN

- (1) Untuk Ibu dan adik-adik**
- (2) Untuk teman-teman fisika**
- (3) Untuk teman-teman kos**



MOTTO

- (1) Sesungguhnya, rahmat Allah amat dekat kepada orang-orang yang berbuat baik(QS.Al A'raf: 56).
- (2) Tugas kita bukanlah untuk berhasil. Tugas kita adalah untuk mencoba, karena didalam mencoba itulah kita menemukan dan belajar membangun kesempatan untuk berhasil(Mario Teguh).
- (3) Jangan takut akan kegagalan manakala telah kita rencanakan, tapi takutlah jika kita tidak pernah mau berusaha berbuat lebih baik lagi.



PRAKATA

Puji syukur Alhamdulillah penulis panjatkan kehadiran Allah SWT atas berkah serta rahmat-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul PENERAPAN SENSOR OPTOCOUPLER PADA ALAT PENGUKUR KECEPATAN ANGIN BERBASIS MIKROKONTROLER AVR ATmega8535.

Selama penulis melakukan penelitian dalam pembuatan skripsi ini, penulis telah banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada:

- (1) Bapak Dr. Putut Marwoto, M.S selaku Ketua Jurusan Fisika.
- (2) Bapak Dr. Agus Yulianto, M.Si selaku Ketua Prodi Jurusan Fisika.
- (3) Bapak Sunarno, M.Si selaku Pembimbing Utama yang telah memberikan bimbingan dan masukan dalam penyelesaian skripsi ini.
- (4) Bapak Dr. Khumaedi, M.Si selaku Pembimbing Pendamping yang telah memberikan bimbingan dan masukan dalam penyelesaian skripsi ini.
- (5) Bapak Dr. Sulhadi, M.Si. selaku Penguji.
- (6) Teman-teman Fisika 2004; Joko, Edi, Rudi, Deny dan Yusuf serta semua pihak yang telah berpartisipasi dalam proses penulisan skripsi.

Semoga Allah SWT membalas amal kebaikan yang telah diberikan dan apa yang telah penulis uraikan dalam skripsi ini dapat bermanfaat bagi penulis khususnya dan para pembaca pada umumnya.

Semarang, 6 September 2011

Penulis

ABSTRAK

Nugroho, Y, A. Penerapan Sensor Optocoupler Pada Pengukur Kecepatan Angin Berbasis Mikrokontroler AVR Atmega8535. Skripsi, Jurusan Fisika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Semarang. Pembimbing Utama Sunarno, S.Si, M.Si. dan Pembimbing Pendamping Dr. Khumaedi, M.Si.

Kata kunci: Sensor *optocoupler*, alat pengukur kecepatan angin, mikrokontroler AVR Atmega8535.

Penelitian ini dilatarbelakangi adanya keinginan membuat sistem pengukur kecepatan angin digital. Sistem pengukuran kecepatan angin yang digunakan berbasis Mikrokontroler AVR ATmega8535, yang merupakan piranti alat ukur portable, murah dan handal yang diharapkan mampu diaplikasikan pada komputer untuk pengukuran secara terus-menerus maupun penyimpanan data historis dengan memanfaatkan *internal timer Personal Computer* dengan bantuan program bahasa komputer. Untuk mendeteksi jumlah putaran mangkok-mangkok pada pengukur kecepatan angin digunakan sensor optocoupler. Dalam penelitian ini metode yang digunakan dibagi menjadi dua yaitu perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras meliputi perancangan alat pengukur kecepatan angin, rangkaian sensor *optocoupler* dan rangkaian mikrokontroler, sedangkan untuk perancangan perangkat lunak adalah pemrograman AVR menggunakan bahasa assembly.

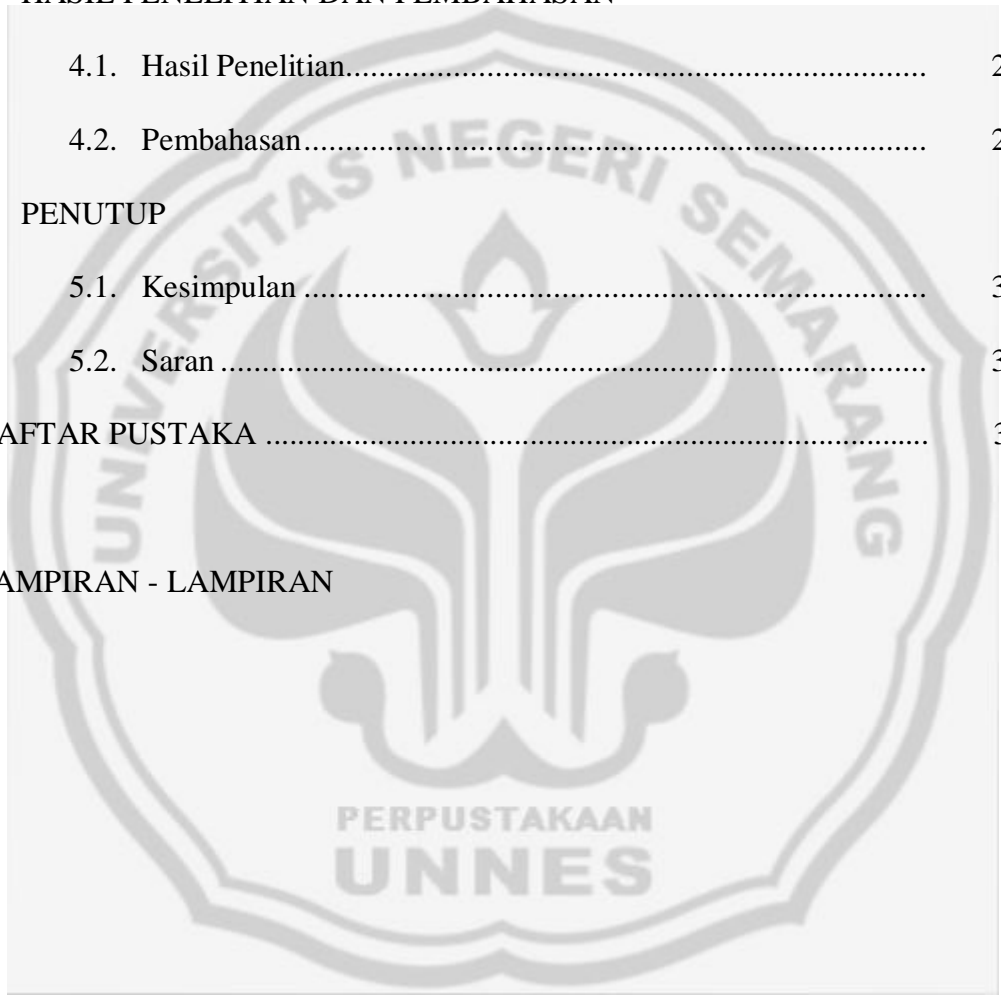
Hasil penelitian ini adalah suatu alat pengukur kecepatan angin dengan sensor *optocoupler* berbasis mikrokontroler AVR Atmega8535 dan dapat dijadikan alat ukur yang tepat jika setiap pembacaan datanya ditambahkan faktor

kalibrasi $v = \frac{100}{37} x$.

DAFTAR ISI

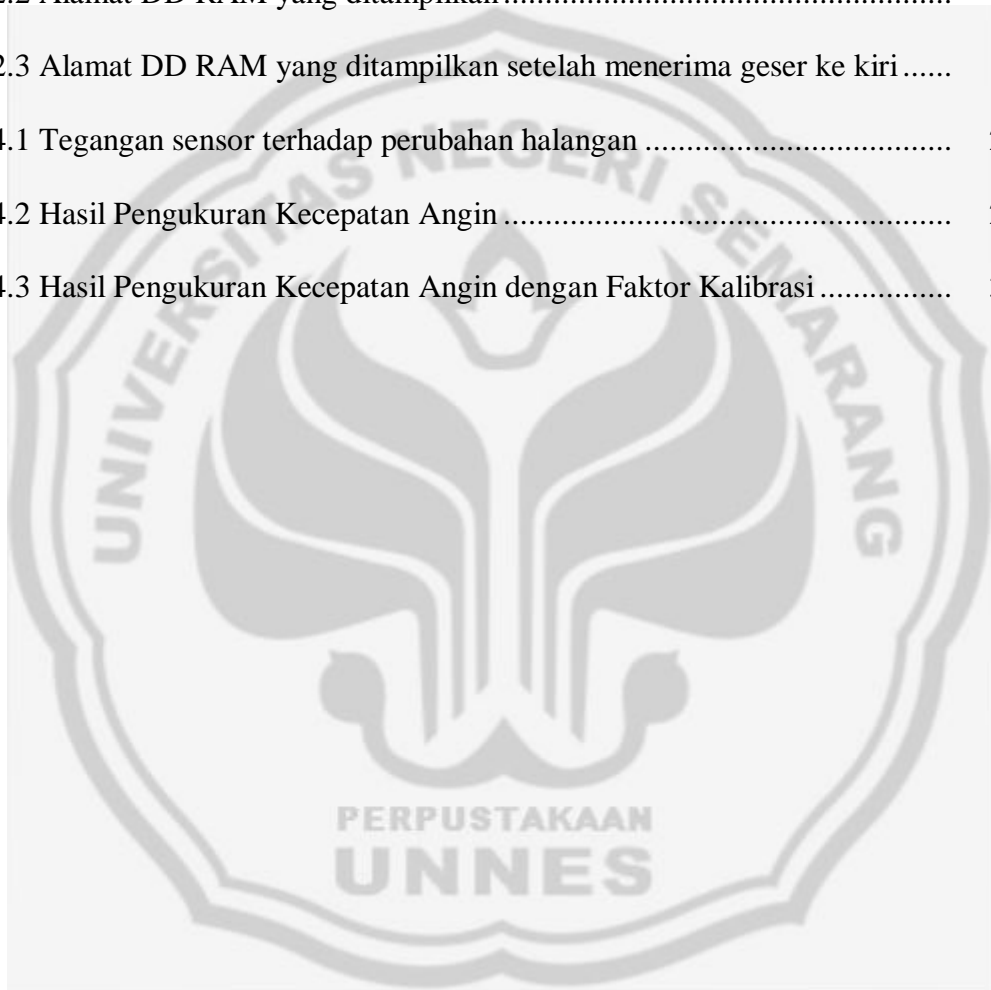
	Halaman
PRAKATA	vii
ABSTRAK	viii
DARTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiii
BAB	
1. PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Sistematika Skripsi	3
2. TINJAUAN PUSTAKA	
2.1 Karakteristik Angin	5
2.2. Tranduser	6
2.3. Sensor	6
2.4. Sensor Optocoupler	6
2.5. Mikrokontroler AVR Atmega8535	8
2.6. IC 74LS14.....	15
2.7. LCD M1632	16

3. METODE PENELITIAN	
3.1. Metode Penelitian	20
3.2. Perencanaan Rangkaian.....	20
3.3. Perancangan Sistem.....	20
4. HASIL PENELITIAN DAN PEMBAHASAN	
4.1. Hasil Penelitian.....	25
4.2. Pembahasan.....	28
5. PENUTUP	
5.1. Kesimpulan	32
5.2. Saran	32
DAFTAR PUSTAKA	33
LAMPIRAN - LAMPIRAN	



DAFTAR TABEL

Tabel	Halaman
2.1 Konfigurasi kaki M1632	17
2.2 Alamat DD RAM yang ditampilkan	19
2.3 Alamat DD RAM yang ditampilkan setelah menerima geser ke kiri	19
4.1 Tegangan sensor terhadap perubahan halangan	26
4.2 Hasil Pengukuran Kecepatan Angin	27
4.3 Hasil Pengukuran Kecepatan Angin dengan Faktor Kalibrasi	31

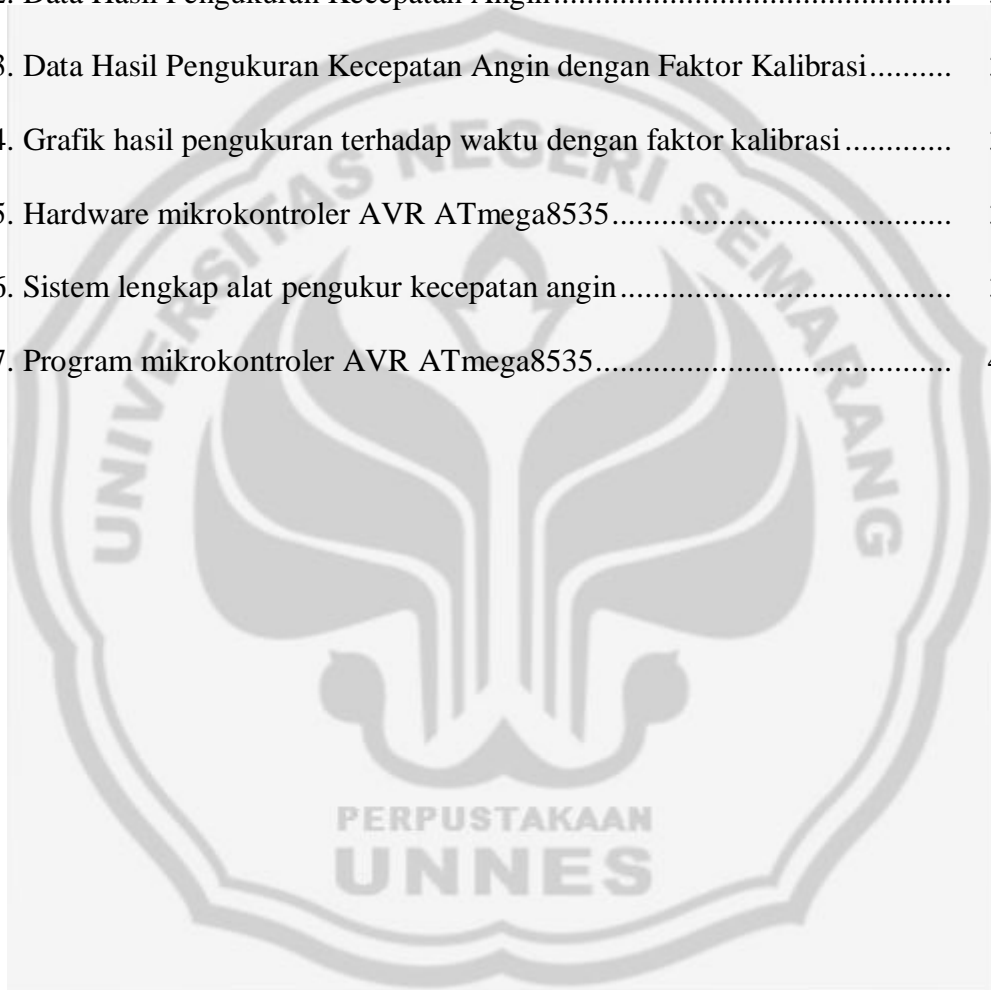


DAFTAR GAMBAR

Gambar	Halaman
2.1 Sensor Optocoupler	7
2.2 Skematik Optocoupler.....	8
2.3 Komponen Dasar Sistem Kontrol.....	9
2.4 Blok diagram mikrokontroler AVR.....	9
2.5 Konfigurasi Pin ATmega8535 PDIP.....	12
2.6 Konfigurasi Pin ATmega8535 PLCC	12
2.7 Konfigurasi Pin ATmega8535 TQFPMLF.....	13
2.8 Skema IC 74LS14	16
3.1 Sistem Pengukuran Kecepatan Angin.....	21
3.2 Rancangan Pengukur Kecepatan Angin.....	22
3.3 Flowchart Bahasa Pemrograman	23
3.4 Tampilan LCD	24
4.1 Pengujian tegangan sensor optocoupler	25
4.2 Rangkaian Sensor Optocoupler	28
4.3 Grafik hasil pengukuran terhadap waktu dengan faktor kalibrasi.....	31

DAFTAR LAMPIRAN

Lampiran	Halaman
1. Data Pengujian Sensor	34
2. Data Hasil Pengukuran Kecepatan Angin.....	35
3. Data Hasil Pengukuran Kecepatan Angin dengan Faktor Kalibrasi.....	36
4. Grafik hasil pengukuran terhadap waktu dengan faktor kalibrasi.....	37
5. Hardware mikrokontroler AVR ATmega8535.....	38
6. Sistem lengkap alat pengukur kecepatan angin.....	39
7. Program mikrokontroler AVR ATmega8535.....	40



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Sistem pengukuran mempunyai peranan penting dalam kehidupan manusia, khususnya untuk mengetahui nilai dari besaran fisis. Pengukuran adalah membandingkan besaran fisis yang belum diketahui nilainya dengan besaran fisis yang telah diketahui nilainya. Hasil pengukuran bersifat universal yang dapat dinyatakan dalam angka dan satuan atau nilai dan satuan, misalnya pengukuran kecepatan, berat, suhu dan lain-lain.

Kecepatan termasuk dalam pengukuran tunggal, yang hanya mungkin dilakukan satu kali saja karena obyek pengukuran tidak dapat diulang. Alat ukur yang digunakan dapat memakai berbagai macam alat, sesuai dengan fungsi penggunaannya, dalam bahasan ini untuk mengukur kecepatan angin.

Perkembangan teknologi saat ini telah sampai pada penentuan cuaca dan iklim yang menggunakan satelit. Dalam skala makro satelit dapat digunakan untuk mempelajari pembentukan awan, suhu dan perkiraan cuaca, sedangkan dalam skala mikro dapat digunakan untuk mendeteksi kecepatan angin atau suhu di daerah tertentu. Namun, pemanfaatan satelit dalam skala mikro, misalnya untuk mendeteksi kecepatan angin, belum dapat diandalkan. Salah satu upaya untuk mengatasi kendala tersebut adalah dengan pembuatan alat ukur otomatis.

Alat ukur yang digunakan harus bersifat peka, kuat, sederhana dan teliti.

Secara garis besar ada dua macam metode dalam pembacaan alat ukur, yaitu:

- (1) *Recording* yaitu alat yang dapat mencatat data secara terus-menerus, sejak pemasangan hingga pergantian alat berikutnya.

Contoh : barograf dan anemograf

- (2) *Non recording* yaitu alat yang digunakan bila datanya harus dibaca pada saat-saat tertentu untuk memperoleh data. Contoh: barometer, termometer dan anemometer.

Anemometer adalah alat yang digunakan untuk mengukur kecepatan angin. Satuan dari kecepatan angin adalah Knots (Skala Beaufort) dan dapat juga dalam m/s. Alat pengukur kecepatan angin dikatakan baik jika memberikan hasil pengukuran yang lebih akurat. Pembuatan alat pengukur kecepatan angin dapat memanfaatkan teknologi mikrokontroler. Sama halnya dengan sistem komputer, mikrokontroler mempunyai kemampuan untuk diprogram sesuai dengan kebutuhan pembuatan alat. Pada alat pengukur kecepatan angin terdapat baling-baling atau mangkok yang berputar sesuai dengan arah angin. Makin besar kecepatan angin, makin cepat putaran mangkok-mangkok tersebut. Dari jumlah putaran dalam satu detik maka dapat diketahui kecepatannya. Untuk mendeteksi jumlah putaran mangkok-mangkok pada anemometer digunakan sensor *optocoupler*.

1.2 Perumusan Masalah

Permasalahan yang dapat diambil dalam penelitian ini adalah bagaimana merancang alat pengukur kecepatan angin (*anemometer*) dengan sensor *optocoupler* menggunakan mikrokontroler AVR ATmega8535.

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk menghasilkan alat pengukur kecepatan angin (*anemometer*) dengan sensor *optocoupler* yang berbasis pada mikrokontroler AVR ATmega8535.

1.4 Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian ini adalah dapat mengukur kecepatan angin pada suatu daerah.

1.5 Sistematika Skripsi

Sistematika dalam penelitian ini disusun dengan tujuan agar pokok-pokok masalah yang dibahas dapat secara urut dan terarah serta jelas. Sistematika skripsi terdiri dari tiga bagian utama yaitu : (1) bagian awal (pawacana), (2) bagian pokok (nas) dan (3) bagian akhir.

1.6.1 Prawacana

Prawacana terdiri atas judul, halaman kosong, pernyataan keaslian tulisan, pengesahan, persembahan, motto, prakata, abstrak, daftar isi, daftar gambar, daftar tabel, daftar gambar dan daftar lampiran.

1.6.2 Bagian pokok (nas)

Bagian isi skripsi di bagi menjadi 5 (lima) bab yaitu :

1.6.2.1 Bab 1 Pendahuluan

Bab ini menyajikan gagasan pokok yang paling sedikit terdiri atas empat bagian:

(1) latar belakang, (2) masalah, (3) tujuan penelitian dan (4) sistematika skripsi.

1.6.2.2 Bab 2 Tinjauan Pustaka

Bagian ini berisi kajian teoriterdiri dari kajian pustaka dan hasil-hasil penelitian terdahulu yang menjadi kerangka pikir penyelesaian masalah penelitian yang disajikan kedalam sub-bab.

1.6.2.3 Bab 3 Metode Penelitian

Bab ini menyajikan gagasan pokok yang paling sedikit terdiri atas: desain penelitian, subjek (sampel dan populasi) dan lokasi penelitian, variabel penelitian dan indikatornya, pengambilan data (bahan, alat atau instrumen, teknik pengambilan data penelitian), dan analisis data penelitian.

1.6.2.4 Bab 4 Hasil Penelitian dan Pembahasan

Bab ini berisi hasil analisis data dan pembahasannya yang disajikan dalam rangka menjawab permasalahan penelitian.

1.6.2.5 Bab V. Penutup

Bab ini berisi tentang simpulan dan saran.

1.6.3 Bagian akhir skripsi

Bagian akhir skripsi sekurang-kurangnya terdiri atas daftar pustaka dan lampiran.

Bagian akhir skripsi ini terkait dengan nas, karenanya setiap pustaka dan lampiran yang ditulis di bagian akhir harus dirujuk dalam nas.

BAB 2

TINJAUAN PUSTAKA

2.1 Karakteristik angin

Angin adalah udara yang bergerak dari tekanan tinggi ke tekanan rendah atau dari suhu udara yang rendah ke suhu udara yang tinggi. Angin terjadi karena adanya perbedaan tekanan udara atau perbedaan suhu udara pada suatu daerah atau wilayah. Hal ini berkaitan dengan besarnya energi panas matahari yang di terima oleh permukaan bumi. Pada suatu wilayah, daerah yang menerima energi panas matahari lebih besar akan mempunyai suhu udara yang lebih panas dan tekanan udara yang cenderung lebih rendah. Sehingga akan terjadi perbedaan suhu dan tekanan udara antara daerah yang menerima energi panas lebih besar dengan daerah lain yang lebih sedikit menerima energi panas, akibatnya akan terjadi aliran udara pada wilayah tersebut. Angin mempunyai energi kinetik yang diperoleh dari perkalian antara massa dan kecepatan pergerakan udara.

Menurut *Tipler, P.A. (1998:158)*, energi kinetik sebuah benda dengan massa m dan bergerak dengan kecepatan v adalah:

$$K = \frac{1}{2}mv^2$$

Jika ditinjau suatu luasan $A(\text{m}^2)$ yang dilalui angin dengan kecepatan v (ms^{-2}) maka massa angin yang lewat pada luasan tersebut per satuan waktu sama dengan ρA , ρ adalah kerapatan udara berkisar antara $0,9 \text{ kg/m}^3$ sampai dengan

1,4 kg/m³. Oleh karena itu, energi per satuan waktu (daya) yang dimiliki oleh angin adalah:

$$P = 0.5 \times \rho \times Av^3 (\text{watt})$$

Terlihat bahwa energi yang dimiliki oleh angin adalah fungsi pangkat tiga dari kecepatannya.

2.2 Transduser

Transduser adalah alat yang mengubah energi dari satu bentuk ke bentuk lain. Transduser dapat dibagi menjadi dua kelas yaitu transduser input dan transduser output. Transduser input listrik mengubah energi non listrik, misalnya suara atau sinar menjadi tenaga listrik. Transduser output listrik bekerja pada urutan yang sebaliknya, yaitu mengubah energi listrik pada bentuk energi non listrik menurut *Petruzela (1996:157)*.

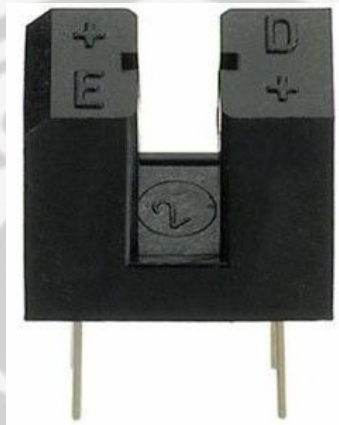
2.3 Sensor

Sensor adalah alat yang digunakan untuk mendeteksi atau mengukur nilai suatu besaran fisis tertentu. Sensor adalah jenis transduser yang digunakan untuk mengubah variasi mekanis, magnetis, panas, sinar dan kimia menjadi tegangan dan arus listrik. Sensor biasanya dikategorikan melalui pengukur dan memegang peranan penting dalam pengendalian proses fabrikasi modern menurut *Petruzela (1996:157)*.

2.4 Sensor Optocoupler

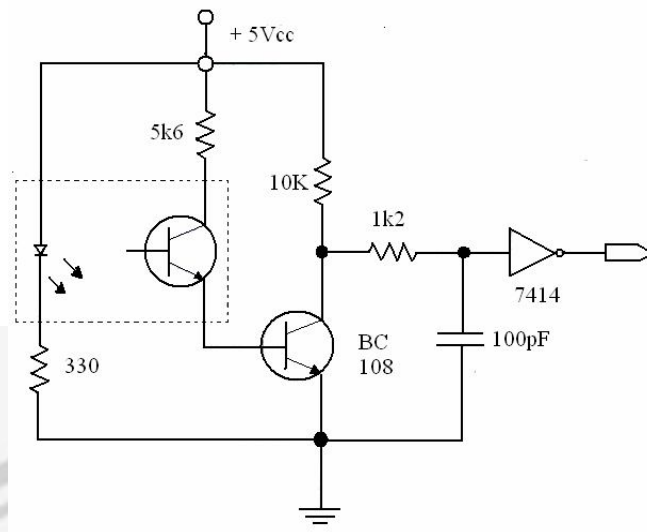
Sensor *Optocoupler* merupakan alat yang digunakan untuk mendeteksi perubahan sinar inframerah, seperti yang terlihat pada Gambar 2.1. Sensor ini banyak dipakai untuk mendeteksi jarak ataupun pergerakan suatu benda dengan

cara memberikan kisi-kisi ataupun baling-baling sehingga akan terdapat celah dan penghalang. Cara kerja dari sensor *optocoupler* adalah bila terhalang maka *output* akan open, dan bila tidak terhalang *output* akan *short*. Dengan cara kerja tersebut, sinar inframerah akan putus-putus dan menimbulkan pulsa-pulsa listrik. Pulsa-pulsa itu kemudian dapat diolah dan ditangkap oleh mikrokontroler.



Gambar 2.1 Sensor Optocoupler
(soselectronic.com)

Bagian dari sensor *optocoupler* ini adalah : sebuah led merah biasa atau led inframerah sebagai *transmitter* dan sebuah fototransistor sebagai *receiver*. Pada bagian *transmitter* dapat kita hubungkan ke tegangan yang cukup untuk menghidupkan led dan bagian *receiver* dihubungkan secara seri ke sumber tegangan dan lainnya menjadi terminal keluaran. Berikut ini merupakan skema dari sensor *optocoupler*.



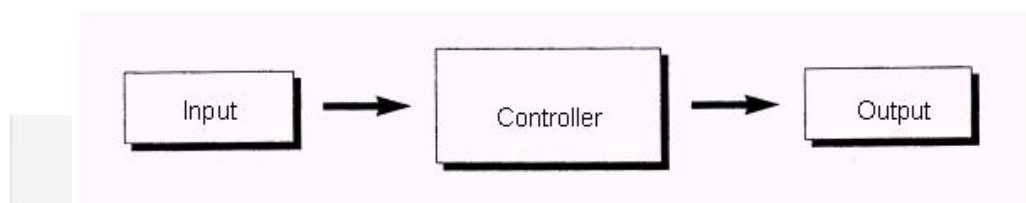
Gambar 2.2 Skematik Optocoupler

(soselectronic.com)

2.5 Mikrokontroler AVR ATmega8535

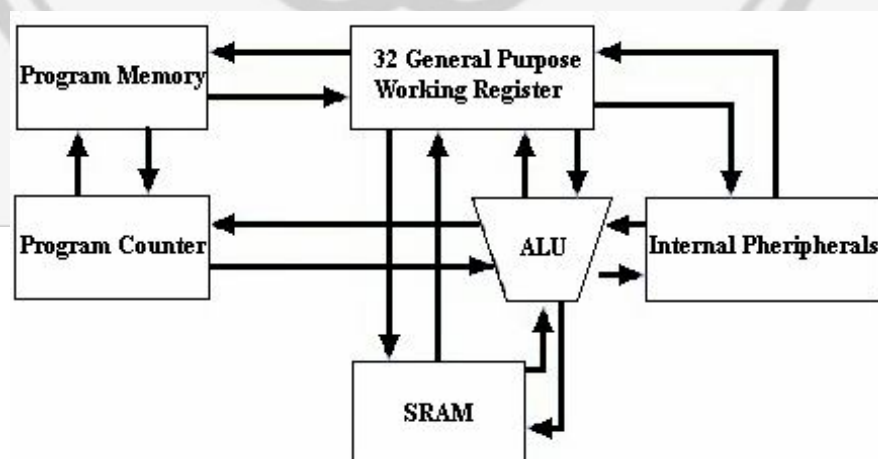
Kontrol dapat didefinisikan sebagai sesuatu yang dilakukan terhadap suatu target agar bisa berjalan dengan benar sesuai dengan tujuannya. Target biasanya disebut dengan rencana, dan hal yang harus dilakukan agar sesuai dengan rencana disebut dengan *controller*. Kelompok yang terdiri dari bagian dan komponen yang dipakai agar target bisa terlaksana disebut dengan sistem. Sistem kontrol adalah sesuatu yang menghubungkan antara komponen atau sistem dengan sistem lainnya agar bisa saling berhubungan. Sistem artinya kombinasi secara sistematis, dan dipakai untuk menjalankan sistem lainnya. Misalnya untuk mengendarai mobil, reaksi berkendara mengikuti kontrol, mobil ke perakitannya dan pengemudi ke *controller*. Kombinasi elemen-elemen tersebut bisa disebut dengan sistem kontrol. Sama seperti yang mengelilingi kita setiap harinya, yang mungkin dikontrol oleh rencana kita sendiri dengan melakukan pengontrolan. Bila

seseorang melakukan peran dalam pengendalian, maka disebut dengan kontrol manual dan apabila pengendolannya secara otomatis, maka disebut dengan kontrol otomatis.



Gambar 2.3 Komponen Dasar Sistem Kontrol

Mikrokontroler yaitu teknologi semikonduktor dengan kandungan transistor yang lebih banyak namun hanya membutuhkan ruang yang kecil, dapat digunakan untuk membuat suatu aplikasi. Pada mikrokontroler, program kontrol disimpan dalam ROM yang ukurannya relatif lebih besar, sementara RAM digunakan sebagai tempat penyimpanan sementara, termasuk register-register yang digunakan pada mikrokontroler yang bersangkutan. Salah satu keluarga dari mikokontroler 8 bit AVR adalah mikrokontroler ATmega8535.



Gambar 2.4 Blok diagram mikrokontroler AVR

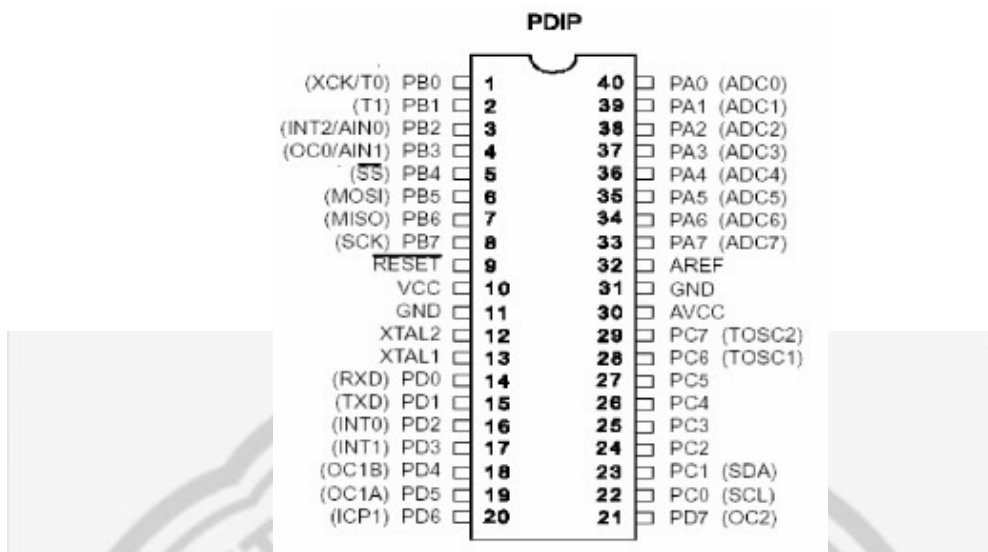
Bagian-bagian blok diagram di atas dapat dijelaskan sebagai berikut:

- (1) ALU (*Arithmetic Logic Unit*) adalah *processor* yang bertugas mengeksekusi (eksekutor) kode program yang ditunjuk oleh *program counter*
- (2) Program Memori adalah memori Flash PEROM yang bertugas menyimpan program (*software*) yang kita buat dalam bentuk kode-kode program (berisi alamat memori beserta kode program dalam ruangan memori alamat tersebut) yang telah dicompile berupa bilangan heksa atau biner
- (3) *Program Counter* (PC) adalah komponen yang bertugas menunjukkan ke ALU alamat program memori yang harus diterjemahkan kode programnya dan dieksekusi. Sifat dari PC adalah linier artinya menghitung naik satu bilangan yang bergantung alamat awalnya. Misalnya jika isi PC 0x000 maka naik satu menjadi 0x001 yang berarti menyuruh ALU mengeksekusi kode program yang berada pada alamat 0x001 program memori. Jika isi PC dari 0x002 dipaksa (instruksi lompatan) 0x02A maka akan naik satu menjadi 0x02B dan melakukan tugasnya begitu seterusnya
- (4) 32 *General Purpose Working Register* (GPR) adalah *register file* atau register kerja (R0-R31) yang mempunyai ruangan 8-bit. Tugas GPR adalah tempat ALU melibatkan GPR. GPR terbagi dua yaitu kelompok atas (R16-R31) dan kelompok bawah (R0-R15), di mana kelompok bawah tidak bisa digunakan untuk mengakses data secara langsung (*imidiety*) data konstan seperti instruksi *assembly* LDI, dan hanya bisa digunakan antar-register, SRAM, atau register I/O (*register port*). Sedangkan kelompok

atas sama dengan kelompok bawah hanya mempunyai kelebihan dapat mengakses data secara langsung (*imidiety*) data konstan. Kelebihan lain dari GPR adalah terdapat register pasangan yang digunakan untuk pointer (penunjuk ke alamat tertentu XH:XL(R27:R:26), YH:YL(R29:R28), ZH:ZL(R31:R30), hanya *register pointer Z* yang dapat digunakan untuk menunjuk ke alamat memori program

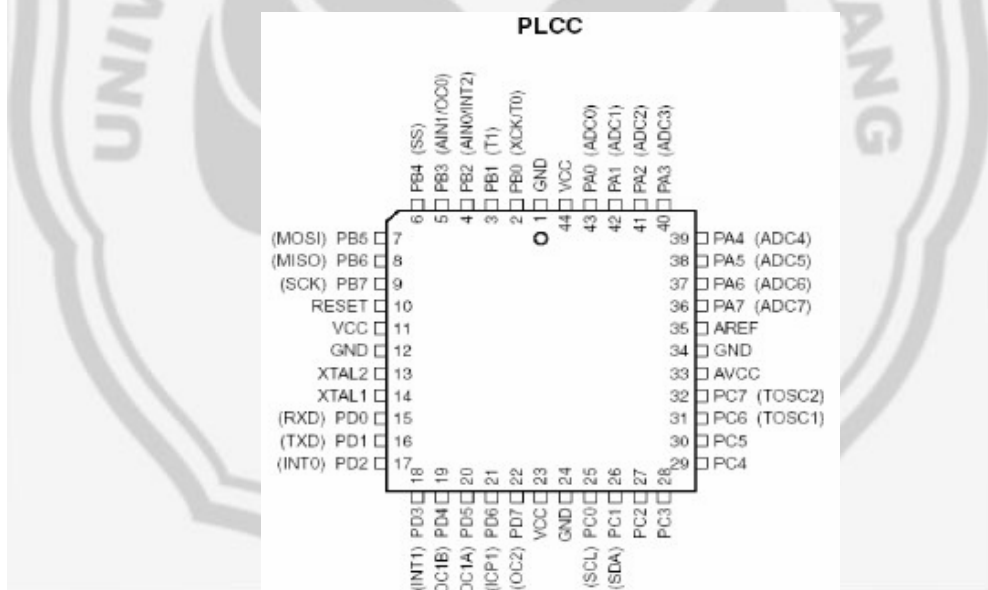
(5) *Static Random Access Memory* (SRAM) adalah RAM yang bertugas menyimpan data sementara sama seperti RAM pada umumnya mempunyai alamat dan ruang data. Alamat terakhir dari SRAM bergantung pada kapasitas SRAM, biasanya sudah didefinisikan pada file header dengan nama RAMEND, jadi kita tidak perlu mengingat alamat SRAM yang terakhir, pakai saja RAMEND. RAMEND biasanya digunakan untuk membuat *stack* (alamat terakhir dari SRAM). Dalam bahasa C, pembuatan *stack* menjadi tanggungan *compiler*

(6) *Internal Pheripheral* adalah peralatan/modul internal yang ada dalam mikrokontroler seperti saluran I/O, interupsi eksternal, *Timer/Counter*, USART, EEPROM dan lain-lain. Tiap peralatan internal mempunyai *register port* (*register I/O*) yang mengendalikan peralatan internal tersebut. Kata-kata port dan I/O di atas bukan hanya pin input atau output tetapi semua peralatan internal yang ada di dalam chip, di sini disebut port atau I/O (dengan kata lain di luar CPU adalah I/O walaupun kenyataanya berada dalam chip).



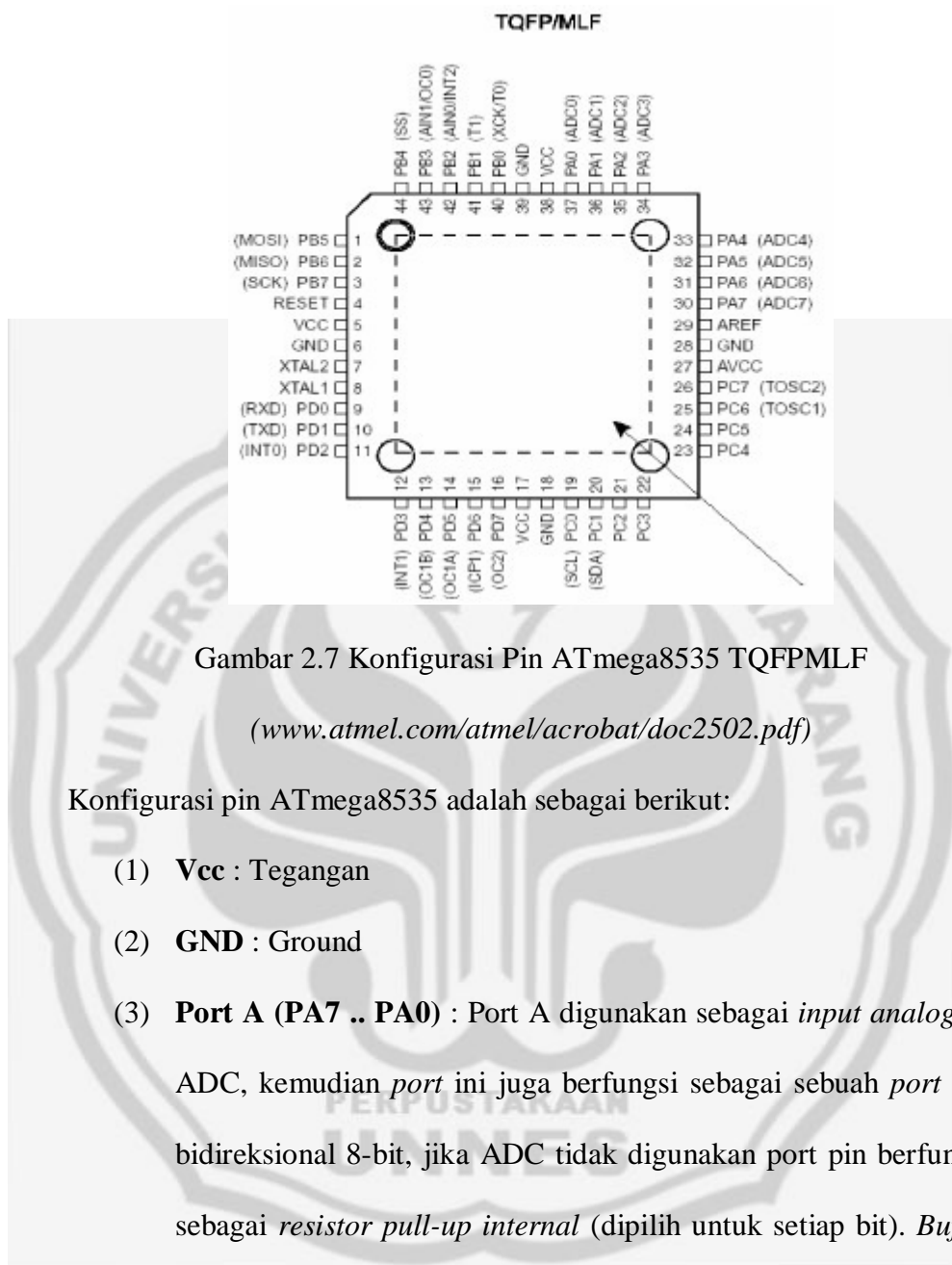
Gambar 2.5 Konfigurasi Pin ATmega8535 PDIP

(www.atmel.com/atmel/acrobat/doc2502.pdf)



Gambar 2.6 Konfigurasi Pin ATmega8535 PLCC

(www.atmel.com/atmel/acrobat/doc2502.pdf)



Gambar 2.7 Konfigurasi Pin ATmega8535 TQFPMLF

(www.atmel.com/atmel/acrobat/doc2502.pdf)

Konfigurasi pin ATmega8535 adalah sebagai berikut:

- (1) **Vcc** : Tegangan
- (2) **GND** : Ground
- (3) **Port A (PA7 .. PA0)** : Port A digunakan sebagai *input analog* ke ADC, kemudian *port* ini juga berfungsi sebagai sebuah *port I/O* bidireksional 8-bit, jika ADC tidak digunakan port pin berfungsi sebagai *resistor pull-up internal* (dipilih untuk setiap bit). *Buffer output port A* memiliki karakter *drive* yang simetrik dengan *high sink* (serap) dan *source* (catu) *capability*. Ketika pin PA0 ke PA7 digunakan sebagai *input pulled-low eksternal*, pin-pin tersebut akan dicatu jika *pull-up resistor* diaktifkan

- (4) **Port B (PB7 .. PB0)** : Port I/O 8-bit dengan *pull-up internal* tiap pin. *Buffer port B* mempunyai kapasitas menyerap (*sink*) dan mencatu (*source*). Khusus PB6 dapat digunakan sebagai input kristal (*inverting oscillator amplifier*) dan *input* ke rangkaian *clock internal*, bergantung pada pengaturan *fuse bit* (ada dalam *software programmer/downloader*) yang digunakan untuk memilih sumber *clock*. Khusus PB7 dapat digunakan *output* kristal (*output inverting oscillator amplifier*) bergantung pada pengaturan *fuse bit* yang digunakan untuk memilih sumber *clock*. Jika sumber *clock* yang dipilih dari *oscillator internal*, PB7 dan PB6 dapat digunakan sebagai I/O atau jika menggunakan *asynchronous timer/counter2* maka PB6 dan PB7 (TOSC2 dan TOSC1) digunakan untuk saluran *input counter*.
- (5) **Port C (PC5 .. PC0)** : Port I/O 7-bit ([PC6], PC5...PC0) dengan *resistor pullup internal* tiap pin. *Buffer port C* mempunyai kapasitas menyerap (*sink*) dan mencatu (*source*)
- (6) **RESET/PC6** : Jika fuse bit RSTDISBL di "programmed", PC6 digunakan sebagai pin I/O. Jika fuse bit RSTDISBL di "unprogrammed", PC6 digunakan sebagai pin RESET (active low)
- (7) **Port D (PD7 .. PD0)** : Port I/O 8-bit dengan *resistor pull-up internal* tiap pin. *Buffer port D* mempunyai kapasitas menyerap (*sink*) dan mencatu (*source*)

(8) **AVcc** adalah pin tegangan catu untuk *A/D converter*, PC3..PC0, dan ADC(7..6). AVcc harus dihubungkan ke Vcc, walaupun ADC tidak digunakan. Jika ADC digunakan, maka AVcc harus dihubungkan ke Vcc melalui "*low pass filter*". Catatan: PC5, PC4 digunakan catu tegangan Vcc digital

(9) **AREF** : untuk pin tegangan referensi analog untuk ADC

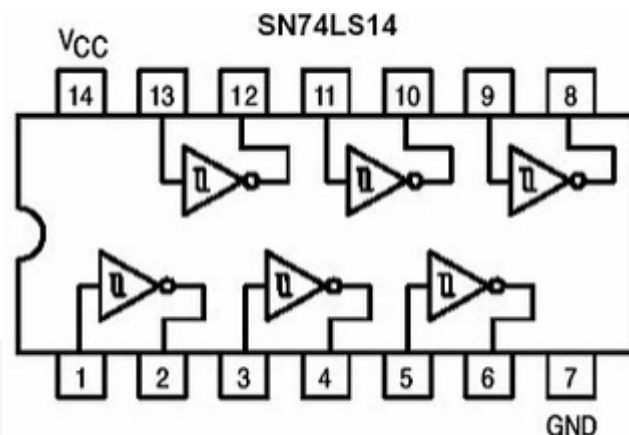
(10) **ADC7..6 (TQPF, QFN/MLF)**: Hanya ada pada kemasan TQPF dan QFN/MLF, ADC7..6 digunakan untuk pin *input* ADC.

2.6 IC 74LS14

IC 74LS14 atau sering disebut *Schmitt trigger* merupakan suatu rangkaian yang dapat mendeteksi tegangan *input* yang melintasi suatu peringkat tertentu. Selain itu *schmitt trigger* sangat berguna untuk pengkondisi sinyal segitiga ataupun bentuk gelombang lainnya, maka *output schmitt trigger* akan menghasilkan suatu keluaran gelombang segi empat atau pulsa digital.

IC 74LS14 merupakan salah satu jenis IC yang telah terpaket yang terdiri dari 6 buah *inverter* dengan *schmitt trigger*. Fungsi dari IC ini adalah sebagai pembalik dan pemantap atau untuk mendeteksi taraf dan membentuk kembali pulsa-pulsa yang buruk pada bagian tepinya (membentuk sinyal kotak).

Adapun konfigurasi pin IC 74LS14 ditunjukkan pada gambar 2.8.



Gambar 2.8 Skema IC 74LS14

2.7 LCD M1632

Modul LCD (*Liquid Crystal Display*) adalah salah satu alat yang digunakan sebagai tampilan. Sistem pengaturan LCD memiliki standar yang sama walaupun sangat banyak macamnya baik ditinjau dari perusahaan pembuat maupun dari ukurannya. M1632 merupakan modul *dot-matrix* tampilan kristal cair (LCD) dengan tampilan 2 x 16 baris dengan konsumsi daya rendah. Modul LCD ini telah dilengkapi dengan mikrokontroler yang didesain khusus untuk mengendalikan LCD, berfungsi sebagai pengatur (*system controller*) dan penghasil karakter (*character generator*). Fungsi pin yang terdapat pada LCD M1632 ditunjukkan seperti pada table 2.1.

Tabel 2.1 Konfigurasi kaki M1632

No	simbol	Level	Fungsi
1	Vss	-	0 volt
2	Vcc	-	5 + 10% volt
3	Vee	-	Penggerak LCD
4	RS	H/L	H = memasukan data L = memasukan Inst
5	R/W	H/L	H =baca, L = tulis
6	E		Enable Signal
7	DB0	H/L	Data BUS
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	
12	DB5	H/L	
13	DB6	H/L	
14	DB7	H/L	
15	V+BL	-	Kecerahan LCD
16	V-BL	-	

Ada beberapa hal yang perlu diketahui dalam pengoperasian LCD, yaitu:

2.7.1 Register

LCD tipe M1632 di dalamnya terdapat mikrokontroler yang telah menjadi satu kesatuan dengan LCD yang berguna untuk mengatur kerja pada LCD. Kontrol ini memiliki dua macam register yaitu *Instruction register (IR)* dan *Data Register (DR)*. Untuk memilih salah satu register digunakan *Register Select (RS)*, apabila RS berlogika '0' maka register yang dipilih adalah IR, dan bila RS berlogika '1' maka register yang dipilih adalah DR.

IR berfungsi mengirimkan kode perintah, informasi alamat DD RAM (*Display Data RAM*) dan RAM penghasil karakter (*Character Generator RAM*,

CG RAM). IR bisa ditulis dari mikrokontroler tetapi tidak bisa dibaca dan DR berfungsi menyimpan data sementara untuk dibaca dari LCD ke mikrokontroler maupun untuk ditulis dari mikrokontroler.

2.7.2 Busy Flag (BF)

Busy Flag digunakan untuk mengecek apakah modul LCD sudah siap menerima perintah selanjutnya. Mengecek BF dapat dilakukan pada jalur data DB7 dengan cara membuat RS berlogika '0' dan R/W berlogika '1', apabila DB7 berlogika '1' berarti modul LCD masih bekerja secara internal sehingga modul LCD tidak dapat menerima intruksi. Intruksi baru dapat diambil bila DB7 telah berlogika '0', oleh karena itu *busy flag* harus dicetak terlebih dahulu sebelum menulis perintah selanjutnya.

2.7.3 Address Counter

Address Counter (AC) adalah sebuah counter yang menunjukkan suatu alamat data DD RAM apabila dibaca maupun ditulis, apabila suatu informasi alamat dituliskan pada IR maka informasi alamat itu akan diteruskan dari IR ke AC. Bila suatu data dituliskan ke CG RAM atau DD RAM maka secara otomatis AC RAM akan bertambah maupun berkurang sesuai instruksi *Entry Mode Set* yang terlebih dahulu diatur. Informasi alamat yang akan ditunjuk ditulis pada jalur data DB0-DB7 dengan membuat RS berlogika '0' dan R/W berlogika '1'.

2.7.4 Display Data RAM (DD RAM)

Modul M1632 memiliki kemampuan untuk dapat menampilkan karakter sebanyak 32 karakter, yang semua datanya disimpan di dalam DD RAM. Kapasitas DD RAM pada modul M1632 adalah sebanyak 80 karakter. Baris

pertama yang ditampilkan LCD memiliki alamat DD RAM dari 00H-0FH sedangkan baris kedua memiliki alamat 40H-4FH. Alamat DD RAM ditunjukkan pada table 2.2.

Tabel 2.2 Alamat DD RAM yang ditampilkan

00h	01h	02h	03h	04h	05h	06h	07h	08h	09h	0Ah	0Bh	0Ch	0Dh	0Eh	0Fh
40h	41h	42h	43h	44h	45h	46h	47h	48h	49h	4Ah	4Bh	4Ch	4Dh	4Eh	4Fh

Bila terdapat perintah untuk menggeser karakter yang ditampilkan ke kiri, maka alamat DD RAM akan berubah. Perubahan alamat DD RAM ditunjukkan pada tabel 2.3.

Tabel 2.3 Alamat DD RAM yang ditampilkan setelah menerima geser ke kiri

01h	02h	03h	04h	05h	06h	07h	08h	09h	0Ah	0Bh	0Ch	0Dh	0Eh	0Fh	10h
41h	42h	43h	44h	45h	46h	47h	48h	49h	4Ah	4Bh	4Ch	4Dh	4Eh	4Fh	50h

Dari tabel 2.3 dapat dilihat bahwa karakter yang ditampilkan LCD bergeser ke kiri. LCD menampilkan karakter yang memiliki alamat 01H-10H pada baris pertama dan 40H-50H pada baris kedua pada DD RAM.

BAB 3

METODE PENELITIAN

3.1 Metode Penelitian

Metode yang digunakan pada penelitian ini adalah merancang suatu sistem pengukur kecepatan angin menggunakan mikrokontroler AVR Atmega8535. Hasil perancangan selanjutnya diuji untuk mengetahui besar kecepatan angin pada suatu daerah. Perancangan telah terpenuhi apabila pengukuran kecepatan angin pada alat uji sesuai dengan pengukuran kecepatan pada alat ukur standar.

3.2 Perencanaan Rangkaian

Rangkaian yang digunakan terdiri dari Rangkaian Catu daya 5V, Rangkaian Sensor arah angin, Rangkaian sensor kecepatan angin, piranti Mikrokontroler AVR Atmega8535 dan modul LCD (*Liquid crystal Display*) untuk menampilkan hasil pengukuran kecepatan.

3.3 Perancangan Sistem

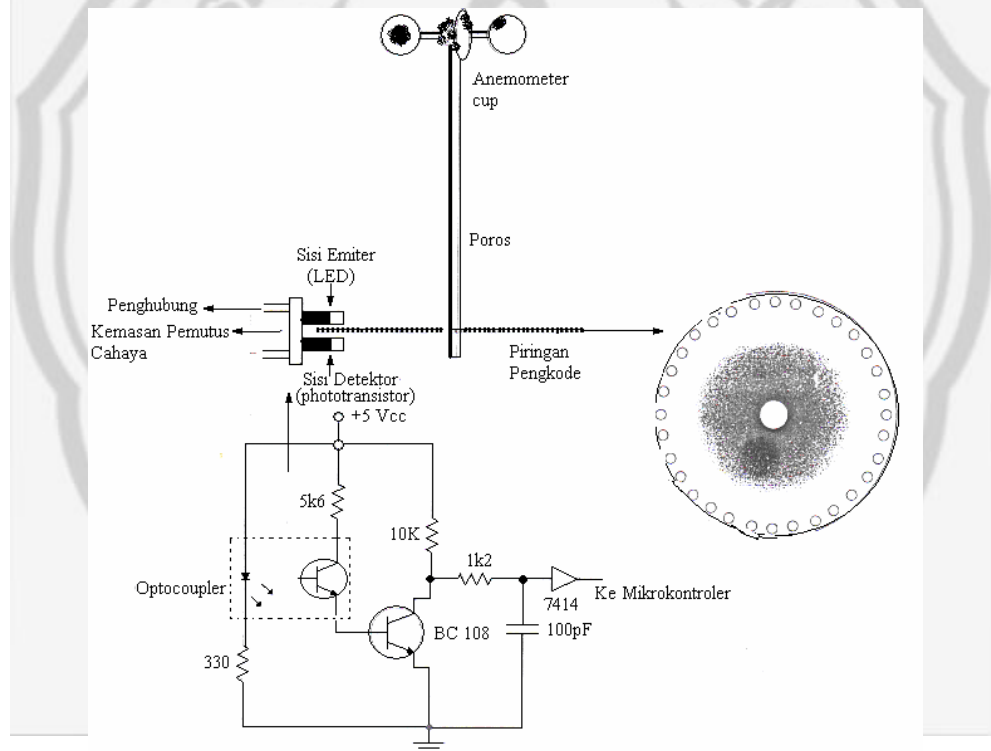
3.3.1 Perancangan Sistem Pengukuran Kecepatan Angin

Pengukur kecepatan angin terdiri dari sensor berupa balingbaling mangkok yang dikopel dengan piringan pengkode, *optocoupler*, dan IC pemacu *schmitt* 74LS14.

Baling-baling mangkok dikopel dengan piringan yang digunakan untuk mengindra kecepatan putar baling-baling mangkok tersebut. Titik pusat piringan

dan titik pusat baling-baling mangkok dihubungkan dengan sebuah poros, sehingga piringan berputar sesuai dengan kecepatan baling-baling mangkok.

Phototransistor dan LED inframerah diletakkan secara terpisah dan berhadapan, diantara LED dan *phototransistor*, diletakkan sebuah piringan pengkode yang terdiri atas bagian yang tembus cahaya dan bagian yang tidak tembus cahaya. Bagian yang tembus cahaya mengindikasikan keadaan pada logika 1, sedangkan bagian yang tidak tembus cahaya menunjukkan keadaan pada logika 0, seperti tampak pada gambar 3.1.



Gambar 3.1 Sistem Pengukuran Kecepatan Angin

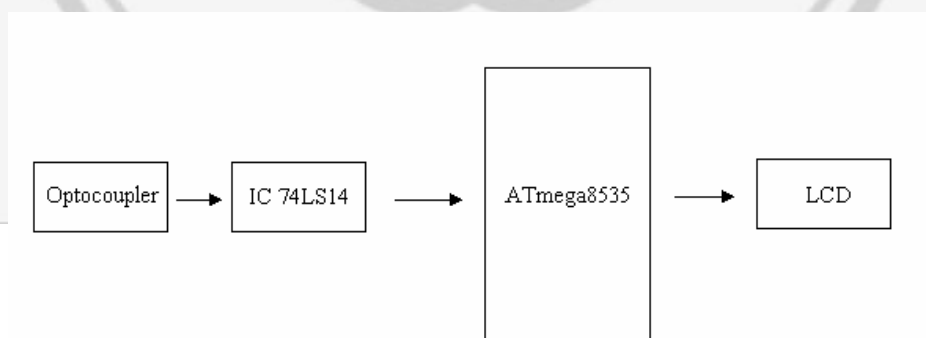
Untuk menghasilkan taraf tegangan yang sesuai maka keluaran phototransistor dikuatkan dengan menggunakan penguat emitter bersama. Untuk memperbaiki bentuk sinyal keluaran penguat dihubungkan pemacu *Schmitt*. Pemacu *Schmitt* diperlukan karena tegangan keluaran dari *optocoupler* pada

pengindera kecepatan angin berosilasi. Osilasi tegangan ini terjadi pada saat transisi dari logika 0 ke logika 1 ataupun sebaliknya, karena itulah diperlukan pemacu *Schmitt* untuk mengatasi osilasi tegangan tersebut.

IC yang digunakan sebagai pemacu *Schmitt* adalah IC 74LS14 yang mempunyai karakteristik waktu transisi *low to high* dan *high to low* yang sangat cepat, yaitu 15 ns. Waktu transisi yang cepat sebesar 15 ns cukup untuk menanggapi laju perubahan tegangan dari *optocoupler* pada pengindera kecepatan angin.

3.3.2 Perancangan Perangkat Keras(Hardware)

Prinsip kerja anemometer digital, yaitu sensor mengubah besaran kecepatan angin menjadi sinyal digital. Sinyal ini diolah oleh suatu rangkaian elektronik IC digital yaitu 74LS14 (sebagai pemacu *Schmitt*), data diolah oleh piranti Mikrokontroler, yang kemudian hasil pengolahan data ditampilkan pada LCD (*Liquid Crystal Display*). Blok diagram rancangan perangkat keras pengukur kecepatan angin diperlihatkan pada gambar 3.2 seperti di bawah ini.

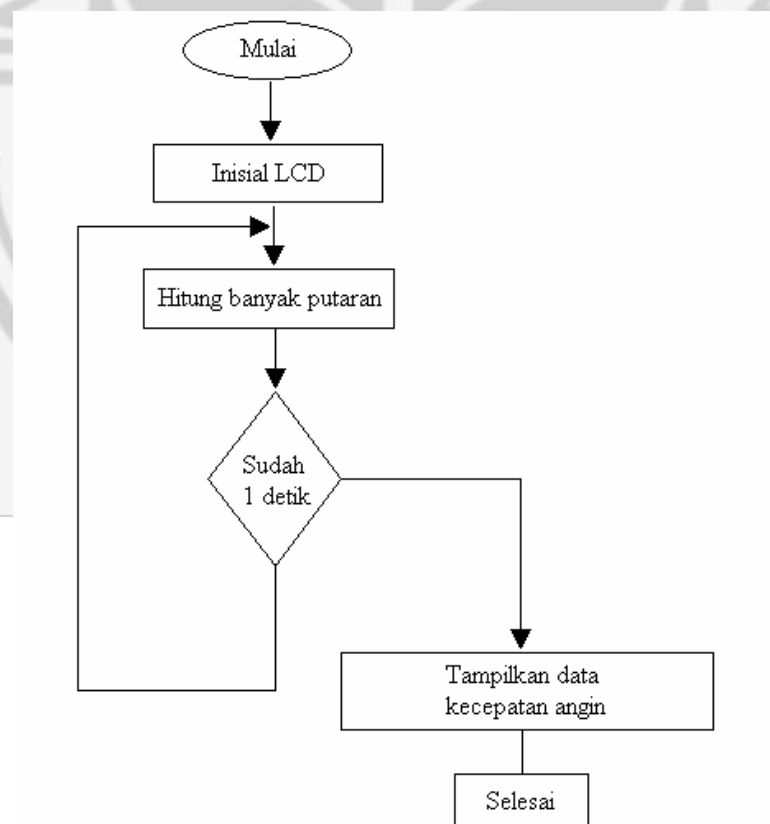


Gambar 3.2 Rancangan Pengukur Kecepatan Angin

3.3.3 Perancangan Perangkat Lunak

Perancangan sistem pengukuran kecepatan angin menggunakan bahasa pemrograman yaitu bahasa *Assembly*. Penggunaan bahasa *Assembly* sebagai pengatur lalu-lintas data ke Mikrokontroller, data yang diterima kemudian diolah untuk ditampilkan pada modul LCD (Liquid Crystal Display) sebagai hasil pengukuran kecepatan angin.

Dalam perencanaan sistem menggunakan *port* B0 hingga B4 sebagai *input*, sedangkan *port* A.0 hingga A.7 digunakan sebagai *output* untuk LCD, bahasa pemrograman sebelumnya telah disimpan ke dalam memori Mikrokontroller melalui *Downloader*. Bahasa pemrograman pengukuran kecepatan angin ditunjukkan *flowchart* pada gambar 3.3.

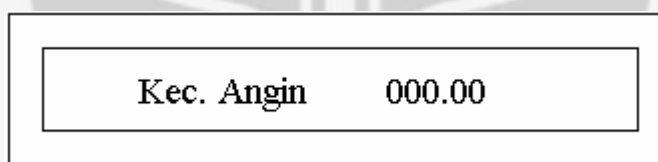


Gambar 3.3 Flowchart Bahasa Pemrograman

Pada gambar 3.3 di atas menunjukkan diagram alir dari bagian program yang dimulai dengan inisialisasi LCD kemudian dilanjutkan dengan mendeteksi sensor-sensor yang menunjukkan kecepatan angin.

Angin yang menerpa anemometer cup menggerakkan baling-baling dan memutar piringan pengkode data (*encoder*) 16 *hole*. Sensor mendeteksi jumlah *hole* yang melewati sensor pada piringan, setiap putaran tiap detik.

Data kecepatan yang ditampilkan pada LCD merupakan data perubahan kecepatan angin tiap detik, sehingga perubahan kecepatan angin dapat diamati setiap detik dengan tepat. Data yang terekam pada Mikrokontroler diteruskan menuju modul LCD, yang sebelumnya telah diolah didalam Mikrokontroler. Hasil pengukuran kecepatan angin ditampilkan pada LCD (Liquid Crystal Display) ditunjukkan pada gambar 3.4 seperti dibawah ini.



Gambar 3.4 Tampilan LCD

3.3.4 Kalibrasi

Kalibrasi dilakukan bila dalam pengukuran kecepatan angin dengan menggunakan alat uji memiliki hasil yang berbeda dengan hasil pengukuran alat pengukur kecepatan angin acuan.

BAB 4

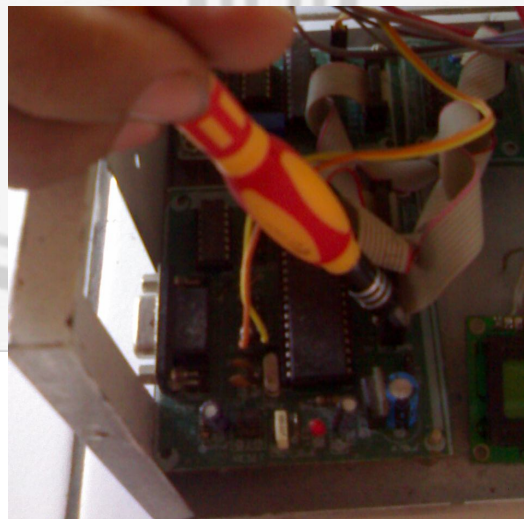
HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Penelitian

Sistem pengukuran kecepatan angin dengan sensor *optocoupler* berbasis Mikrokontroler AVR ATmega8535 merupakan piranti alat ukur *portable*, murah dan handal yang diharapkan mampu diaplikasikan pada komputer untuk pengukuran secara terus-menerus maupun penyimpanan data historis dengan memanfaatkan *internal timer Personal Computer* dengan bantuan program bahasa komputer.

4.1.1 Pengujian Sistem Pengukuran Kecepatan Angin

4.1.1.1 Pengujian Tegangan Keluaran Sensor Optocoupler



Gambar 4.1 Pengujian tegangan sensor optocoupler

Pengujian tegangan keluaran sensor kecepatan angin terhadap perubahan keadaan sensor oleh piringan adalah untuk mengetahui nilai tegangan keluaran

sensor kecepatan angin dan tegangan keluaran *schmitt trigger* ketika terhalang atau tidak.

Tabel 4.1 Tegangan sensor terhadap perubahan halangan

Keadaan Sensor	Tegangan Output	
	Sensor	Schmitt Trigger
Terhalang	2.0 - 5	0.15
Tak Terhalang	0.2 - 1	4.56

Dari tabel 4.1 diatas dapat diketahui bahwa tegangan output dari sensor untuk keadaan sensor terhalang antara 2,0-5V, sedangkan pada keadaan sensor tidak terhalang sebesar 0,2-1V, hal ini menunjukkan bahwa tegangan output dari sensor tidak stabil.

Tegangan output *schmitt trigger* untuk masukan sebesar 2,0-5V akan menghasilkan nilai 0,15V sedangkan untuk masukan 0,2-1V akan menghasilkan tegangan sebesar 4,56V. Ini menunjukkan bahwa tegangan output *schmitt trigger* relatif stabil karena tegangan output ini akan dibaca komputer logika 0 untuk tegangan 0,15V atau logika 1 untuk tegangan 4,56V. Data logika yang dihasilkan *schmitt trigger* merupakan kebalikan dari data logika sensor, namun ini dapat diatasi dengan memberi perintah *Not* pada perangkat lunak, sehingga untuk sensor terhalang dibaca logika 1 dan sensor tak terhalang dibaca logika 0. Output *schmitt trigger* tidak pada tegangan terlarang sebesar 1-2,5V karena tegangan terlarang akan dibaca komputer sebagai logika 0 atau dapat juga sebagai logika 1.

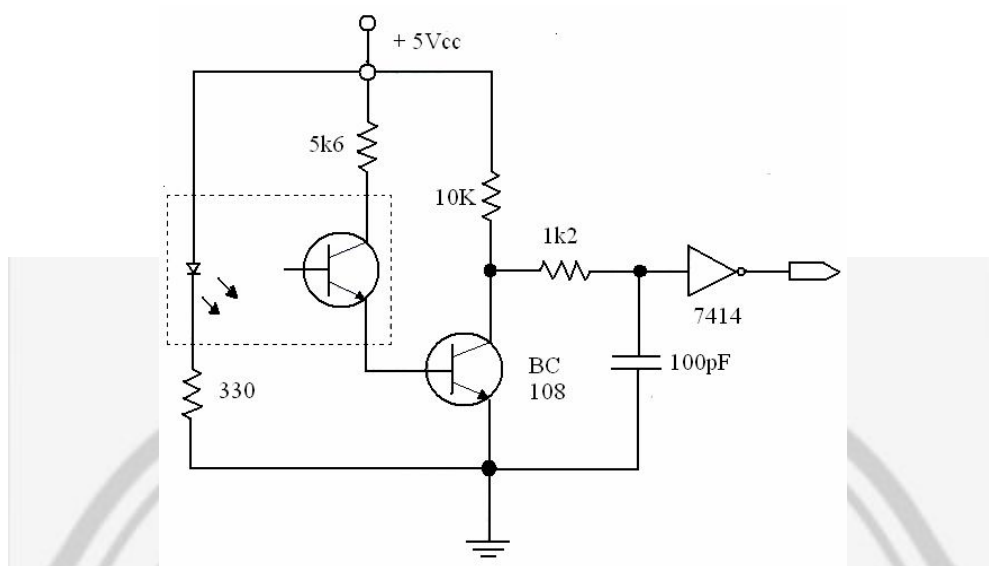
4.1.1.2 Pengujian Alat Pengukur Kecepatan Angin

Pengujian alat pengukur kecepatan angin dilakukan untuk mengetahui kemampuan alat dalam melaksanakan fungsinya. Pengujian dilakukan dengan membandingkan alat pengukur kecepatan angin hasil penelitian dengan alat pengukur kecepatan angin acuan. Pengukuran dilakukan dengan mengukur kecepatan angin secara langsung di lingkungan Unnes. Hasil pengukuran kecepatan angin ditunjukkan pada tabel 4.2

Tabel 4.2 Hasil Pengukuran Kecepatan Angin

Tanggal	Waktu	Hasil Pengukuran	
		Alat Ukur (m/s)	Alat Uji
1 Agustus 2011	09.00	4	1,48
	09.10	6	2,22
	09.20	4	1,48
	09.30	7	2,59
	09.40	8	2,96
	09.50	5	1,85
	10.00	4	1,48
	10.10	7	2,59
	10.20	6	2,22
	10.30	7	2,59
	10.40	6	2,22
	10.50	7	2,59
	11.00	8	2,96
	11.10	4	1,48
	11.20	5	1,85
11.30	7	2,59	

4.1.1.3 Analisis Rangkaian Sensor Optocoupler



Gambar 4.2 Rangkaian Sensor Optocoupler

Pada rangkaian yang ditunjukkan gambar 4.3 Transistor BC 108 berfungsi sebagai saklar elektronik. Untuk level saturasi dapat memastikan kondisi yang memenuhi syarat:

$$I_B = \frac{I_{Csat}}{\beta_{dc}}$$

Menurut Sutrisno, (1986:123) besarnya nilai I_B dan I_C diketahui dengan perhitungan sebagai berikut:

$$I_B = \frac{V_{CC} - 0,7V}{R_B}$$

$$I_{Csat} = \frac{V_{CC}}{R_C}$$

4.2 Pembahasan

Pengukuran dilakukan dengan mengukur kecepatan angin secara langsung di lingkungan Unnes. Pengujian pengukuran kecepatan angin dilakukan

antara pukul 09.00 hingga 11.30 WIB dengan interval setiap pencatatan 10 menit sekali. Pada tabel 4.2 pengukuran kecepatan angin dengan alat ukur menghasilkan data pengukuran 4 hingga 8 m/s. Sedangkan hasil pengukuran kecepatan dengan alat uji menghasilkan pencatatan 1 hingga 2.96.

Pengukuran yang telah dilakukan menunjukkan adanya perbedaan pengukuran. Perbedaan pengukuran ini dapat diketahui dengan perhitungan sebagai berikut.

$$\varepsilon = M - T$$

$$\varepsilon(\%) = \frac{M - T}{T} \times 100\%$$

Keterangan:

ε = Tingkat Perbedaan

$\varepsilon(\%)$ = Rasio Perbedaan (%)

M = Harga pengukuran

T = Harga sebenarnya

Dari perhitungan data penelitian pada tabel 4.2 dihasilkan rasio perbedaan sebesar 63%. Hal ini dapat diatasi dengan cara kalibrasi data pengukuran.

Kalibrasi dilakukan dengan menambahkan faktor kalibrasi pada hasil pengukuran alat uji yang hasilnya berbeda dengan hasil pengukuran alat ukur acuan. Jika dilihat dari tabel hasil pengukuran pada tabel 4.2 faktor kalibrasi dapat dianalisis sebagai berikut:

$$y = ax$$

dimana v merupakan faktor kalibrasi, a merupakan faktor koreksi dan x merupakan data hasil pengukuran alat uji

dengan;

$$a = \frac{v}{x}$$

$$\Rightarrow x = (100\% - 63\%) \times v$$

$$\Rightarrow x = 37\% \times v$$

$$a = \frac{v}{37\% \times v}$$

$$a = \frac{1}{37\%}$$

$$a = \frac{100}{37}$$

maka faktor kalibrasi menjadi:

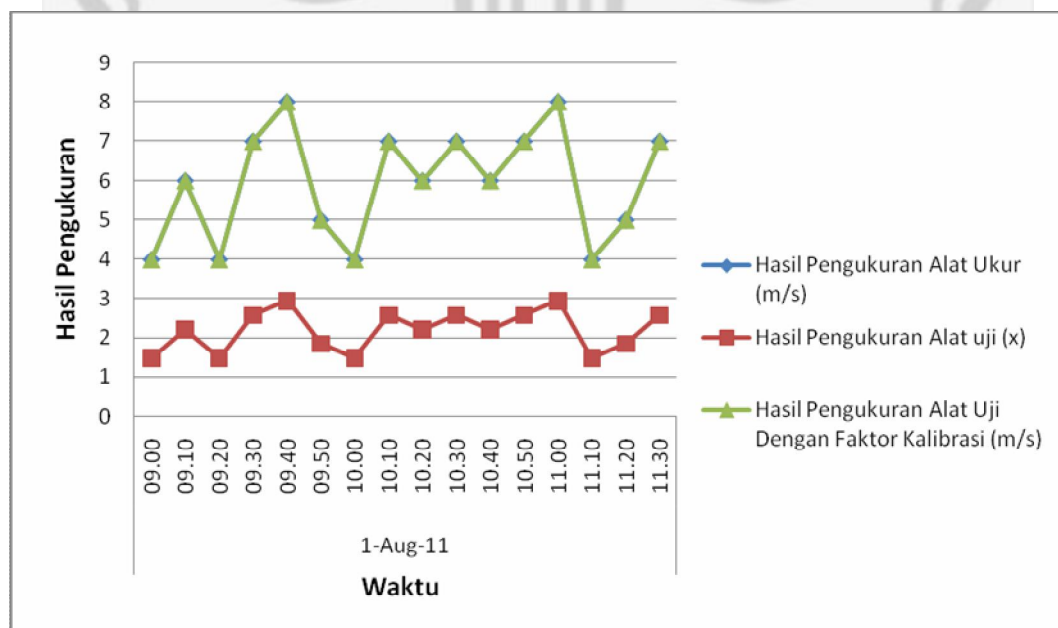
$$v = \frac{100}{37} x$$

Dari persamaan di atas dapat dijelaskan bahwa faktor koreksi (a) merupakan faktor pengali data hasil pengukuran alat uji (x).

Alat uji dalam penelitian ini dapat digunakan sebagai alat ukur yang baik jika setiap pembacaannya ditambahkan faktor kalibrasi. Hasil pengukuran alat uji setelah ditambahkan faktor kalibrasi dapat ditunjukkan dalam tabel 4.3 dan grafik pada halaman selanjutnya.

Tabel 4.3 Hasil Pengukuran Kecepatan Angin dengan Faktor Kalibrasi

Tanggal	Waktu	Hasil Pengukuran		
		Alat Ukur (m/s)	Alat Uji (x)	Faktor Kalibrasi $v = \frac{100}{37}x$ (m/s)
1 Agustus 2011	09.00	4	1,48	4
	09.10	6	2,22	6
	09.20	4	1,48	4
	09.30	7	2,59	7
	09.40	8	2,96	8
	09.50	5	1,85	5
	10.00	4	1,48	4
	10.10	7	2,59	7
	10.20	6	2,22	6
	10.30	7	2,59	7
	10.40	6	2,22	6
	10.50	7	2,59	7
	11.00	8	2,96	8
	11.10	4	1,48	4
	11.20	5	1,85	5
	11.30	7	2,59	7



Gambar 4.3 Grafik hasil pengukuran terhadap waktu dengan faktor kalibrasi

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan perancangan, pengujian dan analisis terhadap alat pengukur kecepatan angin dengan sensor *optocoupler* berbasis Mikrokontroler AVR Atmega8535 dapat ditarik kesimpulan sebagai berikut:

Alat pengukuran kecepatan angin dengan sensor *optocoupler* berbasis Mikrokontroler AVR ATmega8535 telah berhasil dibuat dan dapat dijadikan alat ukur yang tepat jika setiap pembacaan datanya ditambahkan dengan faktor

kalibrasi $v = \frac{100}{37} x$.

5.2 Saran

Untuk pengembangan selanjutnya, penulis berharap supaya proses kalibrasi sistem pada *software* alat uji dapat berjalan dengan sempurna, sehingga tepat dalam hasil pengukurannya sesuai dengan alat ukur acuan.

DAFTAR PUSTAKA

Wardana, L. 2006. *Belajar Sendiri AVR Seri ATmega8535; Simulasi, Hardware, dan Aplikasi*. Yogyakarta: Andi Offset

Fraden, J. 1996. *Modern Sensors*. California: Springer

Giancoli, D.C. 1997. *Fisika Edisi 4*. Jakarta: Erlangga

Malik, I, A. 1997. *Bereksperimen dengan Mikrokontroler 8031*. Jakarta: Elex Media Komputindo

Petruzella, F.D. 1996. *Elektronika Industri*. Yogyakarta: Andi Offset

Putra, A, E. 2002. *Belajar Mikrokontroler AT89C51/52/55 Teori dan Aplikasi*. Yogyakarta: Gava Media

Sutrisno. 1986. *Elektronika Teori dan Terapannya Jilid 2*. Bandung: ITB

Sutrisno. 1987. *Elektronika Lanjut Teori dan Terapannya*. Bandung: ITB

Tippler, P.A. 1998. *Fisika jilid 1*. Jakarta: Erlangga

Tooley, M. 2003. *Rangkaian Elektronika Prinsip dan Aplikasi*. Jakarta: Erlangga

...2010. *Mikrokontroler AVR Atmega8535*. Tersedia di: <http://www.atmel.com/atmel/acrobat/doc2502.pdf>

...2011. *Sensor Optocoupler*. Tersedia di: <http://www.soselectronic.com/>

LAMPIRAN 1

Data Pengujian Sensor

Keadaan Sensor	Tegangan Output	
	Sensor	Schmitt Trigger
Terhalang	2.0 - 5	0.15
Tak Terhalang	0.2 - 1	4.56



LAMPIRAN 2

Data Hasil Pengukuran Kecepatan Angin

Tanggal	Waktu	Hasil Pengukuran	
		Alat Ukur (m/s)	Alat Uji
1 Agustus 2011	09.00	4	1,48
	09.10	6	2,22
	09.20	4	1,48
	09.30	7	2,59
	09.40	8	2,96
	09.50	5	1,85
	10.00	4	1,48
	10.10	7	2,59
	10.20	6	2,22
	10.30	7	2,59
	10.40	6	2,22
	10.50	7	2,59
	11.00	8	2,96
	11.10	4	1,48
	11.20	5	1,85
	11.30	7	2,59

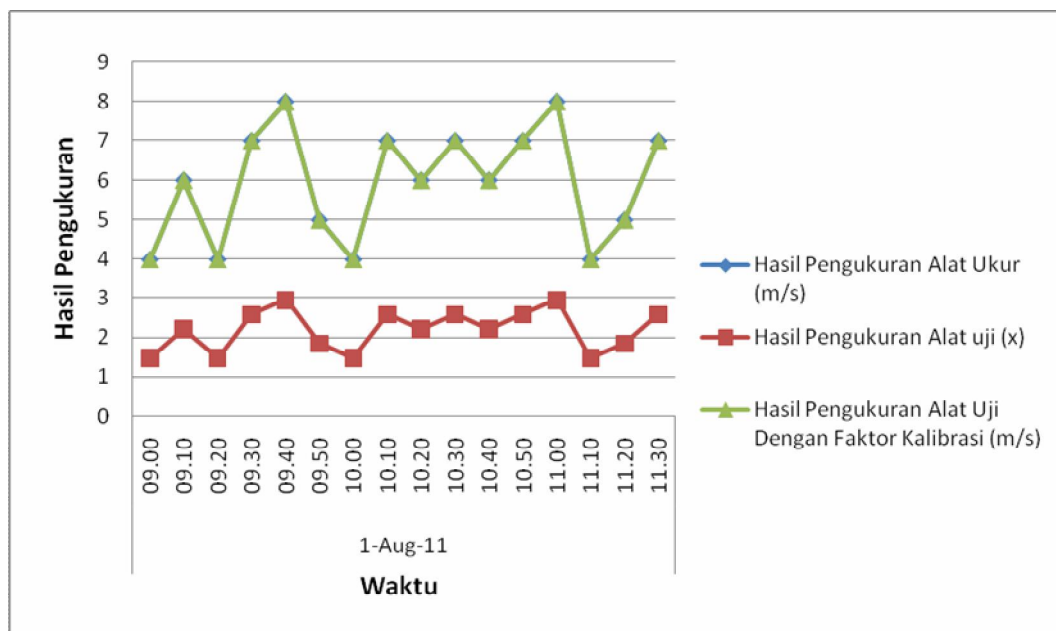
LAMPIRAN 3

Data Hasil Pengukuran Kecepatan Angin dengan Faktor Kalibrasi

Tanggal	Waktu	Hasil Pengukuran		
		Alat Ukur (m/s)	Alat Uji (x)	Faktor Koreksi $v = \frac{100}{37} x$ (m/s)
1 Agustus 2011	09.00	4	1,48	4
	09.10	6	2,22	6
	09.20	4	1,48	4
	09.30	7	2,59	7
	09.40	8	2,96	8
	09.50	5	1,85	5
	10.00	4	1,48	4
	10.10	7	2,59	7
	10.20	6	2,22	6
	10.30	7	2,59	7
	10.40	6	2,22	6
	10.50	7	2,59	7
	11.00	8	2,96	8
	11.10	4	1,48	4
	11.20	5	1,85	5
	11.30	7	2,59	7

LAMPIRAN 4

Grafik hasil pengukuran terhadap waktu dengan faktor kalibrasi



LAMPIRAN 5

Hardware mikrokontroler AVR ATmega8535



LAMPIRAN 6

Sistem lengkap alat pengukur kecepatan angin



LAMPIRAN 7

Program mikrokontroler AVR ATmega8535

```
*****
```

```
;Sistem kontrol
```

```
*****
```

```
#include <mega8535.h>
```

```
.equ sh = $56
```

```
.equ sl = $49
```

```
.equ sht = $57 ;sht = sh + 1
```

```
*****
```

```
;Register
```

```
*****
```

```
#####register sementara#####
```

```
.def temp=r16
```

```
.def temp1=r18
```

```
.def c1=r22
```

```
.def c2=r4
```

```
.def d1=r5
```

```
.def d2=r6
```

```
.def status=r7 ;menyimpan nilai status register
```

```
#####Register waktu#####
```

```
.def count_1=r17
```

```

.def time_1=r1

.def time_0=r2

;#####Registers pengukur kecepatan#####

.def sdigit_01=r19

.def sdigit_1=r20 ;menyimpan nilai biner kecepatan

;#####Register tampilan LCD#####

.def lcd_cmd=r21

.def lcd_dat=r23

.def count=r3

;#####Register delays#####

.def low_del=r24

.def hi_del=r25

;#####Register EEPROM#####

.def eep_reg=r11

.def eep_addr=r12

;r14r15

;*****

;segmen kode

;*****

.cseg

.org 0

    rjmp  RESET

.org INT0addr

```

```
rjmp IntV0
```

```
reti
```

```
reti
```

```
reti
```

```
reti
```

```
.org OVf0addr
```

```
rjmp TimerV0
```

```
reti
```

```
reti
```

```
reti
```

```
reti
```

```
.org $000b
```

```
RESET:
```

```
#####Inisialisasi Stack Pointer#####
```

```
ldi temp,RAMEND
```

```
out SPL, temp
```

```
#####Inisialisasi port I/O#####
```

```
ldi temp, 0b11111111 ;mengatur PORT B untuk keluaran
```

```
out DDRB, temp
```

```
ldi temp,0
```

```
out DDRD,temp
```

```
ldi temp,$ff
```

```
out PORTD,temp ;mengatur PORT D untuk masukan
```

```
#####Inisialisasi INT 0 interrupt#####
```

```
ldi temp,$40 ;mengaktifkan INT 0 interrupt
```

```
out GIMSK,temp
```

```
ldi temp,$03
```

```
out MCUCR,temp
```

```
#####Inisialisasi waktu#####
```

```
ldi temp,$02
```

```
out TIMSK,temp ;mengaktifkan timer overflow interrupt
```

```
ldi temp,$05
```

```
out TCCR0,temp
```

```
ldi temp,$00 ;Inisialisasi Register
```

```
out TCNT0,temp
```

```
clr time_0
```

```
clr time_1
```

```
#####Inisialisasi register kecepatan#####
```

```
clr sdigit_1
```

```
clr sdigit_01
```

```
#####Inisialisasi tampilan LCD#####
```

```
rcall init_lcd
```

```
rcall init_lcd
```

```
rcall hi_delay
```

```
sei
```

```
*****
```



```
;Program utama
```

```
*****
```

```
loop:
```

```
    rcall print_lcd      ;pengulangan tampilan LCD
```

```
    rjmp loop
```

```
*****
```

```
;INT 0 ISR
```

```
*****
```

```
IntV0:
```

```
    in status,SREG
```

```
    in time_0,TCNT0
```

```
    mov time_1,count_1
```

```
    clr count_1
```

```
    out TCNT0,count_1
```

```
    inc dist_count
```

```
    rcall kec           ;pengulangan penghitungan kecepatan
```

```
    out SREG,status
```

```
    reti
```

```
*****
```

```
;Waktu overflow ISR
```

```
*****
```

```
TimerV0:
```

```
    in status,SREG
```

```

inc count_1

cpi count_1,sht

brsh timeff

out SREG,status

reti

```

```
timeff:
```

```

clr count_1
rcall time00
ret

```

```

;*****
;

```

```
;Menghitung kecepatan
```

```

;*****
;

```

```
kec:
```

```

push d1
push d2
push c1
push c2
ldi temp1,$00

```

```
cp time_1,temp1
```

```
breq time0
```

```
kecq:
```

```

clr sdigit_1
ldi temp1,sh

```

```
mov c1,temp1
```

```
ldi temp1,s1
```

```
mov c2,temp1
```

```
rjmp htgkec
```

```
time0:
```

```
cp time_0,temp1
```

```
breq time00
```

```
rjmp kecq
```

```
time00:
```

```
clr sdigit_1
```

```
clr sdigit_01
```

```
pop d1
```

```
pop d2
```

```
pop c1
```

```
pop c2
```

```
ret
```

```
htgkec:
```

```
cp c1,time_1
```

```
brsh htg_kec
```

```
rjmp kec0
```

```
htg_kec:
```

```
cp c2,time_0
```

```
brsh htg_kec1
```

```
cp c1,time_1  
breq kec0  
rjmp htg_kec2
```

```
htg_kec1:
```

```
inc sdigit_1
```

```
sub c1,time_1
```

```
sub c2,time_0
```

```
rjmp htgkec
```

```
htg_kec2:
```

```
inc sdigit_1
```

```
sub c1,time_1
```

```
sub c2,time_0
```

```
dec c1
```

```
rjmp htgkec
```

```
kec0:
```

```
rjmp khtg
```

```
khtg:
```

```
mov d1,c1
```

```
mov d2,c2
```

```
ldi temp1,10
```

```
rjmp khtg1
```

```
khtg1:
```

```
cpi temp1,1
```

brne khtg2

rjmp khtg3

khtg2:

dec temp1

add d2,c2

adc d1,c1

rjmp khtg1

khtg3:

cp d1,time_1

brsh khtg4

rjmp kec1

khtg4:

cp d2,time_0

brsh khtg5

cp d1,time_1

breq kec1

rjmp khtg6

khtg5:

inc sdigit_01

sub d1,time_1

sub d2,time_0

rjmp khtg3

khtg6:

```
inc sdigit_01
sub d1,time_1
sub d2,time_0
dec d1
```

```
rjmp khtg3
```

kec1:

```
pop d1
pop d2
pop c1
pop c2
ret
```

```
;*****
```

```
;Pengulangan baca data dari eeprom
```

```
;eep_addr menyimpan alamat eeprom
```

```
;eep_reg menyimpan data
```

```
;*****
```

eep_nr:

```
sbic EECR,EERE
```

```
rjmp eep_nr
```

baca:

```
out EEAR, eep_addr
```

```
sbi EECR, EERE
```

```

nop

nop

in eep_reg, EEDR

ret

```

```

;*****

```

```

;Pengulangan tulis data ke eeprom

```

```

;eep_addr menyimpan alamat eeprom

```

```

;eep_reg menyimpan data

```

```

;*****

```

```

eeprom_tulis:

```

```

    sbic EECR, EEWE

```

```

    rjmp eeprom_tulis

```

```

tulis:

```

```

    out EEAR,eep_addr

```

```

    out EEDR,eep_reg

```

```

    cli

```

```

    sbi EECR,EEMWE

```

```

    sbi EECR,EEWE

```

```

    sei

```

```

    ret

```

```

;*****

```

```

;Init_Lcd: Inisialisasi modul LCD 16 X 2 data 4 bit

```

```
*****
```

```
init_lcd:
```

```
    ldi lcd_cmd, 3
```

```
    rcall lcd_low_cmd
```

```
    rcall hi_delay
```

```
    ldi lcd_cmd, 3
```

```
    rcall lcd_low_cmd
```

```
    rcall low_delay
```

```
    ldi lcd_cmd, 3
```

```
    rcall lcd_low_cmd
```

```
    rcall low_delay
```

```
    ldi lcd_cmd, $28 ; set 4-bit interface
```

```
    rcall lcd_all_cmd
```

```
    ldi lcd_cmd, 8 ; set alamat DDRAM ke 00
```

```
    rcall lcd_all_cmd
```

```
    ldi lcd_cmd, $0c
```

```
    rcall lcd_all_cmd
```

```
    ldi lcd_cmd, 6
```

```
    rcall lcd_all_cmd ; mode setting
```



```
ret
```

```
,*****
```

```
;Print_Lcd: Menampilkan kecepatan pada modul LCD
```

```
,*****
```

```
print_lcd:
```

```
    ldi lcd_cmd, $80
```

```
    rcall lcd_all_cmd
```

```
    ldi temp, 16
```

```
    mov count, temp
```

```
    ldi ZH, high(msg1*2)
```

```
    ldi ZL, low(msg1*2)
```

```
more1:
```

```
    lpm
```

```
    mov lcd_dat, r0
```

```
    rcall lcd_all_dat
```

```
    adiw ZL, 1
```

```
    dec count
```

```
    ldi temp, 0
```

```
    cp count, temp
```

```
    brne more1
```

```
    ldi temp, 16
```

```
    mov count,temp
```

```

ldi lcd_cmd, $c0
rcall lcd_all_cmd

```

```
more2:
```

```
lpm
```

```
mov lcd_dat, r0
```

```
rcall lcd_all_dat
```

```
adiw ZL, 1
```

```
dec count
```

```
ldi temp, 0
```

```
cp count, temp
```

```
brne more2
```

```
#####Konversi Biner ke BCD#####
```

```
clr d2
```

```
speeddigit: ;Nilai ratusan
```

```
cpi sdigit_1,100
```

```
brsh sdigit_cal
```

```
rjmp print1
```

```
sdigit_cal:
```

```
inc d2
```

```
subi sdigit_1,100
```

```
rjmp speeddigit
```

```
print1:
```

```
ldi lcd_cmd,$86
```

```
rcall lcd_all_cmd
```

```
mov lcd_dat,d2
```

```
andi lcd_dat,$0f
```

```
ori lcd_dat,$30
```

```
rcall lcd_all_dat
```

```
rcall low_delay
```

```
mov c1,sdigit_1
```

```
rcall bcd
```

```
mov sdigit_1,c1
```

```
mov c2,d1
```

```
ldi lcd_cmd,$87 ;Nilai puluhan
```

```
rcall lcd_all_cmd
```

```
mov lcd_dat,c2
```

```
andi lcd_dat,$0f
```

```
ori lcd_dat,$30
```

```
rcall lcd_all_dat
```

```
rcall low_delay
```

```
ldi lcd_cmd,$88 ;Nilai satuan
```

```
rcall lcd_all_cmd
```

```
andi sdigit_1,$0f
```

```
mov lcd_dat,sdigit_1
```

```

ori lcd_dat,$30

rcall lcd_all_dat

rcall low_delay

```

```

ldi lcd_cmd,$8a          ;Nilai desimal

```

```

rcall lcd_all_cmd

```

```

andi sdigit_01,$0f

```

```

mov lcd_dat,sdigit_01

```

```

ori lcd_dat,$30

```

```

rcall lcd_all_dat

```

```

rcall low_delay

```

```

;*****

```

```

;Low_Delay

```

```

;*****

```

```

low_delay:

```

```

push low_del

```

```

push hi_del

```

```

ldi low_del, 20

```

```

ld_hi:

```

```

ldi hi_del, 10

```

```

loop_in:

```

```

dec hi_del

```

```

cpi hi_del, 0

brne loop_in

dec low_del

cpi low_del, 0

brne ld_hi

```

```

pop hi_del

pop low_del

ret

```

```

;*****

```

```

;Hi_Delay

```

```

;*****

```

```

hi_delay:

```

```

    ldi low_del, 25

```

```

more_call:

```

```

    rcall low_delay

```

```

    dec low_del

```

```

    cpi low_del, 0

```

```

    brne more_call

```

```

ret

```

```

;*****

```

```

;Lcd_Low_Cmd: Mengirim perintah 4 bit dari LCD

```

```

;*****

```

```

lcd_low_cmd:

```

```
mov temp, lcd_cmd
```

```
lsl temp
```

```
lsl temp
```

```
lsl temp
```

```
lsl temp
```

```
andi temp, $f0
```

```
out PORTB, temp
```

```
ori temp, $08
```

```
out PORTB, temp
```

```
andi temp, $f7
```

```
out PORTB, temp
```

```
ret
```

```
;*****
```

```
;Lcd_All_Cmd: Mengirim perintah 8 bit dari LCD
```

```
;*****
```

```
lcd_all_cmd:
```

```
push lcd_cmd
```

```
lsl lcd_cmd
```

```
lsl lcd_cmd
```

```
lsl lcd_cmd
```

```
lsl lcd_cmd
```

```
rcall lcd_low_cmd
```

```
pop lcd_cmd
```

```
andi lcd_cmd, $0f
```

```
rcall lcd_low_cmd
```

```
rcall low_delay
```

```
ret
```

```
*****
```

```
;Lcd_All_Dat: Mengirim data byte ke LCD
```

```
*****
```

```
lcd_all_dat:
```

```
push lcd_dat
```

```
lsr lcd_dat
```

```
lsr lcd_dat
```

```
lsr lcd_dat
```

```
lsr lcd_dat
```

```
andi lcd_dat, $0f
```

```
rcall lcdlowdat
```

```
pop lcd_dat
```

```
andi lcd_dat, $0f
```

```
rcall lcdlowdat
```

```
rcall low_delay
```

```
ret
```

```
*****
```

```
;Lcd_Low_Dat: Mengirim data bite 4 bit dari LCD
```

```
*****
```

lcdlowdat:

```
mov temp, lcd_dat
```

```
lsl temp
```

```
lsl temp
```

```
lsl temp
```

```
lsl temp
```

```
andi temp, $f0
```

```
ori temp, $04
```

```
out PORTB, temp
```

```
ori temp, $08
```

```
out PORTB, temp
```

```
andi temp, $f7
```

```
out PORTB, temp
```

```
ret
```

```
*****
```

```
;Message #1
```

```
*****
```

```
msg1: .db "Kec. Angin= ."
```

```
; 12345678123456781234567812345678
```