



**SISTEM MONITORING DATA KINCIR ANGIN BERBASIS
MIKROKONTROLER ATMEGA32 DI PANTAI BARU PANDANSIMO**

PROYEK AKHIR

**Diajukan kepada Fakultas Teknik Universitas Negeri Yogyakarta
Untuk Memenuhi Sebagian Persyaratan Guna
Memperoleh Gelar Ahli Madya Teknik**



**Oleh :
DINI NOTRIA M
NIM. 13507134020**

**PROGRAM STUDI TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2016**

HALAMAN PERSETUJUAN

PROYEK AKHIR

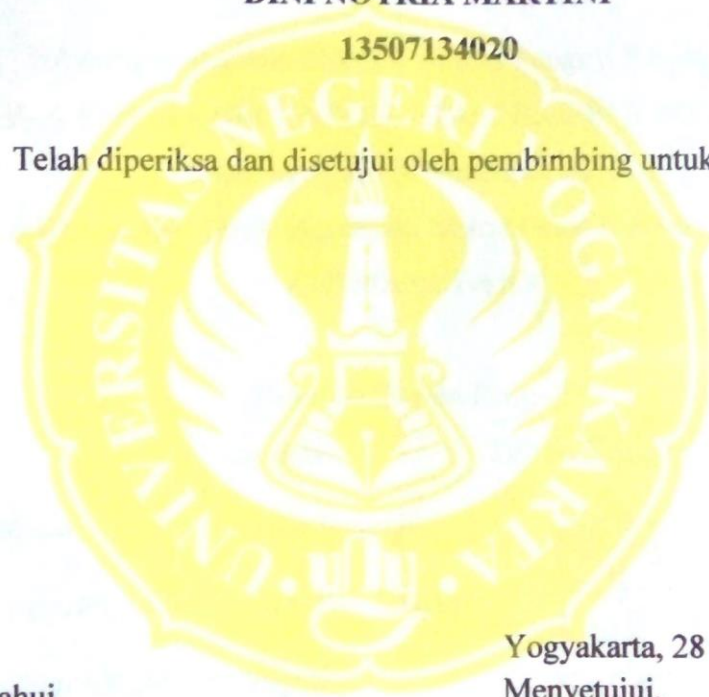
**SISTEM MONITORING DATA KINCIR ANGIN BERBASIS
MIKROKONTROLER ATMEGA32 DI PANTAI BARU PANDANSIMO**

Oleh

DINI NOTRIA MARTINI

13507134020

Telah diperiksa dan disetujui oleh pembimbing untuk diujikan.



Mengetahui,
Kepala Program Studi Teknik Elektronika

Yogyakarta, 28 Oktober 2016
Menyetujui,
Dosen Pembimbing Proyek Akhir

Dr. Sri Waluyanti, M.Pd
NIP. 19581218 198603 2 001

Dr. Sri Waluyanti, M.Pd
NIP. 19581218 198603 2 001

HALAMAN PENGESAHAN
PROYEK AKHIR
SISTEM MONITORING DATA KINCIR ANGIN BERBASIS
MIKROKONTROLER ATMEGA32 DI PANTAI BARU PANDANSIMO

Dipersiapkan dan Disusun Oleh:

DINI NOTRIA MARTINI

13507134020

Telah dipertahankan didepan Dewan Penguji Proyek Akhir
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

Pada tanggal 10 November 2016

Dan Dinyatakan Telah Memenuhi Syarat Guna Memperoleh Gelar
Ahli Madya Teknik

Susunan Dewan Penguji

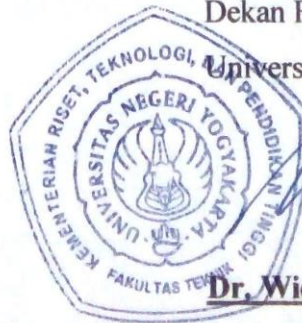
Nama	Jabatan	Tanda Tangan	Tanggal
Dr. Sri Waluyanti, M.Pd	Ketua Penguji		7 - Des - 2016
Nuryake Fajaryati, M.Pd	Sekretaris Penguji		6 - Des - 2016
Drs. Djoko Santoso, M.Pd	Penguji		7 - Des - 2016

Yogyakarta,⁸ Desember 2016

Mengetahui

Dekan Fakultas Teknik

Universitas Negeri Yogyakarta




Dr. Widarto, M.Pd.

NIP. 19631230 198812 1 001 4

HALAMAN PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama : Dini Notria Martini

NIM : 13507134020

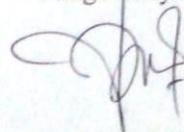
Program Studi : Teknik Elektronika D-III

Judul Proyek Akhir : Sistem monitoring data kincir angin berbasis mikrokontroler atmega32 di pantai baru pandansimo

Menyatakan bahwa Proyek Akhir ini adalah hasil pekerjaan saya sendiri, dan sepanjang pengetahuan saya, tidak berisi materi yang ditulis oleh orang lain sebagai persyaratan penyelesaian studi di Universitas Negeri Yogyakarta atau perguruan tinggi lain, kecuali bagian-bagian tertentu saya ambil sebagai acuan dengan mengikuti kaidah penulisan karya ilmiah yang benar. Jika ternyata terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab saya.

Yogyakarta, 27 Oktober 2016

Yang Menyatakan,



Dini Notria Martini

NIM. 13507134020

HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur kepada Allah SWT, kupersembahkan karya ini kepada:

1. Allah SWT yang telah memberikan rahmat dan hidayahnya, sehingga tidak ada kesulitan yang tidak bisa diselesaikan dalam pengerjaan proyek akhir ini
2. Kedua orang tua, Bapak (Parsono) dan Ibu (Tri Murti) yang tak pernah lelah untuk selalu memberi dukungan, perhatian, kasih sayang dan do'a untuk keberhasilan anak – anaknya
3. Dilla Notria saudara kembar yang telah memberikan dukungan penuh dan menemani penyusunan Proyek Akhir ini.
4. Teman-Teman Caramel (clara, kyky, malla, efril) yang selalu memberikan motivasi agar penyusunan laporan Proyek Akhir bisa segera selesai.
5. Teman – teman kelas B 2013 dan tim kincir angin terimakasih atas kebersamaannya
6. Semua pihak yang telah membantu dalam proses penyelesaian proyek akhir ini

MOTTO

- Karena sesungguhnya sesudah kesulitan ada kemudahan (QS.An insyirah)
- Be not sad, Surely Allah is with us (Qur'an 9:40)
- Kebanyakan orang merasa sukses itu adalah hasil jerih payah sendiri, tanpa campur tangan Tuhan. Mengingat Tuhan adalah sebagian ibadah vertikal dan menolong sesama sebagai ibadah horizontal -Bob Sadino-
- Aku berjanji akan menjadi seperti boneka daruma,yang selalu segera bangkit tegak kembali dengan muka yang tegar walaupun jatuh terguling-guling -Nobita-

SISTEM MONITORING DATA KINCIR ANGIN BERBASIS MIKROKONTROLER ATMEGA32 DI PANTAI BARU PANDANSIMO

Oleh : Dini Notria M

NIM : 13507134020

ABSTRAK

Sistem monitoring data kincir angin berbasis mikrokontroler atmega 32 di pantai baru pandansimo adalah sebuah sistem monitoring untuk memantau Tegangan, Arus, dan Daya. Tujuan dari proyek akhir ini adalah membuat, merancang, dan mengetahui unjuk kerja *data logger* sebagai sistem monitoring data pada Pembangkit Listrik Tenaga Angin.

Pembuatan *data logger* terdiri dari beberapa tahap, meliputi analisis kebutuhan alat yang dibutuhkan seperti perancangan catu daya, perancangan sensor tegangan, perancangan sensor arus, perancangan LCD, dan ATmega32. Perancangan alat ini meliputi blok diagram alat, perancangan *hardware* yang meliputi pembuatan PCB, box dan *software* pemograman dengan menggunakan CV-AVR. Pengujian unjuk kerja alat ini meliputi pengujian tanpa beban, dengan beban dan penyimpanan data.

Berdasarkan hasil pengujian yang telah dilakukan diperoleh hasil bahwa sistem monitoring data kincir angin berbasis mikrokontroler atmega 32 di pantai baru pandansimo adalah sebuah sistem monitoring untuk memantau Tegangan, Arus, dan Daya yang dihasilkan oleh kincir angin dengan memanfaatkan angin yang ada di pantai baru pandansimo. Unjuk kerja alat dengan pengujian Arus, Tegangan, Daya yang ditampilkan pada LCD yang dilakukan dengan pengukuran-pengukuran pada bagian-bagian tertentu sudah selesai dengan yang diharapkan.

Kata Kunci : Data Logger, Mikrokontroler ATmega32

KATA PENGANTAR



Puji syukur penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan penulisan laporan proyek akhir ini dengan judul “Sistem Monitoring Data Kincir Angin Berbasis Mikrokontroler Atmega 32 Di Pantai Baru Pandansimo” di Universitas Negeri Yogyakarta. Proyek akhir ini merupakan salah satu persyaratan memperoleh gelar Ahli Madya pada program Diploma-III Jurusan Pendidikan Teknik Elektronika, Program Studi D3-Teknik Elektronika, Universitas Negeri Yogyakarta. Proyek akhir ini dapat diselesaikan tak lepas dari berbagai pihak yang telah memberikan bantuan. Untuk itu pada kesempatan ini penulis menyampaikan ucapan terimakasih kepada:

Kedua orang tua dan keluarga yang selalu memberikan dorongan, semangat dan kasih sayangnya sehingga proyek akhir ini dapat diselesaikan.

1. Ibu Dr. Sri Waluyanti, M.Pd. selaku Ketua Progam Studi Pendidikan Teknik Elekronika dan selaku dosen pembimbing yang selalu memberikan masukan dan bimbingannya dalam mengerjakan proyek akhir.
2. Bapak Dr. Widarto, M.Pd. selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta
3. Bapak Dr. Fatchul Arifin, M.T selaku Ketua Jurusan Pendidikan Teknik Elekronika.
4. Para Dosen, Teknisi Lab, dan Staff Jurusan Pendidikan Teknik Elektronika yang telah memberikan bantuan selama ini.

5. Teman-Teman kelas B jurusan Teknik Elektronika angkatan 2013 yang telah memberikan bantuan dan semangatnya
6. Semua pihak yang telah memberikan bantuan moral maupun material sehingga proyek akhir ini dapat selesai dengan cepat.

Akhirnya dengan segala kerendahan hati, Penulis menyadari dalam pembuatan proyek akhir ini masih jauh dari sempurna. Saran dan kritik yang membangun sangat dibutuhkan guna menyempurnakan laporan proyek akhir ini. Semoga proyek akhir ini dapat memberikan manfaat bagi siapa saja yang membacanya

Yogyakarta,

Penulis

Dini Notria M
NIM. 13507134020

DAFTAR ISI

HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN KEASLIAN	iv
HALAMAN PERSEMBAHAN.....	v
MOTTO	vi
ABSTRAK	vii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiv
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Identifikasi Masalah.....	3
C. Batasan masalah.....	3
D. Rumusan masalah	3
E. Tujuan	4
F. Manfaat penulisan.....	4
G. Keaslian Gagasan.....	5
BAB II KAJIAN TEORI.....	6
A. Kincir Angin	6
B. Data Loger	9
C. ATmega 32.....	13

D. Sensor Arus	18
E. Sensor Tegangan (rangkaiian pembagi tegangan)	20
F. Modul RTC	20
G. Modul SD.....	21
H. LCD.....	23
BAB III PERANCANGAN DAN PEMBUATAN ALAT	25
A. Idenifikasi Kebutuhan	25
B. Analisis Kebutuhan	26
C. Tahap Perancangan dan Pembuatan.....	30
D. Diagram Alir / <i>Flowchart</i> Alat	36
E. Perancangan Pengujian dan Pengambilan Data	39
BAB IV HASIL PENGUJIAN DAN PEMBAHASAN	41
A. Pengujian.....	41
B. Pembahasan.....	45
BAB V KESIMPULAN DAN SARAN.....	50
A. Kesimpulan	50
B. Keterbatasan Alat.....	50
C. Saran	51
DAFTAR PUSTAKA.....	52
LAMPIRAN.....	53

DAFTAR GAMBAR

Gambar 1. <i>Blade wind turbine</i> Sumbu Horizontal.....	7
Gambar.2 <i>Blade wind turbine</i> Sumbu Vertikal.....	8
Gambar 3. Blok diagram Mikrokontroler AVR ATmega32.....	14
Gambar 4. Konfigurasi pin Mikrokontroler AVR ATmega32.....	14
Gambar 5. Sensor Arus	18
Gambar 6. Efek <i>Hall</i> oleh ACS712	19
Gambar 7. Rangkaian pembagi tegangan.....	20
Gambar 8. Modul SD-Card.....	21
Gambar 9. LCD 16x2.....	24
Gambar 10. Blok Diagram Alat	27
Gambar 11. Perancangan Catu Daya	30
Gambar 12. Perencanaan Sensor Tegangan.....	31
Gambar 13. Perancangan Sensor Arus.....	32
Gambar 14. Rangkaian Modul SD.....	33
Gambar 15. Rangkaian LCD 16x2.....	33
Gambar 16. Rangkaian Atmega 32	34
Gambar 16. Rangkaian Alat Monitoring Kualitas Daya.....	35
Gambar 17. Skematik alat monitoring data.....	35
Gambar 18. Layout alat monitoring data	36
Gambar 19. Flowchart.....	37
Gambar 20. Grafik peningkatan tegangan output generator	46

Gambar 21. Grafik pengaruh kecepatan angin terhadap tegangan output dan arus generator pada kondisi berbeban.....	47
Gambar 22. File DataLogger pada SD-Card.....	48
Gambar 23. Grafik perbandingan.....	49

DAFTAR TABEL

Tabel 1. Fungsi Pin I/O <i>Port</i> A pada ATmega 32	15
Tabel 2. Fungsi Pin I/O <i>Port</i> B pada ATmega 32.....	16
Tabel 3. Fungsi Pin I/O <i>Port</i> C pada ATmega 32.....	17
Tabel 4. Fungsi Pin I/O <i>Port</i> D pada ATmega 32	18
Tabel 5. Keterangan dan Fungsi <i>Port</i> pada Modul SD-Card.....	22
Tabel 6. Daftar Peralatan.....	28
Tabel 7. Bahan-Bahan yang Digunakan.....	29
Tabel 11. Data hasil pengujian tanpa beban.....	42
Tabel 12. Data hasil pengujian dengan beban.....	44
Tabel 13. Data hasil pengujian sistem penyimpanan data	44

DAFTAR LAMPIRAN

Lampiran 1. Foto Uji Kerja Kincir Angin.....	54
Lampiran 2. Rangkaian Keseluruhan	55
Lampiran 3. Program	56

BAB I

PENDAHULUAN

A. Latar Belakang

Energi merupakan bagian yang sangat penting dalam kehidupan sehari-hari pada masyarakat dan alam sekitar, karena hampir semua aktivitas manusia selalu membutuhkan energi. Diantaranya energi ini digunakan untuk pertanian, perkebunan, penerangan dan proses perindustrian dalam pengoprasian alat.

Sebagai contoh energi listrik, kebutuhan listrik di Indonesia semakin lama semakin bertambah dan sebagian besar berasal dari pembangkit dengan sumber energi fosil berupa batu bara. Sumber energi fosil ini merupakan energi yang tak terbarukan (*unrenewable energy*). Sehingga ketersediaan sumber fosil ini semakin lama akan semakin habis.

Sebagai negara kepulauan Indonesia merupakan negara yang mempunyai banyak pulau dan tepian pantai. Letak geografis Indonesia mempunyai sumber energi terbarukan yang banyak sehingga dapat dikembangkan dan dipotensikan untuk pemanfaatan energi menjadi energi jenis lain. Diantaranya adalah sumber energi angin. Potensi energi angin di Indonesia cukup memadai.

Potensi kecepatan angin dipantai baru Pandansimo Bantul Yogyakarta dengan kecepatan 2,8 s/d 7,2 m/s sangat layak untuk dimanfaatkan untuk sumber energi. Energi yang dihasilkan dari angin di pantai baru tentunya harus mempertimbangkan desain dimensi *blade* yang maksimal untuk menghasilkan

energi yang tinggi. Oleh karena itu kami merancang desain *blade* dengan sistem *blade* sayap elang yang sesuai dengan kecepatan angin di pantai baru.

Melalui penelitian ini kami mengembangkan pembangkit listrik bertenaga angin sebagai pemanfaatan *energy* terbarukan yang bisa dijadikan sebagai unggulan mengingat panjangnya pesisir pantai di Indonesia. Pengkajian secara mendalam mengenai kincir angin harus selalu dikembangkan sehingga setiap daerah yang mempunyai potensi *energy* angin bisa menjadi daerah mandiri.

Pembangkit Listrik Tenaga Angin akan bekerja berdasarkan kekuatan angin yang ada. Kondisi angin ini tidak selalu tetap, terkadang sangat kuat dan terkadang juga lemah. Kondisi angin ini akan mempengaruhi tingkat kekuatan dari Pembangkit Listrik Tenaga Angin dalam menghasilkan energi listrik. Kondisi angin yang tidak stabil ini memepersulit untuk mengetahui dari daya arus dan tegangan yang dihasilkan oleh pembangkit listrik tenaga angin ini.

Berdasarkan hal-hal tersebut penulis merancang alat monitoring data *logger* Pembangkit Listrik Tenaga Angin. Alat monitoring data *logger* dikembangkan menjadi alat yang murah, ekonomis, akurat dan praktis. Alat monitoring data *logger* ini dirancang menyesuaikan dengan perkembangan teknologi saat ini yaitu pengembangan teknologi berbasis mikrokontroler. Alat monitoring data *logger* berbasis mikrokontroler merupakan alat monitoring yang dihasilkan oleh Pembangkit Listrik Tenaga Angin. Keluaran dari alat monitoring data *logger* ini meliputi tegangan, arus, dan daya yang kemudian ditampilkan pada LCD. Alat monitoring data *logger* dibuat dengan harapan dapat mempermudah pembacaan data dari keluaran pembangkit listrik tenaga angin.

B. Identifikasi Masalah

Dari latar belakang tersebut identifikasi masalah yang ada adalah sebagai berikut :

1. Pengembangan kincir angin listrik masih belum terlalu banyak dikembangkan.
2. Belum adanya alat monitoring pada Pembangkit Listrik Tenaga Angin.
3. Perlunya alat monitoring yang multiguna pada Pembangkit Listrik Tenaga Angin.

C. Batasan masalah

Berdasarkan identifikasi masalah yang telah diuraikan, batasan masalah dari sistem ini yaitu pada sistem monitoring data *logger* kincir angin listrik berbasis ATmega32. Dalam Proyek akhir ini penulis membatasi masalah dalam monitoring pembangkit listrik tenaga angin berupa tegangan, arus, daya yang terpakai pada pembangkit listrik tenaga angin.

D. Rumusan masalah

Berdasarkan pemaparan di atas maka rumusan pada sistem monitoring ini yaitu sebagai berikut :

1. Bagaimana tahap perancangan alat monitoring data loger kincir angin listrik?
2. Bagaimana pembuatan alat monitoring data loger kincir angin listrik?
3. Bagaimana unjuk kerja dari alat monitoring data loger kincir angin listrik?

E. Tujuan

Pembuatan proyek akhir ini mempunyai tujuan sebagai berikut:

1. Mengetahui cara merancang sistem monitoring data loger kicir angin listrik
2. Mengetahui pembuatan alat monitoring data loger kincir angin listrik.
3. Mengetahui unjuk kerja data loger sebagai sistem monitoring data pada Pembangkit Listrik Tenaga Angin.

F. Manfaat penulisan

1. Bagi mahasiswa :
 - a. Dapat mempratekkan ilmu teori yang sudah didapatkan pada perkuliahan.
 - b. Mahasiwa mendapatkan pengalaman dalam pembuatan alat berupa monitoring Pembangkit Listrik Tenaga Angin.
 - c. Mahasiswa mampu mengetahui kondisi dilapangan terkadang tak semudah diteori.
 - d. Mahasiswa mampu berfikir kreatif dan kritis dalam pengembangan ide dan kebutuhan dalam masyarakat dalam perancanganPembangkit Listrik Tenaga Angin.
2. Bagi masyarakat :
 - a. Masyarakat dalam belajar dalam pengembangan kincir angin.
 - b. Masyarakat dapat mengetahui cara perawatan kincir angin.
 - c. Dapat membantu masyarakat dalam memenuhi kebutuhan sehari-hari dan operasional.

- d. Dapat memancing masyarakat menjadi masyarakat yang mandiri dan tidak bergantung pada pemerintah.
3. Bagi institusi :
- a. Dapat mendorong institusi untuk ikut serta dalam pengembangan alat.
 - b. Dapat menjadi referensi dalam pengembangan kreatifitas mahasiswa dalam perkuliahan.
 - c. Mendorong institusi untuk membangun kincir angin untuk media pembelajaran secara nyata.

G. Keaslian Gagasan

Pengembangan energi terbarukan khususnya kincir angin listrik belum terlalu banyak dikembangkan, dalam proyek akhir ini difokuskan pada pembuatan Alat Monitoring data logger kincir angin listrik di pantai baru pandansimo dengan sistem yang sudah dipaparkan dalam latar belakang laporan proyek akhir ini. Sejauh pengetahuan penulis belum pernah ada pengembangan proyek akhir yang mengkhususkan pada proses pembuatan alat monitoring data logger khususnya dilakukan dikalangan mahasiswa jurusan Teknik Elektronika FT UNY maupun mahasiswa UNY.

BAB II

KAJIAN TEORI

A. Kincir Angin

Pembangkit listrik tenaga angin, memanfaatkan energi angin sebagai sumber energinya. Pemanfaatan energi angin ini yaitu menggunakan kincir angin lalu dihubungkan menggunakan generator ataupun turbin. Setelah itu, proses yang dilakukan akan menghasilkan tenaga listrik yang dapat dimanfaatkan sebagai sumber energi. Energi angin merupakan bentuk yang jauh berkelanjutan bebas dengan polusi energi. Pemanfaatan angin ini memang sangat disarankan karena jumlahnya yang tidak terbatas dan juga melimpah. Pemanfaatan energi angin ini sangat menarik karena tidak perlu menggunakan bahan bakar sebagai sumber energi. Tidak hanya itu, pemanfaatan energi angin ini juga tidak memberikan hasil gas rumah kaca dan juga limbah ataupun racun yang berlebihan. Energi ini berasal dari energi kinetik yang dikonversi dan hadir dalam bentuk angin. Kemudian angin diolah menjadi bentuk yang lebih bermanfaat atau berguna.

Ada dua jenis kincir angin yang umum digunakan saat ini, yaitu berdasarkan arah poros berputar (sumbu): turbin angin sumbu horisontal dan turbin angin sumbu vertikal.

1. Blade wind turbine Sumbu Horisontal

Kebanyakan turbin angin yang digunakan saat ini adalah tipe sumbu horisontal. Turbin angin sumbu horisontal memiliki bilah baling-baling

seperti di pesawat. Turbin angin jenis ini memiliki shaft rotor dan generator pada puncak tower dan harus diarahkan ke arah angin bertiup.



Gambar 1. *Blade wind turbine* Sumbu Horizontal

Berdasarkan penjelasan tentang *wind turbine* sumbu horizontal berikut kekurangan dan kelebihan dari jenis turbin angin poros horizontal tersebut.

Kekurangan turbin angin horizontal :

- a. Membutuhkan konstruksi tower yang besar untuk mendukung beban gear box, blade dan juga generator.
- b. Membutuhkan sistem pengereman untuk mencegah turbin mengalami kerusakan pada turbin ketika ada angin kencang.
- c. Membutuhkan pengawasan dan kontrol secara berkala untuk mengarahkan blade ke arah angin.

Kelebihan turbin angin horizontal :

- a. Towernya yang tinggi memungkinkan untuk mendapatkan angin dengan kekuatan yang lebih besar untuk mendapatkan energi.

b. Efisiensi lebih tinggi. Hal ini dikarenakan blade selalu bergerak tegak lurus terhadap angin.

2. Turbin Angin Sumbu Vertikal

Jenis turbin angin yang kedua adalah turbin angin poros vertikal. Turbin angin jenis ini memiliki bilah yang memanjang dari atas ke bawah. Turbin angin vertikal biasanya berdiri setinggi 100 meter dengan lebar 50 kaki. Dengan sumbu vertikal, generator dan komponen primer lainnya dapat ditempatkan dengan permukaan tanah, sehingga tentu saja ini dapat mempermudah maintenance lebih mudah. Jika dibandingkan dengan turbin angin poros horizontal, turbin angin ini memiliki kecepatan yang lambat, sehingga energi angin yang tersedia pun lebih rendah.



Gambar.2 *Blade wind turbine* Sumbu Vertikal

Kekurangan turbin angin poros vertical :

a. Memiliki penurunan efisiensi. Jika dibandingkan dengan turbin angin poros horizontal, turbin angin poros vertikal memiliki

penurunan efisiensi. Hal ini dikarenakan adanya hambatan tambahan yang mereka miliki sebagai pisau memutar ke angin.

- b. Memiliki kecepatan angin yang rendah. Yang kedua adalah jenis memiliki kecepatan angin yang rendah. Karena turbin angin poros vertikal memiliki rotor dekat dengan tanah.

Kelebihan turbin angin poros vertikal:

- a. Yang pertama adalah turbin angin tidak memerlukan perawatan yang ekstra. Sehingga tidak membutuhkan biaya yang lebih banyak untuk merawatnya.
- b. Yang kedua adalah turbin angin juga sangat mudah dirawat karena letaknya yang dekat dengan tanah.

Turbin angin ini memiliki kecepatan stratum angin rendah dibandingkan dengan turbin horizontal.

B. Data Logger

Data logger (perekam data) adalah sebuah alat elektronik yang mencatat data dari waktu ke waktu baik yang terintegrasi dengan sensor dan instrumen didalamnya maupun eksternal sensor dan instrumen. Atau secara singkat data logger adalah alat untuk melakukan data logging. Biasanya ukuran fisiknya kecil, bertenaga baterai, portabel, dan dilengkapi dengan mikroprosesor, memori internal untuk menyimpan data dan sensor. Beberapa *data logger* diantarmukakan dengan komputer dan menggunakan *software* untuk mengaktifkan *data logger* untuk melihat dan menganalisa data yang terkumpul, sementara yang lain memiliki peralatan antarmuka sendiri (*keypad* dan *LCD*)

dan dapat digunakan sebagai perangkat yang berdiri sendiri (*Stand-alone device*). Salah satu keuntungan menggunakan *data logger* adalah kemampuannya secara otomatis mengumpulkan data setiap 24 jam. Setelah diaktifkan, *data logger* digunakan dan ditinggalkan untuk mengukur serta merekam informasi selama periode pemantauan. Hal ini memungkinkan untuk mendapatkan gambaran yang komprehensif tentang kondisi lingkungan yang dipantau.

1. Tegangan

Tegangan yang diukur pada alat monitoring data logger adalah tegangan DC. Tegangan DC merupakan tegangan arus searah. Tegangan arus searah adalah arus listrik yang mengalir pada suatu hantaran yang tegangannya berpotensi tetap dan tidak berubah-ubah. Listrik DC adalah listrik yang original, artinya listrik dasar yang dapat dihasilkan dari sumber-sumber susunan material alam. Tegangan DC arus listrik ini bergerak dari kutub yang selalu sama, yaitu dari kutub positif ke kutub negative dan polaritas arus ini selalu tetap. Sumber arus searah misalnya aki, baterai, beberapa jenis elemen dan generator searah. Tegangan DC sumber arus ini biasanya ditandai adanya kutub positif dan kutub negatif.

2. Arus Listrik

Arus listrik adalah aliran dari muatan listrik dari satu titik ke titik yang lain. Arus listrik terjadi karena adanya media penghantar antara dua titik yang mempunyai beda potensial. Semakin besar beda potensial listrik antara dua titik tersebut maka semakin besar pula arus yang mengalir. Dari aliran

arus listrik inilah diperoleh tenaga listrik yang disebut dengan daya. Satuan kuat arus dinyatakan dalam Ampere atau disingkat dengan huruf A besar. Nilai kuat arus sebesar 1 Ampere adalah aliran muatan listrik sejumlah 1 Coloumb dalam waktu 1 detik. Muatan listrik adalah satuan terkecil dari atom. Dalam inti atom terdapat muatan positif yang disebut proton dan muatan netral yang disebut neutron. Sedangkan pada kulit atom terdapat muatan negatif yang disebut elektron. Satuan kuat arus yang lebih besar bisa dinyatakan dalam kiloAmpere disingkat kA ($1\text{kA}=1000\text{A}$) dan untuk satuan yang lebih kecil bisa dinyatakan dalam miliAmpere disingkat mA ($1\text{mA}=1/1000\text{A}$).

Arus Searah (DC)

Arus searah mengalir secara searah dari titik yang memiliki potensial tinggi ke titik yang memiliki potensial lebih rendah. Meskipun sebenarnya yang mengalir adalah elektron (muatan negatif) namun disepakati bahwa yang mengalir adalah arus positif, dari kutub positif ke kutub negatif. Jika dilihat bentuk gelombangnya dengan oscilloscope, arus searah terlihat sebagai garis lurus.

Arus Bolak-balik (AC)

Sedangkan arus bolak-balik memiliki aliran arus yang berubah-ubah arahnya. Perubahan arah arus bolak-balik ini mengikuti garis waktu sehingga jika dilihat dengan oscilloscope, arus bolak-balik membentuk sebuah gelombang dengan frekuensi tertentu. Bentuk gelombang arus bolak-balik ada yang beraturan dan tidak beraturan

3. Daya

Daya listrik merupakan bagian dari besarnya beda potensial, kuat arus, hambatan dan waktu. Satuan daya adalah joule/sekon atau volt \times ampere atau lebih umum disebut watt, karena watt merupakan satuan Sistem Internasional. Oleh karena itu daya dapat dirumuskan dengan rumus yang ditunjukkan pada persamaan 1:

$$P = W/t \dots\dots\dots (1)$$

Keterangan :

P = Daya (Watt)

W = Usaha (Joule)

t = Waktu (Sekon)

Berdasarkan persamaan (1) dapat disimpulkan bahwa daya ini terdapat pada tegangan searah atau bolak-balik. Akan tetapi dari perbedaan tersebut daya pada tegangan DC berbeda dengan tegangan AC. Oleh karena itu rumus yang digunakan untuk menentukan daya pada tegangan DC ditunjukkan pada persamaan 2, persamaan 3 dan persamaan 4 sebagai berikut :

$$P = I^2 \cdot R \dots\dots\dots (2)$$

$$P = V^2/R \dots\dots\dots (3)$$

$$P = V \cdot I \dots\dots\dots (4)$$

Keterangan :

P = Daya (Watt)

I = Arus (Ampere)

V = Tegangan (Volt)

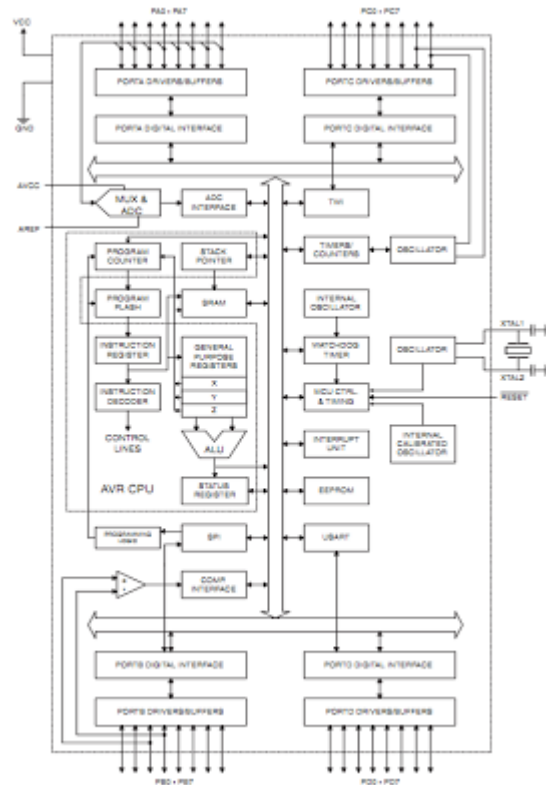
R = Hambatan (Ohm)

Ketiga rumus tersebut merupakan rumus untuk menentukan daya yang terpakai dalam suatu beban. Daya ini timbul jika terdapat arus yang mengalir pada loop tertutup.

C. ATmega 32

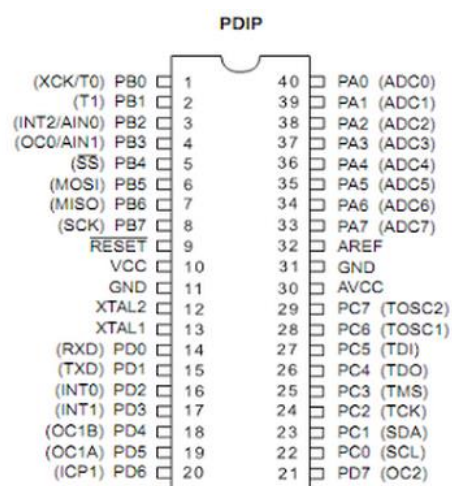
Mikrokontroler AVR ATmega32 merupakan CMOS dengan konsumsi daya rendah, mempunyai 8-bit proses data (CPU) berdasarkan arsitektur AVR RISC. Dengan mengeksekusi instruksi dalam satu (siklus) clock tunggal, ATmega32 memiliki kecepatan data rata-rata (*throughputs*) mendekati 1 MIPS per MHz, yang memungkinkan perancang sistem dapat mengoptimalkan konsumsi daya dan kecepatan pemrosesan. Arsitektur AVR ini menggabungkan perintah secara efektif dengan 32 register umum. Semua register tersebut langsung terhubung dengan *Arithmetic Logic Unit (ALU)* yang memungkinkan 2 register terpisah diproses dengan satu perintah tunggal dalam satu *clock cycle*. Hal ini menghasilkan kode yang efektif dan kecepatan prosesnya 10 kali lebih cepat dari pada mikrokontroler CISC biasa.

Berikut adalah blok diagram Mikrokontroler AVR ATmega32



Gambar 3. Blok diagram Mikrokontroler AVR ATmega32
(Sumber : <http://ignatius.ilearning.me/?p=233>)

Konfigurasi pin Mikrokontroler AVR ATmega32



Gambar 4. Konfigurasi pin Mikrokontroler AVR ATmega32
(Sumber : <http://www.immersa-lab.com/pengenalan-mikrokontroler.htm>)

Secara fungsional konfigurasi pin ATmega32 adalah sebagai berikut:

- a. VCC
 - Tegangan sumber
- b. GND (Ground)
 - Ground
- c. Port A (PA7 – PA0)

Port A adalah 8-bit port I/O yang bersifat *bi-directional* dan setiap pin memiliki *internal pull-up resistor*. *Output buffer port A* dapat mengalirkan arus sebesar 20 mA. Ketika *port A* digunakan sebagai *input* dan di *pull-up* secara langsung, maka *port A* akan mengeluarkan arus jika *internal pull-up resistor* diaktifkan. Pin-pin dari *port A* memiliki fungsi khusus yaitu dapat berfungsi sebagai *channel ADC (Analog to Digital Converter)* sebesar 10 bit.

Tabel 1. Fungsi Pin I/O Port A pada ATmega 32

<i>Port</i>	<i>Alternate Function</i>
<i>PA7</i>	<i>ADC7 (ADC input channel 7)</i>
<i>PA6</i>	<i>ADC6 (ADC input channel 6)</i>
<i>PA5</i>	<i>ADC5 (ADC input channel 5)</i>
<i>PA4</i>	<i>ADC4 (ADC input channel 4)</i>
<i>PA3</i>	<i>ADC3 (ADC input channel 3)</i>
<i>PA2</i>	<i>ADC2 (ADC input channel 2)</i>
<i>PA1</i>	<i>ADC1 (ADC input channel 1)</i>
<i>PA0</i>	<i>ADC0 (ADC input channel 0)</i>

d. Port B (PB7 – PB0)

Port B adalah 8-bit port I/O yang bersifat *bi-directional* dan setiap pin mengandung *internal pull-up resistor*. *Output buffer port B* dapat mengalirkan arus sebesar 20 mA. Ketika *port B* digunakan sebagai *input* dan di *pull-down* secara *external*, *port B* akan mengalirkan arus jika *internal pull-up resistor* diaktifkan.

Pin-pin port B memiliki fungsi-fungsi khusus, diantaranya :

- *SCK port B, bit 7*
Input pin clock untuk up/downloading memory.
- *MISO port B, bit 6*
Pin output data untuk uploading memory.
- *MOSI port B, bit 5*
Pin input data untuk downloading memory.

Tabel 2. Fungsi Pin I/O Port B pada ATmega 32

<i>Port</i>	<i>Alternate Function</i>
<i>PB7</i>	<i>SCK (SPI Bus Serial Clock)</i>
<i>PB6</i>	<i>MISO (SPI Bus Master Input/Slave Output)</i>
<i>PB6</i>	<i>MOSI (SPI Bus Master Output/Slave Input)</i>
<i>PB5</i>	<i>SS (SPI Slave Select Input)</i>
<i>PB3</i>	<i>AIN1 (Analog Comparator Negative Input)</i> <i>OCO (Timer/Counter0 Output Compare Match Output)</i>
<i>PB2</i>	<i>AIN0 (Analog Comparator Positive Input)</i> <i>INT2 (External Interrupt 2 Input)</i>
<i>PB1</i>	<i>T1 (Timer/Counter1 External Counter Input)</i>
<i>PB0</i>	<i>T0 (Timer/Counter External Counter Input) XCK (USART External Clock Input/Output)</i>

e. *Port C (PC7 – PC0)*

Port C adalah 8-bit port I/O yang berfungsi *bi-directional* dan setiap pin memiliki *internal pull-up resistor*. Output buffer port *C* dapat mengalirkan arus sebesar 20 mA. Ketika port *C* digunakan sebagai input dan di *pull-down* secara langsung, maka port *C* akan mengeluarkan arus jika *internal pull-up resistor* diaktifkan.

Tabel 3. Fungsi Pin I/O Port C pada ATmega 32

<i>Port</i>	<i>Alternate Function</i>
<i>PC7</i>	<i>TOSC2 (Timer Oscillator Pin 2)</i>
<i>PC6</i>	<i>TOSC1 (Timer Oscillator Pin 1)</i>
<i>PC6</i>	<i>TD1 (JTAG Test Data In)</i>
<i>PC5</i>	<i>TD0 (JTAG Test Data Out)</i>
<i>PC3</i>	<i>TMS (JTAG Test Mode Select)</i>
<i>PC2</i>	<i>TCK (JTAG Test Clock)</i>
<i>PC1</i>	<i>SDA (Two-wire Serial Bus Data Input/Output Line)</i>
<i>PC0</i>	<i>SCL (Two-wire Serial Bus Clock Line)</i>

f. *Port D (PD7 – PD0)*

Port D adalah 8-bit port I/O yang berfungsi *bi-directional* dan setiap pin memiliki *internal pull-up resistor*. Output buffer port *D* dapat mengalirkan arus sebesar 20 mA. Ketika port *D* digunakan sebagai input dan di *pull-down* secara langsung, maka port *D* akan mengeluarkan arus jika *internal pull-up resistor* diaktifkan.

Tabel 4. Fungsi Pin I/O Port D pada ATmega 32

Port	Alternate Function
PD7	OC2 (Timer / Counter2 Output Compare Match Output)
PD6	ICP1 (Timer/Counter1 Input Capture Pin)
PD6	OC1B (Timer/Counter1 Output Compare B Match Output)
PD5	TD0 (JTAG Test Data Out)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

D. Sensor Arus



Gambar 5. Sensor Arus

(Sumber : <https://www.amazon.com/uxcell-Electrical-Current-Sensor>)

Sensor arus 5 Ampere ini merupakan modul sensor untuk mendeteksi besar arus yang mengalir lewat blok terminal menggunakan *current sensor chip* ACS712-5 yang memanfaatkan efek *Hall*. Besar arus maksimum yang dapat dideteksi sebesar 5A di mana tegangan pada pin keluaran akan berubah secara *linear* mulai dari 2,5 Volt ($\frac{1}{2} \times V_{CC}$, tegangan catu daya $V_{CC} = 5V$) untuk kondisi tidak ada arus hingga 4,5V pada arus sebesar +5A atau 0,5V pada arus sebesar -5A (positif/negatif tergantung polaritas, nilai di bawah 0,5V atau di atas 4,5V dapat dianggap lebih dari batas maksimum). Perubahan tingkat tegangan berkorelasi *linear* terhadap besar arus sebesar 400mV/Ampere. Efek *Hall* adalah fenomena fisika dimana aliran listrik / elektron dalam pelat konduktor

terpengaruh oleh paparan medan magnet. Secara sederhana pemanfaatan efek *Hall* oleh IC ACS712 ini dapat digambarkan sebagai berikut:



Gambar 6. Efek *Hall* oleh ACS712
(Sumber : <http://blog.vcc2gnd.com>)

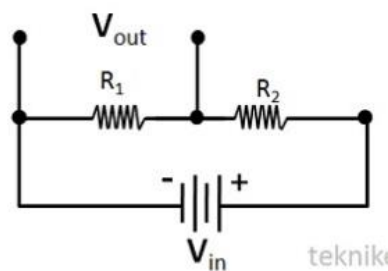
Beberapa fitur penting dari sensor arus ACS-712 ELC05B adalah :

- Jalur sinyal analog yang rendah *noise*.
- *Bandwidth* perangkat diatur melalui pin FILTER yang baru.
- Waktu naik keluaran 5 *mikrodetik* dalam menanggapi langkah masukan aktif.
- *Bandwith* 50 kHz.
- Total *error* keluaran 1,5% pada $T_A = 25^\circ$, dan 4% pada -40° C sampai 85° C
- Bentuk yang kecil, paket SOIC8 yang kompak.
- Resistansi internal 1.2 m Ω .
- 2.1 kV_{RMS} tegangan isolasi minimum dari pin 1-4 ke pin 5-8.
- Operasi catu daya tunggal 5.0 V.

- Sensitivitas keluaran 66-185 mV/A.
- Tegangan keluaran sebanding dengan arus AC atau DC.
- Akurasi sudah diatur oleh pabrik.
- Tegangan offset yang sangat stabil.
- *Hysteresis magnetic* hampir mendekati nol.
- Keluaran *ratiometric* diambil dari sumber daya

E. Sensor Tegangan (rangkaiian pembagi tegangan)

Voltage Divider atau Pembagi Tegangan adalah suatu rangkaian sederhana yang mengubah tegangan besar menjadi tegangan yang lebih kecil. Fungsi dari Pembagi Tegangan ini di Rangkaian Elektronika adalah untuk membagi Tegangan Input menjadi satu atau beberapa Tegangan Output yang diperlukan oleh Komponen lainnya didalam Rangkaian. Hanya dengan menggunakan dua buah Resistor atau lebih dan Tegangan Input



Gambar 7. Rangkaian pembagi tegangan

F. Modul RTC

RTC (*Real time clock*) adalah jam elektronik berupa *chip* yang dapat menghitung waktu (mulai detik hingga tahun) dengan akurat dan menjaga/menyimpan data waktu tersebut secara *real time*. Karena jam tersebut

bekerja *real time*, maka setelah proses hitung waktu dilakukan *output* datanya langsung disimpan atau dikirim ke *device* lain melalui sistem antarmuka.

Chip RTC sering dijumpai pada *motherboard* PC (biasanya terletak dekat *chip* BIOS). Semua komputer menggunakan RTC karena berfungsi menyimpan informasi jam terkini dari komputer yang bersangkutan. RTC dilengkapi dengan baterai sebagai pensuplai daya pada *chip*, sehingga jam akan tetap *up-to-date* walaupun komputer dimatikan. RTC dinilai cukup akurat sebagai pewaktu (*timer*) karena menggunakan *osilator* kristal.

G. Modul SD

Modul SD Card adalah sebuah modul yang berfungsi untuk membaca dan menulis data ke/dari SD Card. Modul ini memiliki *interfacing* menggunakan komunikasi SPI. Tegangan kerja dari modul ini dapat menggunakan level tegangan 3.3 V DC atau 5V DC, yang dapat digunakan salah satunya.

Bentuk fisik dari Modul SD ditunjukkan pada Gambar 8.



Gambar 8. Modul SD-Card
(Sumber : <https://www.raspberrypi.org/>)

Modul ini digunakan untuk membuat piranti-piranti yang membutuhkan suatu penyimpanan bersifat *non-volatile* (data akan tetap tersimpan walaupun tidak mendapatkan supply tegangan) dengan kapasitas besar, hingga mencapai Gigabyte. Modul ini banyak digunakan untuk pembuatan perekaman medis,

perekam dan *playback* musik, *data logger* dan juga untuk pembuatan basis data.

Berikut ini adalah pin dalam modul sd card, diantaranya :

1. GND
2. VCC 3.3V
3. VCC 5V
4. CS
5. MOSI
6. SCK
7. MISO
8. GND

Pin-pin diatas merupakan pin yang terdapat dalam sd card yang mempunyai fungsi secara tersendiri. Berikut ini adalah beberapa fungsi dari pin pada sd card ditunjukkan pada tabel 5.

Tabel 5. Keterangan dan Fungsi *Port* pada Modul SD-Card

Pin	Nama	Fungsi	Keterangan
1	GND	Input	Referensi <i>Ground</i>
2	+3.3V	Input	Tegangan <i>reverensi</i>
3	+5 V	Input	Terhubung keSumber Tegangan +5 VDC
4	CSSD		<i>Chip Select</i> , diberi logika 0 untuk mengakses SD Card, diberi logika 1 jika tidak mengakses SD Card.
5	MOSI	Input	Jalur data masuk ke SD Card
6	SCK	Input	Jalur clock dari mikrokontroler untuk mengakses SD Card
7	MISO	Output	Jalur data keluar dari SD Card
8	GND	Input	Referensi <i>Ground</i>

Keterangan diatas bahwa modul sd card ini digunakan sebagai piranti untuk menghubungkan data dari sumber data ke sd card atau mengambil data dari sd card ke suatu program atau oleh data. Disamping itu alat monitoring data ini akan dapat berfungsi dengan baik jika dibutuhkan piranti-piranti pendukung dalam suatu sistem pada pembangkit listrik tenaga angin.

H. LCD

Liquid Cristal Display (LCD) adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf atau angka. *Liquid Cristal Display (LCD)* adalah salah satu jenis display elektronik yang dibuat dengan teknologi CMOS *logic* yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*.

Material *Liquid Cristal Display (LCD)* adalah sebuah lapisan dari campuran organik antara lapisan kaca bening dengan elektroda transparan *indium oksida* dalam bentuk tampilan *seven-segment* dan lapisan elektroda pada kaca belakang. Ketika elektroda diaktifkan dengan medan listrik (tegangan), molekul organik yang panjang dan silindris menyesuaikan diri dengan elektroda dari *segmen*. Lapisan *sandwich* memiliki *polarizer* cahaya *vertikal* depan dan *polarizer* cahaya *horisontal* belakang yang diikuti dengan lapisan reflektor. Cahaya yang dipantulkan tidak dapat melewati molekul-molekul yang telah menyesuaikan diri dan segmen yang diaktifkan terlihat menjadi gelap dan membentuk karakter data yang ingin ditampilkan.

LCD yang digunakan pada alat monitoring data ini menggunakan lcd16x2. Lcd 16x2 ini merupakan LCD yang terdiri dari 2 baris dan 16 karakter. Bentuk fisik dari cd 16x2 ini dapat dilihat pada Gambar 9.



Gambar 9. LCD 16x2

(Sumber : <https://www.cedelettronica.com/>)

Kontroler LCD (*Liquid Cristal Display*) ini juga sudah dilengkapi dengan modul program LCD (*Liquid Cristal Display*) yang terdapat pada mikrokontroler yang berfungsi sebagai pengendali tampilan karakter pada LCD.

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

Perancangan alat monitoring data logger ini beracuan pada lingkungan sekitar pantai baru pandansimo yang memanfaatkan energi listrik dari kincir angin. Alat monitoring data logger ini digunakan untuk mempermudah pembacaan dari arus tegangan dan daya yg dikeluarkan dari kincir angin pembangkit listrik. Berdasarkan hal tersebut metode rancang bangun merupakan metode yang tepat. Secara berurutan metode tersebut adalah identifikasi kebutuhan yang diperlukan. Kemudian kebutuhan tersebut dianalisis untuk mendapatkan komponen secara spesifik. Selanjutnya dilakukan perancangan secara *hardware* dan *software* pembuatan serta pengujian.

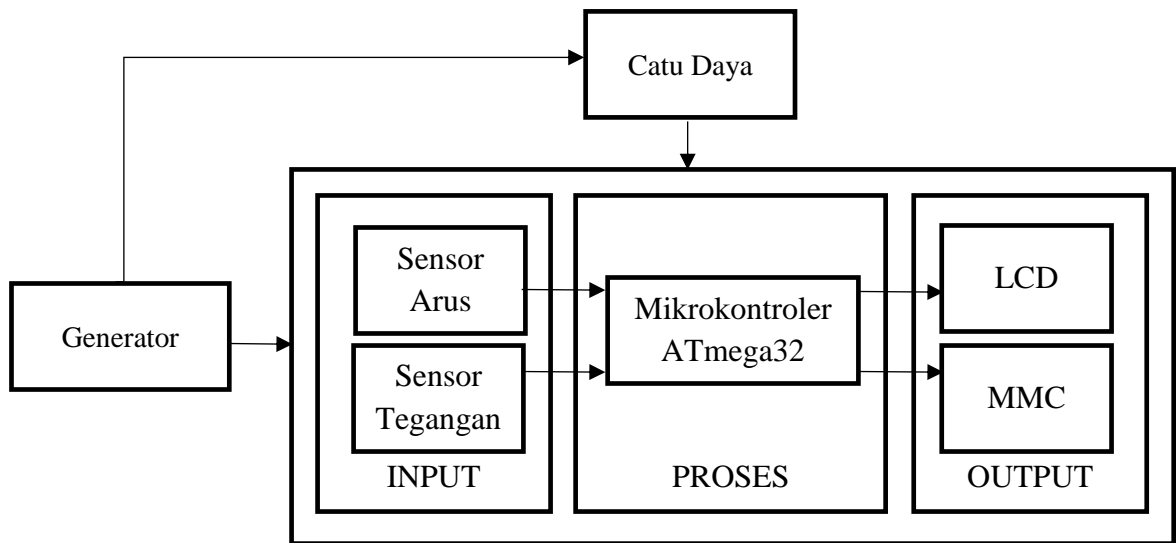
A. Identifikasi Kebutuhan

Identifikasi kebutuhan sebagai berikut :

1. Perancangan catu daya untuk power supply
2. Perancangan sensor tegangan untuk memproses tegangan yang dikeluarkan dari kincir angin.
3. Perancangan sensor arus untuk memproses arus yang dikeluarkan dari kincir angin.
4. Modul SD Card digunakan untuk mentransfer data ke SD card yang didapat dari kincir angin untuk bisa di buka dengan menggunakan PC
5. LCD untuk menampilkan data arus tegangan dan daya yang diperoleh.
6. ATmega 32 sebagai kontroler data logger.

B. Analisis Kebutuhan

Alat monitoring data pada pembangkit listrik tenaga angin ini merupakan alat yang dirancang untuk memonitoring kinerja pembangkit listrik tenaga angin. Saat ini belum banyak pengembangan monitoring pembangkit listrik tenaga angin, oleh karena itu tujuan pembuatan alat ini diharapkan dapat menjadi pengembangan teknologi yang murah dan handal dalam kaitannya mengetahui parameter-parameter pada pembangkit listrik tenaga angin. Alat monitoring ini digunakan untuk mempermudah konsumen/pengguna dalam mengetahui kondisi pembangkit listrik tenaga angin disetiap detik selama 24 jam. Alat ini merupakan alat yang multiguna dan ekonomis dikarenakan alat ini mampu bekerja pada *range* tegangan dan arus yang besar, sehingga alat ini mampu ditempatkan pada jenis kincir angin yang berkapasitas besar. Data-data yang diolah dapat berupa tegangan, arus, dan daya pada pembangkit listrik tenaga angin. Sehingga konsumen/pengguna tidak perlu memonitoring pada tempat dan waktu saat itu juga untuk mengetahui keluaran arus, tegangan dan daya dari pembangkit listrik tenaga angin tersebut. Alat monitoring data dilengkapi dengan sistem penyimpanan data otomatis atau *auto saving sistem* yang akan memudahkan konsumen untuk mengetahui kinerja dari pembangkit listrik tenaga angin tersebut dalam 24 jam yang disimpan dalam satu *chip memory*.



Gambar 10. Blok Diagram Alat

a. Blok Catu Daya

Generator memberikan sumber untuk catu daya yang kemudian menghidupkan data logger yang meliputi sensor arus, sensor tegangan, mikrokontroler ATmega32 LDC dan MMC.

b. Blok *Input*

Blok *input* disini ialah dari generator menghasilkan arus dan tegangan, kemudian dibaca oleh sensor arus dan tegangan untuk mengetahui keluaran dari generator pembangkit listrik tenaga angin. Kemudian data akan dikirim untuk diproses pada mikrokontroler

c. Blok Proses

Sistem kontrol yang digunakan adalah mikrokontroler ATmega32 dengan rancang bangun yang disesuaikan agar sesuai dengan apa yang dibutuhkan.

d. Blok *Output*

Pada monitoring data logger kincir angin listrik menggunakan *output* berupa LCD dan SD Card. Data yang telah diproses pada mikrokontroler akan ditampilkan pada LCD kemudian tersimpan pada SD Card yang dapat dilihat dengan

Proses pembuatan alat monitoring data ini membutuhkan peralatan-peralatan dan sarana bahan untuk menunjang dalam pembuatan alat monitoring data. Beberapa peralatan yang dibutuhkan dalam pembuatan alat monitoring data ditunjukkan pada Tabel 6.

Tabel 6. Daftar Peralatan

No	Nama Alat	Spesifikasi	Jumlah	Satuan
1.	Bor listrik	Bor tangan	1	Buah
2.	Mata bor	0.8 mm	1	Buah
		1 mm	1	Buah
		3 mm	1	Buah
		5 mm	1	Buah
3.	Obeng+	-	1	Buah
4.	Obeng-	-	1	Buah
5	<i>Multimeter</i>	Digital	1	Buah
6	Pengupas Kabel	-	1	Buah
7	Pisau cutter	-	1	Buah
8	Penggaris	60cm	1	Buah
9	Tang potong	-	1	Buah
10	Solder	40 watt	1	Buah
11	Kikir	Pipih	1	Buah
12	Gergaji besi	-	1	Buah
13	Anemo meter	-	1	Buah
14	Tang cucut	-	1	Buah
15	Tang Kombinasi	-	1	Buah

Dari alat-alat yang dibutuhkan diatas berikut ini adalah daftar kebutuhan

bahan yang dibutuhkan untuk menunjang kinerja alat ditunjukkan pada Tabel7 :

Tabel 7. Bahan-Bahan yang Digunakan

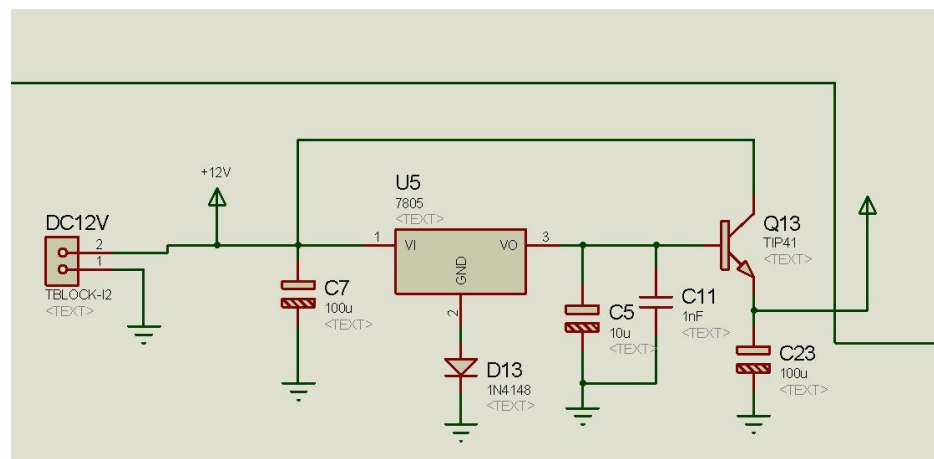
No	Nama bahan	Spesifikasi	Jumlah	Satuan
1.	<i>Blade</i>	Aluminium	1	Buah
2.	Tiang penyangga	Besi	1	Buah
3.	Generator DC	500W. 3500rpm	1	Buah
4.	Baterai	20 Ah	1	Buah
6.	Lampu Pijar	5 Watt	1	Buah
7.	Tenol	-	1	Gulung
8.	PCB	-	1	Buah
9.	Kabel Nyaf	2.5mm	1	Gulung
10.	Kotak Kontak	Isi 1	1	Buah
11.	Saklar On/Off	-	2	Buah
12.	Penjepit Buaya	-	2	Buah
13.	Mur Baut	-	30	Buah
14.	Box	-	1	Buah
15.	Colokan	Steker	1	Buah
16.	Led	5mm, 12 V	5	Buah
17.	Mikrokontroler	-	1	Unit
18.	Lem G	-	1	Buah
19.	Resistor	15w/100Ohm	4	Buah
20.	Lampu	35w/12V	10	Buah
21.	Kotak Kontak	220V	1	Buah
22.	Kabel Nya	1.5 mm ²	10 m	Buah
23.	Fiting	1	1	Buah
24.	Akrilik	30x40cm	1	Buah
25.	Box Plastik	Tebal : 2 mm	1	Buah
26.	Modul Sd		1	Buah
27.	LCD 16x2		1	Buah
28.	Sensor Tegangan		1	Buah
29.	Sensor Arus		1	Buah

C. Tahap Perancangan dan Pembuatan

Tahap perancangan merupakan tahapan untuk merancang rangkaian yang dibutuhkan dalam pembuatan alat monitoring kualitas daya. Proses perancangan dan pembuatan alat monitoring kualitas daya ini melalui beberapa tahap. Tahap-tahap perancangan dan pembuatan meliputi perancangan penggunaan catu daya dan perancangan penggunaan sensor dan perancangan box pelindung.

1. Perancangan Catu Daya

Catu daya adalah rangkaian yang digunakan untuk mensuplay tegangan ke alat monitoring data logger. Rangkaian catu daya bekerja dengan tegangan dari batre 12 VDC, direduksi dengan regulator seri 7805, sehingga tegangannya menjadi 5 VDC, output tersebut digunakan untuk mensuplay sistem minimum.



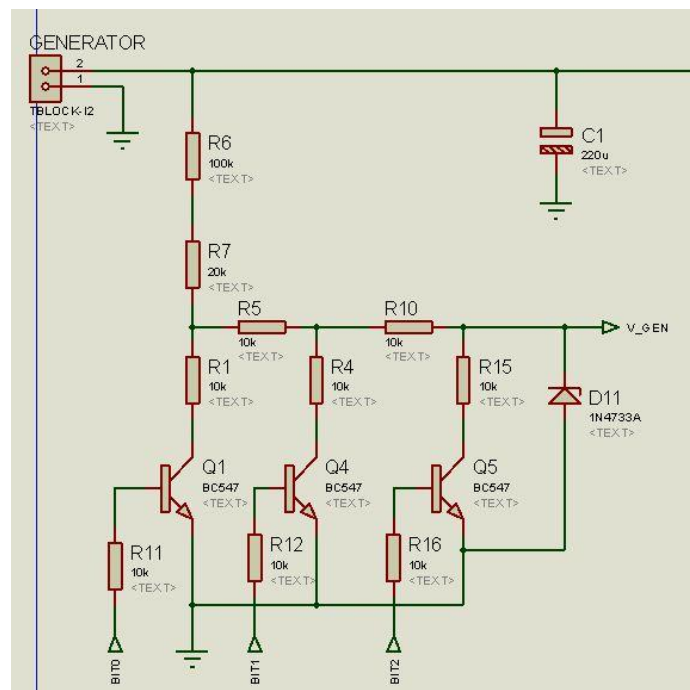
Gambar 11. Perancangan Catu Daya

2. Perancangan Penggunaan Sensor dan Komponen

Perancangan penggunaan sensor ini meliputi perancangan penggunaan sensor arus, tegangan.

a. Perencanaan Sensor Tegangan

Sensor tegangan pada alat monitoring data berfungsi sebagai input data tegangan. Sensor ini berfungsi menurunkan tegangan, dari tegangan yang diukur menjadi tegangan yang diijinkan pada mikrokontroler. Berikut ini adalah rangkaian sensor tegangan yang ditunjukkan pada Gambar 11.



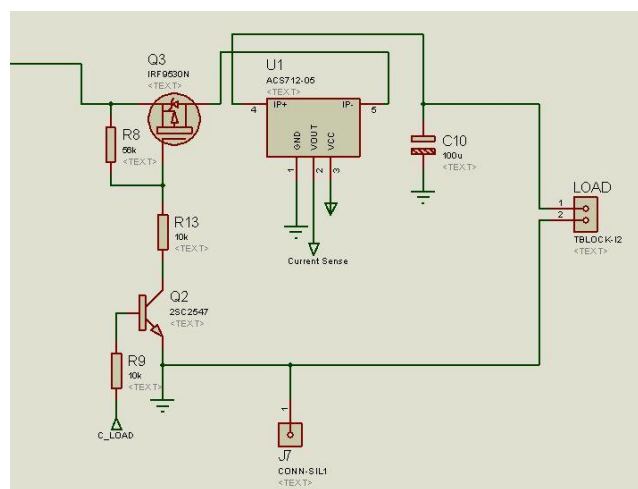
Gambar 12. Perencanaan Sensor Tegangan

Prinsip catu daya yang digunakan adalah tegangan inputan dari generator masuk ke resistor untuk dikurangi nilai tegangan dan arusnya. tegangan tersebut masuk pada kaki kolektor, transistor seri BC

547. Kemudian data disimpan dalam memori melalui modul, data memori tersebut dapat dilihat dengan PC data berupa txt.

b. Perencanaan Sensor Arus

Perancangan pemilihan sensor arus menggunakan komponen sensor ACS712. Perancangan sensor arus dengan menggunakan sensor ACS712 dapat dilihat pada Gambar 12 :

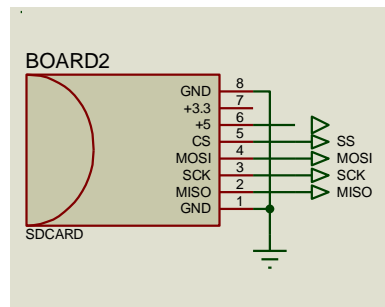


Gambar 13. Perancangan Sensor Arus

Rangkaian tersebut bekerja dengan prinsip sensor ACS712 memperoleh masukan arus dan melakukan pembacaan dari arus yang masuk, arus tersebut mengeluarkan ADC yang dibaca oleh mikrokontroler. kemudian data disimpan dalam memori melalui modul, data memori tersebut dapat dilihat dengan PC data berupa txt.

c. Perencanaan Penggunaan Modul SD Card

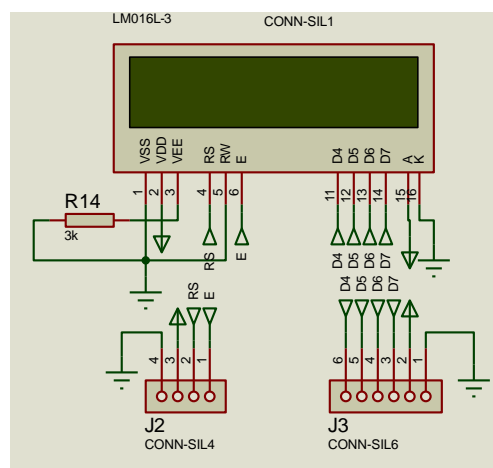
Modul SD Card ini digunakan untuk menyimpan data berupa tegangan, arus, daya dan kecepatan blade. Berikut ini adalah rangkaian modul sd yang akan digunakan ditunjukkan pada Gambar 14 :



Gambar 14. Rangkaian Modul SD
(Sumber : <https://www.raspberrypi.org/>)

d. Perencanaan Penggunaan LCD 16x2

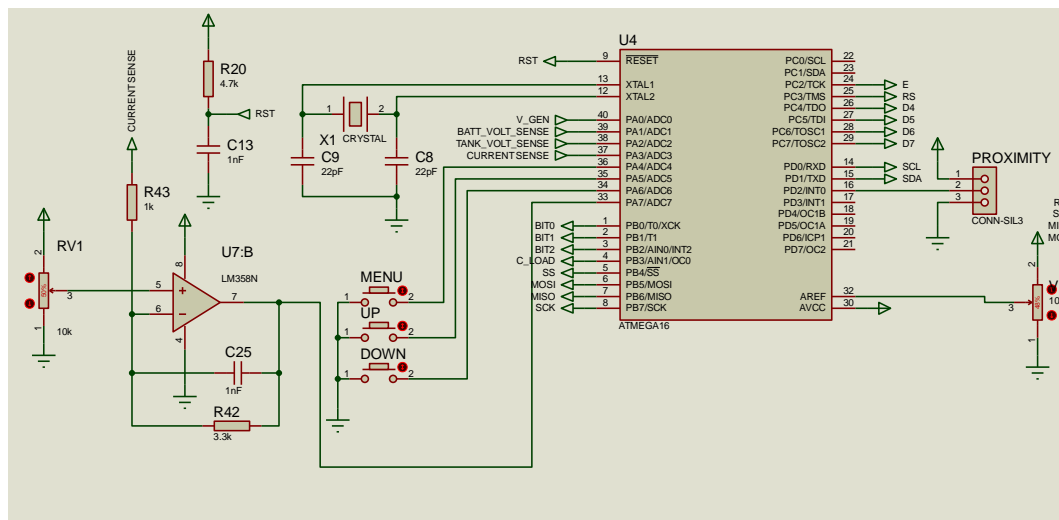
Penggunaan LCD 16x2 ini digunakan untuk menampilkan data yang sudah didapatkan dan diolah pada mikrokontroler. LCD ini digunakan sebagai penampil data secara visual berdasarkan pengukuran dari sensor-sensor tersebut. Lcd 16x2 ini merupakan lcd yang mempunyai 16 karakter dan 2 baris. Dengan menggunakan lcd 16x2 cukup untuk menampilkan data-data atau parameter yang diolah oleh mikrokontroler. Berikut ini adalah rangkaian LCD 16x2 ditunjukan pada Gambar 15.



Gambar 15. Rangkaian LCD 16x2
(Sumber : <https://www.cedeletronica.com/>)

e. Perencanaan Penggunaan ATmega 32

Penggunaan Atmega 32 pada proses pembuatan alat monitoring kualitas daya digunakan untuk pengolah data saat penyimpanan. Penggunaan ATmega 32 hal ini dikarenakan untuk mempermudah penyimpanan pada modul sd card, dikarenakan penyimpanan data pada ATmega 32 lebih besar yaitu sebesar 32 bit. Sistem penyimpanan dengan menggunakan ATmega 32 akan dikendalikan dengan program, program yang digunakan yaitu dengan program CV AVR. Oleh karena itu dengan pemberian program ini ATmega 32 dapat bekerja sesuai dengan fungsinya. Rangkaian dari ATmega 32 ini ditunjukkan pada Gambar 16.

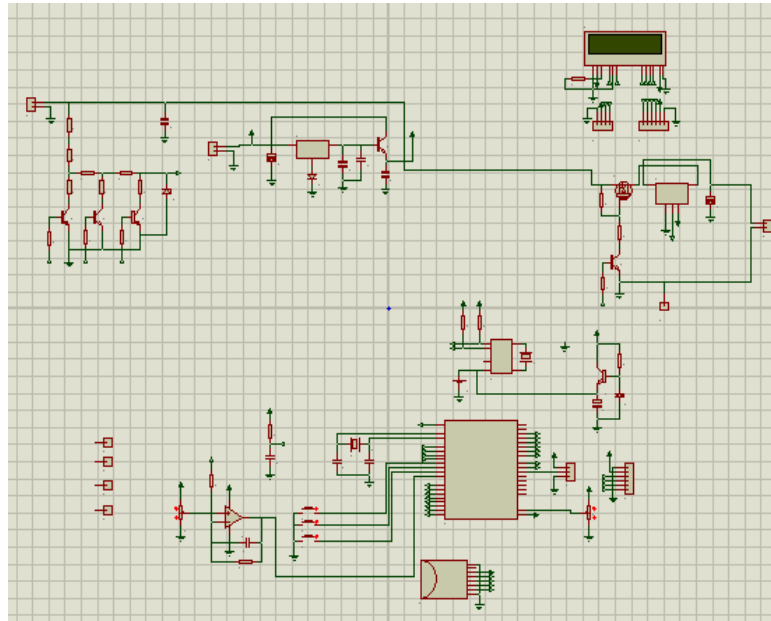


Gambar 16. Rangkaian Atmega 32
(Sumber : <https://www.elprocus.com/>)

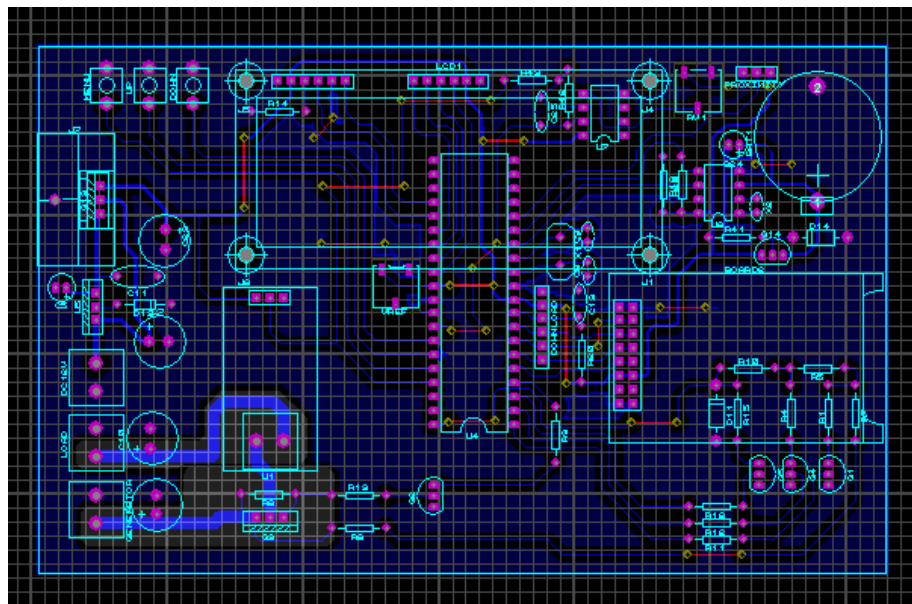
f. Penggambaran Skema Rangkaian Alat Monitoring Data Logger

Skema rangkaian alat monitoring kualitas daya yaitu melakukan perancangan penggambaran rangkaian untuk pembuatan layout pcb. Perancangan Gambar skema rangkaian alat monitoring kualitas daya ini

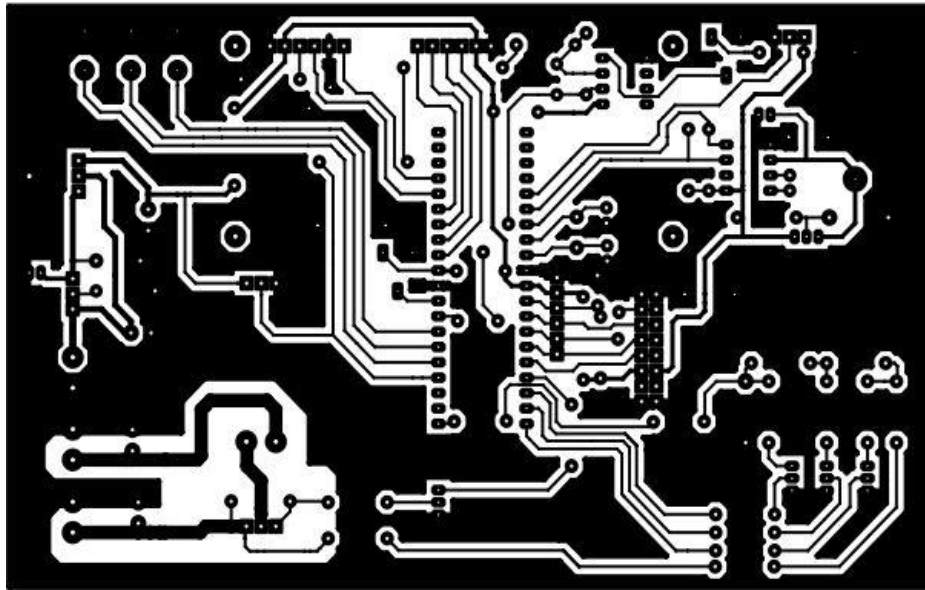
menggunakan aplikasi proteus yang ditunjukkan pada Gambar16 dan untuk layout pcb ditunjukkan pada Gambar17.



Gambar 16. Rangkaian Alat Monitoring Kualitas Daya



Gambar 17. Skematik alat monitoring data



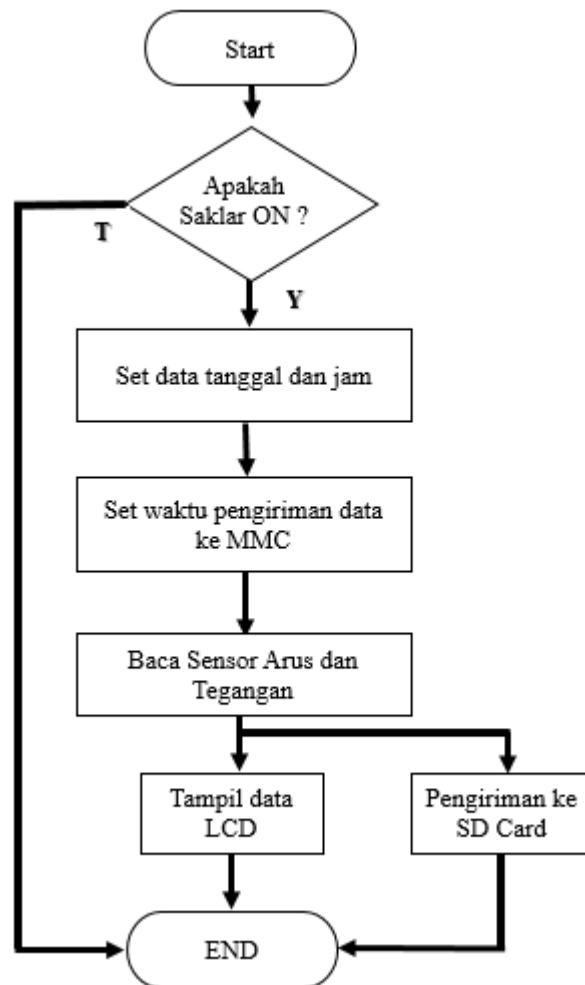
Gambar 18. Layout alat monitoring data

D. Diagram Alir / *Flowchart* Alat

Berikut algoritma dari *flowchart* alat :

1. Saklar pada posisi ON
2. Mengatur tanggal dan jam
3. Mengatur waktu pengiriman data ke MMC
4. Membaca sensor arus dan sensor tegangan
5. Menampilkan data ke LCD
6. Mengirim data ke SD Card
7. Saklar off, END

Alur program secara lengkap dapat dilihat pada Gambar 19.



Gambar 19. Flowchart

Pseudocode :

1. Deklarasi :

Sub rutin program : Unsigned Char

Sensor : Float

Satuan pengukuran : char

Satuan waktu : integer, Unsigned char

Variabel waktu : unsigned int

2. Definisi pin :

Pin A4 = "menu"

pin A5 = “up”

pin A6 =”down”

3. Definisi port :

Port d.5 = output ke RTC

Port d.4 = output ke RTC 2

Port b.3 = input sensor

4. Deskripsi

Setting interrupt → #define T1_OVF_FREQ 100

Setting clock prescaller → #define T1_PRESC 1024

Isi indeks :

Indeks = 0 → tampilan pada LCD

Indeks = 1 → tampilan pada LCD

Indeks = 2 → tampilan pada LCD

Indeks = 3 → set waktu (JAM)

indeks = 4 → set waktu (MENIT)

indeks = 5 → set waktu (DETIK)

indeks = 6 → simpan set waktu (JAM) untuk pengiriman data

indeks = 7 → simpan set waktu (MENIT) untuk pengiriman data

indeks = 8 → simpan set waktu (DETIK) untuk pengiriman data

Pemanggilan fungsi setting waktu :

Void button()

If (indeks==3) then set waktu(JAM)

If (indeks==4) then set waktu(MENIT)

If (indeks==5) then set waktu(DETIK)

If (indeks==6) simpan set waktu (JAM) untuk pengiriman data

If (indeks==7) simpan set waktu (JAM) untuk pengiriman data

If (indeks==8) simpan set waktu (JAM) untuk pengiriman data

Program utama:

Read(sensor) → read_adc(0)*0.17910447761194029850746268656716);

Then ,Hitung :

Beban = Get_Ampere(read_adc(3));

mBeban = Get_mAmpere(read_adc(7));

olah_satuan();

read(RTC) → rtc_get_time(&h,&m,&s);

rtc_get_date(&week,&date,&month,&year);

if(!menu)

{

(isi subrutin indeks(0-8))

Void tombol();

Conversi_waktu();

~ tampil();

}

E. Perancangan Pengujian dan Pengambilan Data

Perancangan sistem dan pengambilan data dimaksudkan untuk mengetahui kinerja alat monitoring data logger. Proses ini digunakan untuk mengetahui

tingkat kesesuaian pengukuran tegangan, arus, daya di alat monitoring data.

Proses pengujian alat dilakukan dalam 3 proses pengujian yaitu :

a. Pengujian tanpa beban

Pengujian tanpa beban digunakan untuk mengetahui kemampuan generator dalam menghasilkan tegangan berdasarkan besarnya putaran dari blade pembangkit listrik tenaga angin. Pengujian ini akan mengetahui tegangan yg dihasilkan dari kincir angin listrik.

b. Pengujian dengan Beban

Pengujian saat berbeban yaitu pengujian dengan menggunakan beban lampu 35 Watt. Pengujian ini dilakukan untuk mengetahui besarnya tegangan saat berbeban, arus yang mengalir dan daya beban.

c. Pengujian sistem penyimpanan data

Pengujian sistem penyimpanan data digunakan untuk mengetahui sistem penyimpanan pada alat monitoring data. Sistem penyimpanan ini akan menyimpan data setiap satu menit sekali. Data yang disimpan adalah parameter tegangan, arus dan daya. Data tersebut disimpan dalam SD-Card dalam format *.txt. sistem akan tetap menyimpan data tergantung besar penyimpanan SD-Card tersebut.

BAB IV

HASIL PENGUJIAN DAN PEMBAHASAN

A. Pengujian

1. Tujuan Pengujian

Pengujian alat Sistem Monitoring Data Kincir Angin Berbasis Mikrokontroler ATmega32 merupakan proses yang dilakukan untuk mengetahui kemampuan dan kondisi alat. Pengujian Sistem Monitoring Data Kincir Angin Berbasis Mikrokontroler ATmega32 meliputi pengujian tanpa beban, pengujian dengan beban dan pengujian sistem penyimpanan data.

2. Tempat Pengujian

Tempat pengujian Proyek Akhir yang berjudul “Sistem Monitoring Data Kincir Angin Berbasis Mikrokontroler ATmega32” ini dilakukan di Pantai Baru Pandansimo. Daerah Pantai Baru Pandansimo terdapat angin laut yang cukup besar, sehingga memudahkan untuk melakukan pengujian alat.

3. Hasil Pengujian

Berikut hasil pengujian yang telah dilakukan:

a. Data hasil pengujian tanpa beban

Pengujian tanpa beban pada alat Sistem Monitoring Data Kincir Angin Berbasis Mikrokontroler ATmega32 ini merupakan pengujian

pada alat tanpa penggunaan beban. Berikut ini adalah langkah-langkah pengujiannya :

- 1) Menghubungkan catu daya/baterai 12 Vdc ke *input power* alat monitoring data kincir angin.
- 2) Menghubungkan *output* catu daya/generator ke *input* alat monitoring data kincir angin.
- 3) Memasang Sd-Card di alat monitoring kualitas daya.
- 4) Mengamati hasil pengukuran parameter yang ditampilkan pada LCD dengan hasil pengukuran ampere meter dan watt meter.
- 5) mencatat hasil pengujian sesuai dengan pengukuran dan penampilan pada alat monitoring data kincir angin.

Setelah dilakukan pengujian didapatkan hasil pengujian tanpa beban yang ditunjukkan oleh Tabel 11.

Tabel 11. Data hasil pengujian tanpa beban

No.	Kecepatan Angin	Output Generator	Monitoring pada alat
			V
1	4,5m/s	20 V	22
2.	4,6 m/s	24 V	23
3.	4,7 m/s	27 V	25
4.	4,8 m/s	28 V	29
5.	4,9 m/s	32 V	34
6.	5,2 m/s	35 V	36
7.	5,3 m/s	37 V	38
8.	5,4 m/s	40 V	41
9.	5,5 m/s	43 V	44
10.	5,6 m/s	45 V	46
11.	5,7 m/s	47 V	48
12.	5,8 m/s	48 V	49
13.	5,9 m/s	48 V	49

b. Data hasil pengujian dengan beban

Pengujian dengan beban pada alat Sistem Monitoring Data Kincir Angin Berbasis Mikrokontroler ATmega32 ini merupakan pengujian pada alat dengan penggunaan beban. Berikut ini adalah langkah-langkah pengujiannya:

- 1) Menghubungkan catu daya/baterai 12 Vdc ke *input power* alat monitoring data kincir angin.
- 2) menghubungkan *output* catu daya/generator ke *input* alat monitoring data kincir angin.
- 3) memasang beban lampu 4 lampu masing-masing 12 volt di *output* alat monitoring data kincir angin.
- 4) memasang Sd-Card di alat monitoring data kincir angin.
- 5) mengamati hasil pengukuran parameter yang ditampilkan pada LCD.

Setelah dilakukan pengujian didapatkan hasil pengujian dengan beban yang ditunjukkan oleh Tabel 12.

Tabel 12. Data hasil pengujian dengan beban

No	Beban (4 Lampu)	Kecepatan Angin	Output Generator	Monitoring Pada Alat		
				V	I	P
1.	12 VDC	4,5 m/s	18 V	19	1	19
2.	12 VDC	4,6 m/s	22 V	23	1,4	32,2
3.	12 VDC	4,7 m/s	25 V	26	1,6	41,6
4.	12 VDC	4,8 m/s	27 V	28	1,7	47,6
5.	12 VDC	4,9 m/s	30 V	32	2	64
6.	12 VDC	5,2 m/s	33 V	34	2,1	71,4
7.	12 VDC	5,3 m/s	36 V	38	2,3	73,6

c. Data hasil pengujian sistem penyimpanan data

Pengujian sistem penyimpanan data merupakan pengujian digunakan untuk mengetahui unjuk kerja sistem penyimpanan data di alat monitoring data kincir angin. Setelah dilakukan pengujian didapatkan hasil yang ditunjukkan pada Tabel 13.

Tabel 13. Data hasil pengujian sistem penyimpanan data

NO	Waktu		Data Alat Monitoring kualitas daya			
	Jam	Tanggal	Tegangan (V)	Arus (I)	Daya (P)	Kecepatan Angin (m/s)
1.	17:01:00	29/07/2016	19	1	19	4,5 m/s
2.	17:02:00	29/07/2016	23	1,4	32,2	4,6 m/s
3.	17:03:00	29/07/2016	26	1,6	41,6	4,7 m/s
4.	17:04:00	29/07/2016	28	1,7	47,6	4,8 m/s
5.	17:05:00	29/07/2016	32	2	64	4,9 m/s
6.	17:06:00	29/07/2016	34	2,1	71,4	5,2 m/s
7.	17:07:00	29/07/2016	38	2,3	73,6	5,3 m/s

B. Pembahasan

1. Perancangan Alat

a. Catu Daya

Catu daya digunakan untuk mensuplay tegangan ke alat monitoring *data logger*. Rangkaian catu daya bekerja dengan tegangan dari batre 12 VDC, direduksi dengan regulator seri 7805, sehingga tegangan outputnya menjadi 5 VDC output tersebut digunakan untuk mensuplay sistem minimum.

b. Sensor Tegangan

Sensor tegangan pada alat monitoring data berfungsi sebagai input data tegangan. Sensor ini berfungsi menurunkan tegangan, dari tegangan yang diukur menjadi tegangan yang diijinkan pada mikrokontroler.

c. Sensor Arus

Sensor arus menggunakan komponen sensor ACS712. ACS712 memperoleh masukan arus dan melakukan pembacaan dari arus yang masuk, arus tersebut mengeluarkan ADC yang dibaca oleh mikrokontroler. Kemudian data disimpan dalam memori melalui modul, data memori tersebut dapat dilihat dengan PC data berupa txt

2. Pembuatan Alat

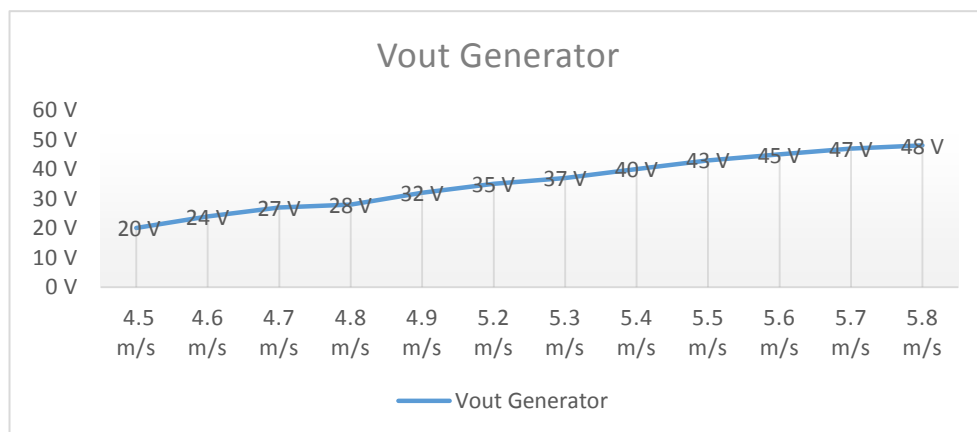
Pembuatan sistem monitoring data kincir angin berbasis mikrokontroler ATmega32 di pantai baru pandansimo melalui beberapa tahapan, yang pertama yaitu pembuatan mekanik yang meliputi

perencanaan ukuran mekanik dan perencanaan mekanik. Dan yang kedua pembuatan PCB yang meliputi pembuatan rangkaian, pembuatan layout PCB, penyablonan PCB, pelarutan PCB, pengeboran PCB, pemasangan komponen dan uji coba.

3. Unjuk Kerja

a. Pengujian Tanpa Beban

Berdasarkan hasil pengujian pada Tabel 11, saat dilakukan pengujian tanpa beban data tidak tersimpan pada *data logger*, karena *data logger* hanya untuk menyimpan data tegangan, arus dan daya saat berbeban. Grafik peningkatan tegangan output generator berbanding lurus dengan kecepatan angin ditunjukkan pada Gambar 20.

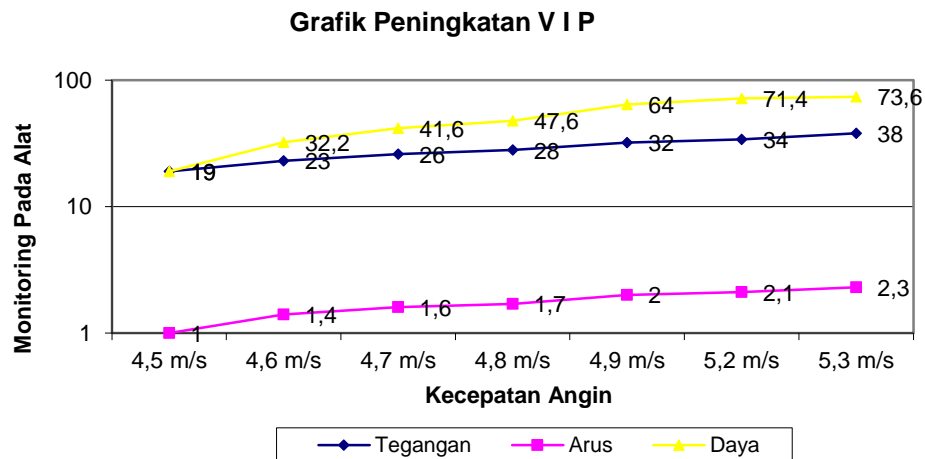


Gambar 20. Grafik peningkatan tegangan output generator

b. Pengujian Berbeban

Berdasarkan data hasil pengujian pada tabel 12 yang kemudian disajikan dalam bentuk grafik yang ditunjukkan pada gambar 21, apabila kecepatan angin meningkat maka tegangan output dan arus

listrik generator akan semakin besar, yang artinya daya yang dihasilkan akan semakin besar.



Gambar 21. Grafik pengaruh kecepatan angin terhadap tegangan output dan arus generator pada kondisi berbeban.

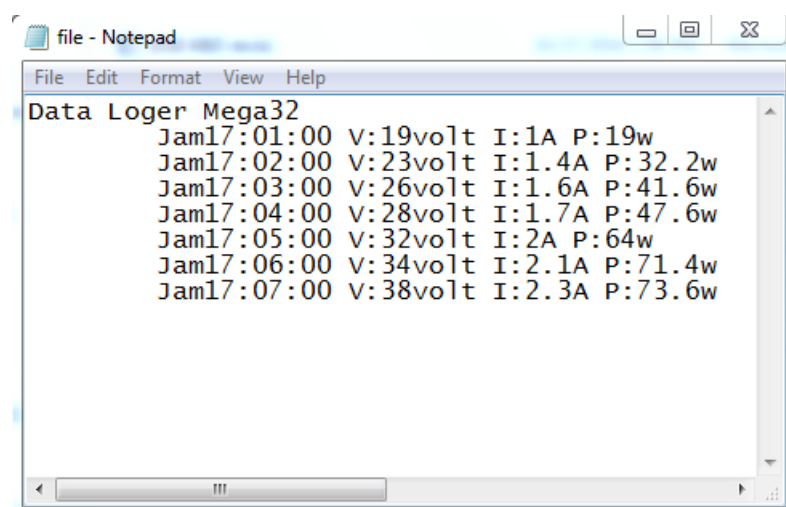
Pada pengujian berbeban menggunakan beban sebuah lampu dengan spesifikasi beban lampu 144watt berdasarkan pengujian diatas di dapatkan maksimal tegangan keluaran dari generator sebesar 48 V dengan beban lampu tersebut, besarnya arus yang mengalir hal itu tergantung besarnya tegangan yang keluar dari generator, semakin besar tegangan yang mengalir dengan beban yang sama maka arus yang dihasilkan akan semakin besar.

c. Pengujian Penyimpanan Data

Sistem penyimpanan data di alat monitoring kualitas daya bekerja menyimpan data berupa tegangan arus dan daya yang dapat menyimpan secara otomatis sesuai dengan waktu yang ditentukan. Tegangan arus dan daya ini akan disimpan dalam suatu chip memori. Parameter-parameter yang telah ditentukan dalam selang waktu satu menit. Berdasarkan data hasil pengujian sistem penyimpanan yang

ditunjukkan pada tabel 13, data disimpan pada jam 17:01:00, kemudian setelah satu menit menyimpan lagi pada jam 17:02:00. sistem penyimpanan data di alat monitoring ini menggunakan sd-card 8 GB. Sd-Card ini yang digunakan media untuk menyimpan data dalam format datalog.txt.

Berdasarkan pengujian yang telah dilakukan maka didapat tegangan manimum pada penyimpanan *data logger* ialah sebesar 19 volt yang tercatat pada jam 17:01:00, pada jam yang sama penyimpanan juga menyimpan arus minimum ialah sebesar 1 Ampere, serta daya minimum yang tersimpan ialah sebesar 19 watt. Dan pada jam 17:07:00 penyimpanan *data logger* menyimpan data tegangan maksimal sebesar 38 volt, data arus maksimal ialah 2,3 ampere, serta data daya maksimum ialah sebesar 73,6 watt. Data-data tersebut dapat menjadi acuan jika kedepannya diterapkan pada kehidupan sehari-hari. Sehingga dapat disesuaikan dengan yang dibutuhkan.



```
file - Notepad
File Edit Format View Help
Data Loger Mega32
Jam17:01:00 V:19volt I:1A P:19w
Jam17:02:00 V:23volt I:1.4A P:32.2w
Jam17:03:00 V:26volt I:1.6A P:41.6w
Jam17:04:00 V:28volt I:1.7A P:47.6w
Jam17:05:00 V:32volt I:2A P:64w
Jam17:06:00 V:34volt I:2.1A P:71.4w
Jam17:07:00 V:38volt I:2.3A P:73.6w
```

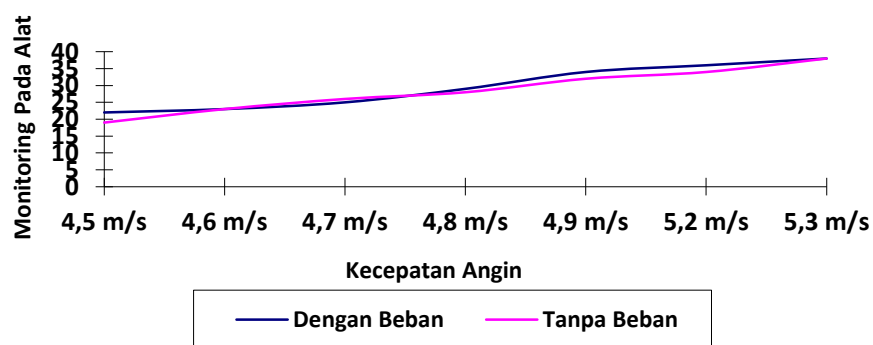
Gambar 22. File DataLogger pada SD-Card

Berdasarkan pengujian sistem penyimpanan data di alat monitoring kualitas daya bekerja dengan baik sesuai fungsinya, proses penyimpanan ini berlangsung setiap satu menit sekali dan menyimpan data berupa tegangan, arus dan daya yang disimpan di Sd-Card dalam file name datalog seperti yang ditunjukkan pada Gambar 22.

d. Perbandingan Dengan Beban dan Tanpa Beban

Berdasarkan grafik di bawah semakin besar kecepatan angin pada saat blade memutar akan menghasilkan tegangan yang semakin besar juga, besar kecepatan angin berbanding lurus dengan besar tegangan output. Output tegangan pada saat tanpa beban lebih besar dari pada output tegangan berbeban, dikarenakan jika dengan beban tegangan akan turun sebab ada hambatan pada beban tersebut.

Grafik Perbandingan V dengan beban dan tanpa beban



Gambar 23. Grafik perbandingan

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Setelah melakukan pengamatan dan pembahasan pada Proyek Akhir yang berjudul Sistem Monitoring Data Kincir Angin Berbasis Mikrokontroler ATmega32, dapat diambil kesimpulan yaitu:

1. Sistem monitoring data kincir angin berbasis mikrokontroler ATmega32 di pantai baru pandansimo adalah sebuah sistem monitoring untuk memantau Tegangan, Arus, dan Daya yang dihasilkan oleh kincir angin dengan memanfaatkan angin yang ada di pantai baru pandansimo.
2. Pembuatan sistem monitoring data kincir angin berbasis mikrokontroler ATmega32 di pantai baru pandansimo, yang terdiri dari : (1) perancangan hardware yaitu pembuatan box sebagai penyimpanan mikrokontroler, (2) Setup progam yaitu untuk memasukkan pin yang diinginkan sebagai input dan output
3. Unjuk kerja alat dengan pengujian Arus, Tegangan, Daya yang ditampilkan pada LCD yang dilakukan dengan pengukuran-pengukuran pada bagian-bagian tertentu sudah selesai dengan yang diharapkan.

B. Keterbatasan Alat

Pembuatan Sistem Monitoring Data Kincir Angin Berbasis Mikrokontroler ATmega32 memiliki keterbatasan yaitu:

1. Sistem pembacaan arus kurang stabil saat kecepatan angin melebihi 10 m/s.

2. Penyimpanan pada Sd Card terkadang tidak tersimpan, dikarenakan perubahan pengaturan pengiriman data.

C. Saran

Dalam pembuatan proyek akhir ini tentunya terdapat kekurangan-kekurangan, sehingga diperlukan pengembangan lebih lanjut. Saran yang membangun dibutuhkan untuk menyempurnakan proyek akhir ini, antara lain sebagai berikut :

1. Menggunakan sensor arus yang karakteristiknya lebih stabil pada pembacaan angin lebih dari 10 m/s.
2. Perlu adanya pengaturan waktu saat pengiriman data yang tepat agar tidak merubah pengaturan awalnya.

DAFTAR PUSTAKA

- Alpen. (2016). Penggunaan listrik tenaga angin. Diakses pada tanggal 20 agustus 2016. Dari <http://www.alpensteel.com/article/47-103-energi-angin--wind-turbine--wind-mill/2979--penggunaan-listrik-tenaga-angin.html>
- Dickson Kho. (2016). Pengertian Daya listrik dan Rumus untuk Menghitungnya. Diakses pada 12 September 2016. Dari <http://teknikelektronika.com/pengertian-daya-listrik-rumus-cara-menghitung/>
- Dita anggraini. (2016). Analisis potensi angin di pantai baru pandansimo. Diakses pada 3 September 2016. Dari <http://ditaanggraini.web.ugm.ac.id/2016/04/09/analisis-potensi-angin-di-pantai-baru-pandansimo-kabupaten-bantul/>
- Eko Purnomo. (2015). Mikrokontroler Avr Atmega32. Diakses pada tanggal 12 Agustus 2016 dari <http://www.nulis-ilmu.com/2015/09/mikrokontroler-avr-atmega32.html>
- Hari Santoso. Konsep dan pengertian arus dan tegangan pada rangkaian listrik. Diakses pada tanggal 9 Agustus 2016, dari <http://www.elangsakti.com/2013/03/konsep-dan-pengertian-arus-dan-tegangan.html>

LAMPIRAN

Lampiran 1. Foto Uji Kerja Kincir Angin



Lampiran 3. Program Alat

```

#include <mega32a.h>
#include <stdio.h>
#include <delay.h>
// I2C Bus functions
#include <i2c.h>
// DS1307 Real Time Clock functions
#include <ds1307.h>
#include <sdcard.h>
#include <ff.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include <spi.h>
#include <math.h>
// Alphanumeric LCD functions
#include <alcd.h>
#define out_time PORTD.5
#define out2 PORTD.4
#define menu PINA.4
#define up PINA.5
#define down PINA.6
#define load PORTB.3
unsigned int sec=0,msec=0,volt=0;
unsigned char week,date,month,year,h,m,s,indeks=0,indeks_compare=0;
int jam,menit,detik,jam1,menit1,detik1;
char bawah[33],sat_arus=0,sat_daya=0;
float arus,daya_tot;
char *bil_arus[3]={"","mA","A"};
char *satunya_daya[3]={"","mW","W"};
eeprom unsigned int dat_jam,dat_mnt,dat_dtk,set_range;
int i,j=0,temp3=0,tanda_arus=0;
unsigned int temp,temp1=0,temp2,array_arus=0;
/* Timer1 overflow interrupt frequency [Hz] */
#define T1_OVF_FREQ 100
/* Timer1 clock prescaler value */
#define T1_PRESC 1024L
/* Timer1 initialization value after overflow */
#define T1_INIT (0x1000L-(MCU_CLOCK_FREQUENCY/(T1_PRESC*T1_OVF_FREQ)))
#define TIMER_FREQ 100 /* Timer increment frequency [Hz] */
// #define T1M2_INIT MCU_CLOCK_FREQUENCY/(1024L*TIMER_FREQ)-1
#define T1M2_INIT (0x100L-(MCU_CLOCK_FREQUENCY/(1024L*TIMER_FREQ)))
/* FAT function result */
FRESULT res;
/* number of bytes written/read to the file */
unsigned int nbytes;
/* will hold the information for logical drive 0: */
FATFS fat;
/* will hold the file information */
FIL file;
/* will hold file attributes, time stamp information */
//FILINFO finfo;
/* will hold file attributes, time stamp information */
FILINFO finfo;

```

```

#define ADC_VREF_TYPE 0x00
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=0x40;
    // Wait for the AD conversion to complete
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCW;
}
// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
    // Place your code here
}
int Arus_indeks=0,mArus_indeks=0;
int sec_Arus,sec_mArus;
// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
    // Place your code here
    TCNT1H=T1_INIT>>8;
    TCNT1L=T1_INIT&0xFF;
    // card access low level timing function
    //disk_timerproc();
    if(++msec>=100)
        {if(++sec>=60){sec=0;}
        if(Arus_indeks==0){sec_Arus=sec_Arus + 1;if(sec_Arus>=60){sec_Arus=0;Arus_indeks=1;}}
        if(mArus_indeks==0){sec_mArus=sec_mArus +
1;if(sec_mArus>=60){sec_mArus=0;mArus_indeks=1;}}
        msec=0;
        }
}
interrupt [TIM2_OVF] void timer_comp_isr(void)
{TCNT2 = TIM2_INIT;
disk_timerproc(); /* Drive timer procedure of low level disk I/O module */
}
void start_timer_mmc()
{
    TCCR2 = (1<<CS22)+(1<<CS21)+(1<<CS20);
    //TCCR2=0x00;
    TCNT2=TIM2_INIT;
    TIMSK= (1 << TOIE2) | (1 << TOIE1);
}
void stop_timer_mmc()
{
    TCCR2 = (0<<CS22)+(0<<CS21)+(0<<CS20);
    //TCCR2=0x00;
    TCNT2=TIM2_INIT;
    TIMSK= (1 << TOIE2) | (1 << TOIE1);
}

```

```

/* error message list */
flash char * flash error_msg[]=
{
    "", /* not used */
    "FR_DISK_ERR",
    "FR_INT_ERR",
    "FR_NOT_READY",
    "FR_NO_FILE",
    "FR_NO_PATH",
    "FR_INVALID_NAME",
    "FR_DENIED",
    "FR_EXIST",
    "FR_INVALID_OBJECT",
    "FR_WRITE_PROTECTED",
    "FR_INVALID_DRIVE",
    "FR_NOT_ENABLED",
    "FR_NO_FILESYSTEM",
    "FR_MKFS_ABORTED",
    "FR_TIMEOUT"
};
/* display error message and stop */
void error(FRESULT res)
{
    if ((res>=FR_DISK_ERR) && (res<=FR_TIMEOUT))
    {
        lcd_gotoxy(0,0);
        sprintf(bawah,"ERROR: %p",error_msg[res]);
        lcd_puts(bawah);
    }
    /* stop here */
    //while(1);
}
/* file path */
char path[]="0:/file.txt";
/* text to be written to the file */
char text2[]="Data Loger Mega32\r\n";
char text3[55];
/* file read buffer */
char buffer[55];
unsigned char status;
void file_name_txt(char *file_name)
{
    if ((res=f_stat(file_name,&finfo))==FR_OK)
    {
        lcd_gotoxy(0,0);sprintf(bawah,"File %s opened",file_name);
        lcd_puts(bawah);
    }
}
void file_baru()
{
    start_timer_mmc();
    if ((res=f_open(&file,path,FA_CREATE_ALWAYS | FA_WRITE))==FR_OK)
    {
        lcd_gotoxy(0,0);sprintf(bawah,"\r\nDisk -> create file %s created OK",path);lcd_puts(bawah);}
    else {lcd_clear();error(res);}
    //if ((res=f_write(&file,text3,sizeof(text3)-1,&nbytes))==FR_OK)
    // {sprintf(display_buffer,"%u bytes written of %u\r\n",nbytes,sizeof(text3)-
    1);lcd_puts(display_buffer);}
}

```

```

// else {error(res);}
if ((res=f_close(&file))==FR_OK)
    {lcd_gotoxy(0,0);
    sprintf(bawah,"File %s closed OK",path);
    lcd_puts(bawah);
    }
    else {error(res);}
stop_timer_mmc();
}
void isi_file()
{start_timer_mmc();
if ((res=f_open(&file,path,FA_WRITE))==FR_OK)
    {lcd_gotoxy(0,0);sprintf(bawah,"File %s opened OK\r\n",path);lcd_puts(bawah);}
else {lcd_clear();error(res);}
/* seek to the end of the file for appending */
if ((res=f_lseek(&file,file.fsize))==FR_OK)
    {lcd_gotoxy(0,0);sprintf(bawah,"Seek to the end of the file OK",path);lcd_puts(bawah);}
    else {error(res);}
if ((res=f_write(&file,text3,sizeof(text3)-1,&nbytes))==FR_OK)
    {lcd_gotoxy(0,0);
    sprintf(bawah,"%u bytes appended of %u",nbytes,sizeof(text3)-1);
    lcd_puts(bawah);
    }
    else {error(res);}
/* close the file */
if ((res=f_close(&file))==FR_OK)
    {lcd_gotoxy(0,0);sprintf(bawah,"File %s closed OK",path);lcd_puts(bawah);}
    else {lcd_clear();error(res);}
delay_ms(200);
stop_timer_mmc();
}
void cek_sdcard() //fungsi utk cek koneksi dg sdcard
{start_timer_mmc();
lcd_gotoxy(0,0);lcd_putsf("SD card driver test\r\nInitializing drive...\r\n");
status=disk_initialize(0);
/* display disk initialization result on the LCD */
if (status & STA_NODISK){lcd_gotoxy(0,0);
    lcd_putsf("Card not present\r\n");
}else if (status & STA_NOINIT){lcd_gotoxy(0,0);
    lcd_putsf("Disk init failed\r\n");
    //error(res);
}else if (status & STA_PROTECT){lcd_gotoxy(0,0);
    lcd_putsf("Card write\nprotected\r\n");
/* all status flags are 0, disk initialization OK */
}else
    {if ((res=f_mount(0,&fat))==FR_OK)
        {lcd_gotoxy(0,0);
        sprintf(bawah,"Logical drive 0: mounted OK");
        lcd_puts(bawah);
        delay_ms(2);
        lcd_clear();
        }
        else {lcd_clear();error(res);}
    }
}
//strcat(text3,text2);

```

```

strcpy(text3,text2);
//strcpy(text3,mid(text2,0,33));
delay_ms(100);
//f_unlink(path);
//stop_timer_mmc();
file_baru();
isi_file();
}
int Amp_Maopi,mAmp_Maopi,Amp_Shiwu=0,mAmp_Shiwu=0;
int jam_temp,menit_temp,detik_temp;
unsigned int Beban,mBeban;
//dat_jam,dat_mnt,dat_dtk//
void conversi_waktu()
{
detik_temp = detik + dat_dtk;
if(detik_temp>59)
{detik_temp = detik_temp - 60;
menit_temp = 1 + menit + dat_mnt;
}else{menit_temp = menit + dat_mnt;}
if(menit_temp>59)
{menit_temp = menit_temp - 60;
jam_temp = 1 + jam + dat_jam;
}else{jam_temp = jam + dat_jam;}
if(jam_temp>23)
{jam_temp = jam_temp - 24;
}
}
void sdcard_kirim()
{lcd_gotoxy(0,0);
lcd_putsf("Kirim SDCARD");
delay_ms(1000);
sprintf(buffer,"Jam%d:%d:%d V:%dvolt I:%.1f%s
P:%.1f%s\r\n",jam,menit,detik,volt,arus,bil_arus[sat_arus],daya_tot,satuan_daya[sat_daya]);
strcpy(text3,buffer);
delay_ms(100);
sprintf(buffer,"%s ",text3);
lcd_gotoxy(0,0);
lcd_puts(buffer);
delay_ms(4000);
isi_file();
delay_ms(4000);
if(indeks_compare==1){conversi_waktu();lcd_clear();indeks_compare=0;}
}
unsigned int Get_Ampere(int AMP);
unsigned int Get_mAmpere(int mAmp);
int indeks_adc=0,btas=25;
int arus_adc[26];
void olah_adc(int dat_dig)
{if(dat_dig>0)
{if(indeks_adc==0)
{if(temp>dat_dig && array_arus>=0 && array_arus<=btas && tanda_arus==0)
{arus_adc[array_arus]=dat_dig;
array_arus=array_arus + 1;
temp=0;indeks_adc=1;
}
}
}
}

```

```

    }
    if(indeks_adc==1)
    { if(temp<dat_dig)
      {indeks_adc=0;}
    }
    temp=dat_dig;sec=0;
  }
  if(array_arus>btas && tanda_arus==0)
  {for(i=0;i<=array_arus;i++)
    {if(temp2<arus_adc[i]){temp2=arus_adc[i];i=j;}
    }
    for(i=0;i<=array_arus;i++)
    {if(i!=j && temp1<arus_adc[i]){temp1=arus_adc[i];}
    }
    tanda_arus=1;
  }
  if(tanda_arus==1)
  {for(i=0;i<=array_arus;i++)
    {if(arus_adc[i]>0){arus_adc[i]=0;}
    }
    temp3=(unsigned int)((float)(temp2+temp1)/2);
    array_arus=0;temp2=0;temp1=0;tanda_arus=0;
  }
  if((dat_dig<=temp3 || dat_dig==0) && sec>2)
  {temp=0;temp1=0;temp2=0;temp3=0;tanda_arus=0;
  array_arus=0;sec=0;
  }
}
void compare_waktu()
{
if(Arus_indeks==1 && mArus_indeks==1)
{load=1;
if(jam>=jam_temp && menit>=menit_temp && detik>=detik_temp)
{indeks_compare=1;
sdcard_kirim();
}
}
}
void Get_Maopi(int Dat_A, int Dat_mA)
{
Amp_Maopi = Dat_A;
mAmp_Maopi = Dat_mA;
}
//interval=3

unsigned int Get_Ampere(int AMP)
{
Amp_Shiwu = AMP - Amp_Maopi;
if(Amp_Shiwu<0){Amp_Shiwu=0;}
if(Amp_Shiwu!=0 && Arus_indeks==0){Amp_Maopi=AMP;sec_Arus=0;}
Amp_Shiwu = (unsigned int)((float)Amp_Shiwu);
return Amp_Shiwu;
}
//interval=30
unsigned int Get_mAmpere(int mAMP)

```

```

{
mAmp_Shiwu = mAMP - mAmp_Maopi;
if(mAmp_Shiwu<0){ mAmp_Shiwu=0;}
if(mAmp_Shiwu!=0 && mArus_indeks==0){mAmp_Maopi=mAMP;sec_mArus=0;}
mAmp_Shiwu = (unsigned int)((float)mAmp_Shiwu);
return mAmp_Shiwu;
}

void tampil();
void display();
// arus = temp3 * 0,53648648648648648648648648648649;//
void olah_satuan()
{if(read_adc(7)<742)
  {olah_adc(mBeban);
  arus = ((float)temp3 * 0.55696202531645569620253164556962);
  sat_arus=1;
  }
if(read_adc(7)>=742)
  {olah_adc(Beban);arus = temp3 * 0.00416666666666666666666666666667;sat_arus=2;}
if((unsigned int)(arus * volt)<1000){daya_tot = arus * volt;sat_daya=1;}
if((unsigned int)(arus * volt)>=1000){daya_tot = (float)((arus * volt)/1000);sat_daya=2;}
}
void buttom()
{
if(indeks==3)
{
if(!up)
  {lcd_clear();
  if(++jam>=24){jam=0;}
  rtc_set_time(jam,menit,detik);
  delay_us(50);
  if(!up){ while(!up){display();};}
  lcd_clear();
  }
if(!down)
  {lcd_clear();
  if(--jam<0){jam=23;}
  rtc_set_time(jam,menit,detik);
  delay_us(50);
  if(!down){ while(!down){display();};}
  lcd_clear();
  }
}
if(indeks==4)
{
if(!up)
  {lcd_clear();
  if(++menit>=60){menit=0;}
  rtc_set_time(jam,menit,detik);
  delay_us(50);
  if(!up){ while(!up){display();};}
  lcd_clear();
  }
if(!down)
  {lcd_clear();

```

```

    if(--menit<0){menit=59;}
    rtc_set_time(jam,menit,detik);
    delay_us(50);
    if(!down){while(!down){display();}}
    lcd_clear();
  }
}
if(indeks==5)
{
  if(!up)
  {lcd_clear();
  detik=0;
  rtc_set_time(jam,menit,detik);
  delay_us(50);
  if(!up){while(!up){display();}}
  lcd_clear();
  }
  if(!down)
  {lcd_clear();
  detik=0;
  rtc_set_time(jam,menit,detik);
  delay_us(50);
  if(!down){while(!down){display();}}
  lcd_clear();
  }
}

if(indeks==6)
{
  if(!up)
  {lcd_clear();
  if(++jam1>=24){jam1=0;}
  dat_jam=jam1;
  delay_us(50);
  if(!up){while(!up){display();}}
  lcd_clear();
  }
  if(!down)
  {lcd_clear();
  if(--jam1<0){jam1=23;}
  dat_jam=jam1;
  delay_us(50);
  if(!down){while(!down){display();}}
  lcd_clear();
  }
}
if(indeks==7)
{
  if(!up)
  {lcd_clear();
  if(++menit1>=60){menit1=0;}
  dat_mnt=menit1;
  delay_us(50);
  if(!up){while(!up){display();}}
  lcd_clear();
  }
}

```

```

    }
    if(!down)
    {lcd_clear();
    if(--menit1<0){menit1=59;}
    dat_mnt=menit1;
    delay_us(50);
    if(!down){while(!down){display();}}
    lcd_clear();
    }
}
if(indeks==8)
{
if(!up)
{lcd_clear();
if(++detik1>=60){detik1=0;}
dat_dtk=detik1;
delay_us(50);
if(!up){while(!up){display();}}
lcd_clear();
}
if(!down)
{lcd_clear();
if(--detik1<0){detik1=59;}
dat_dtk=detik1;
delay_us(50);
if(!down){while(!down){display();}}
lcd_clear();
}
}

}

void display()
{
if(indeks==3)
{lcd_gotoxy(0,0);
lcd_puts("Set Waktu Jam");
sprintf(bawah,"%d %d %d ",jam=h,menit=m,detik=s);
lcd_gotoxy(0,1);
lcd_puts(bawah);
}
if(indeks==4)
{lcd_gotoxy(0,0);
lcd_putsf("Set Waktu Menit");
sprintf(bawah,"%d %d %d ",jam=h,menit=m,detik=s);
lcd_gotoxy(0,1);
lcd_puts(bawah);
}
if(indeks==5)
{lcd_gotoxy(0,0);
lcd_putsf("Set Waktu Detik");
sprintf(bawah,"%d %d %d ",jam=h,menit=m,detik=s);
lcd_gotoxy(0,1);
lcd_puts(bawah);
}
}

```

```

if(indeks==6)
{lcd_gotoxy(0,0);
lcd_putsf("Set intval Jam");
sprintf(bawah,"%d %d %d ",jam1,menit1,detik1);
lcd_gotoxy(0,1);
lcd_puts(bawah);
}
if(indeks==7)
{lcd_gotoxy(0,0);
lcd_putsf("Set intval Menit");
sprintf(bawah,"%d %d %d ",jam1,menit1,detik1);
lcd_gotoxy(0,1);
lcd_puts(bawah);
}
if(indeks==8)
{lcd_gotoxy(0,0);
lcd_putsf("Set intval Detik");
lcd_gotoxy(0,1);
sprintf(bawah,"%d %d %d ",jam1,menit1,detik1);
lcd_puts(bawah);
}

}

void tampil()
{ jam=h;menit=m;detik=s;
jam1=dat_jam;menit1=dat_mnt;detik1=dat_dtk;

if(Arus_indeks==0 || mArus_indeks==0)
{
if(indeks==0)
{
lcd_gotoxy(0,0);
sprintf(bawah,"%d %d %d %d %d ",Amp_Maopi,mAmp_Maopi,sec_Arus,sec_mArus,temp3);
//Weight_Maopi);//
lcd_puts(bawah);
lcd_gotoxy(0,1);
sprintf(bawah,"%d %d %d %d ",Amp_Shivu,mAmp_Shivu,read_adc(3),read_adc(7));
//HX711_Buffer);//
lcd_puts(bawah);
}
}

if(Arus_indeks==1 && mArus_indeks==1)
{
if(indeks==0)
{
lcd_gotoxy(0,0);
sprintf(bawah,"V=%d P=%.1f%s ",volt,daya_tot,satuan_daya[sat_daya]); //Weight_Maopi);//
lcd_puts(bawah);
lcd_gotoxy(0,1);
sprintf(bawah,"% .1f%s %d:%d:%d ",arus,bil_arus[sat_arus],jam,menit,detik); //HX711_Buffer);//
lcd_puts(bawah);
}
}
}

```

```

if(indeks==1)
{
lcd_gotoxy(0,0);
sprintf(bawah,"%d %d %d %d %d ",Amp_Maopi,mAmp_Maopi,sec_Arus,sec_mArus,temp3);
//Weight_Maopi);//
lcd_puts(bawah);
lcd_gotoxy(0,1);
sprintf(bawah,"%d %d %d %d ",Amp_Shiwu,mAmp_Shiwu,read_adc(3),read_adc(7));
//HX711_Buffer);//
lcd_puts(bawah);
}
if(indeks==2)
{
sprintf(bawah,"%d %d %d %d ",read_adc(3),Beban,volt,read_adc(0)); //Weight_Maopi);//
sprintf(buffer,"%d %d %d ",detik=s,read_adc(7),mBeban); //HX711_Buffer);//
lcd_gotoxy(0,0);
lcd_puts(bawah);
lcd_gotoxy(0,1);
lcd_puts(buffer);

}
if(indeks==3)
{ display();
  buttom();

}
if(indeks==4)
{ display();
  buttom();

}
if(indeks==5)
{ display();
  buttom();

}
if(indeks==6)
{ display();
  buttom();

}
if(indeks==7)
{ display();
  buttom();

}
if(indeks==8)
{ display();
  buttom();

}

}

void main(void)

```

```

{
// Declare your local variables here
load=0;
// 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=0x70;
DDRA=0x00;

// Port B initialization
// Func7=Out Func6=In Func5=Out Func4=Out Func3=In Func2=In Func1=In Func0=In
// State7=0 State6=T State5=0 State4=0 State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0xB8;

// 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=0xF0;

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

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 16000,000 kHz
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: On
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
//TCCR1A=0x00;
//TCCR1B=0x01;
//TCNT1H=0x00;
//TCNT1L=0x00;
/* initialize Timer1 overflow interrupts in Mode 0 (Normal) */

```

```

TCCR1A=0x00;
/* clkio/1024 */
TCCR1B=(1<<CS12)|(1<<CS10);
/* timer overflow interrupts will occur with 100Hz frequency */
TCNT1H=T1_INIT>>8;
TCNT1L=T1_INIT&0xFF;
/* enable Timer1 overflow interrupt */

ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
//TCCR2 = (1<<CS22)+(1<<CS21)+(1<<CS20);
TCCR2=0x00;
TCNT2=0x00;
//TCNT2=T1M2_INIT;
//OCR2 = T1M2_INIT;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Falling Edge
// INT1: Off
// INT2: Off
GICR|=0x40;
MCUCR=0x02;
MCUCSR=0x00;
GIFR=0x40;

// Timer(s)/Counter(s) Interrupt(s) initialization
//TIMSK= (1 << TOIE2) | (1 << TOIE1);
TIMSK=0x00;

// USART initialization
// USART disabled
UCSRB=0x00;

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

// ADC initialization
// ADC Clock frequency: 1000,000 kHz
// ADC Voltage Reference: AREF pin

```

```

ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

// SPI initialization
// SPI Type: Master
// SPI Clock Rate: 4000,000 kHz
// SPI Clock Phase: Cycle Start
// SPI Clock Polarity: Low
// SPI Data Order: MSB First
SPCR=0x50;
SPSR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// I2C Bus initialization
// I2C Port: PORTD
// I2C SDA bit: 1
// I2C SCL bit: 0
// Bit Rate: 100 kHz
// Note: I2C settings are specified in the
// Project|Configure|C Compiler|Libraries|I2C menu.
i2c_init();

// DS1307 Real Time Clock initialization
// Square wave output on pin SQW/OUT: Off
// SQW/OUT pin state: 1
rtc_init(0,0,1);

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 3
// RD - PORTC Bit 1
// EN - PORTC Bit 2
// D4 - PORTC Bit 4
// D5 - PORTC Bit 5
// D6 - PORTC Bit 6
// D7 - PORTC Bit 7
// Characters/line: 16
lcd_init(16);

// Global enable interrupts
#asm("sei")
//sdcard_init();
lcd_gotoxy(0,0);
lcd_putsf("Dataloger");
lcd_gotoxy(0,1);
lcd_putsf("  Digital");
delay_ms(1000);
lcd_clear();
if(!menu)
{rtc_set_time(14,10,20);rtc_set_date(2,10,4,16);}
cek_sdcard();

```

```

delay_ms(2000);
temp=0;temp1=0;
temp=0;temp1=0;
for(i=0;i<10;i++){Get_Maopi(read_adc(3),read_adc(7));delay_ms(100);}
rtc_get_time(&h,&m,&s);
jam=h,menit=m,detik=s;
conversi_waktu();
lcd_clear();
delay_ms(1000);
//load=1;

while (1)
{
if(indeks>=0 && indeks<=2)
{
volt=((float)read_adc(0)*0.17910447761194029850746268656716);
Beban = Get_Ampere(read_adc(3));
mBeban = Get_mAmpere(read_adc(7));
olah_satuan();
}
if(indeks>=0 && indeks<=8 && indeks_compare==0)
{
rtc_get_time(&h,&m,&s);
rtc_get_date(&week,&date,&month,&year);
tampil();
}
if(!menu)
{lcd_clear();
if(++indeks>=9)
{conversi_waktu();
indeks=0;
delay_us(50);
}
delay_us(50);
if(!menu)
{ while(!menu){tampil();} }
lcd_clear();
};
if(!up && indeks==0)
{lcd_clear();Amp_Maopi=0;mAmp_Maopi=0;
Arus_indeks=0;mArus_indeks=0;load=0;
delay_us(50);
while(!up && indeks==0)
{tampil();
}
lcd_clear();
};
compare_waktu();
// delay_ms(100);
// lcd_clear();
}
}

```