TUGAS AKHIR – TF 145565

RANCANG BANGUN SISTEM MONITORING KECEPATAN KAKI MANUSIA SAAT BERLARI DI ATAS TREADMILL BERBASIS MIKROKONTROLLER

RIZA FARIDA
NRP. 2413 031 028

Dosen Pembimbing :
Ir. Tutug Dhanardono, MT

PROGRAM STUDI D3 METROLOGI DAN INSTRUMENTASI
JURUSAN TEKNIK FISIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016

FINAL PROJECT – TF145565

DESIGN OF SYSTEM MONITORING SPEED OF THE HUMAN FOOT WHEN RUNNING ON A TREADMILL BASED MICROCONTROLLER

RIZA FARIDA
NRP. 2413 031 028

Counselor Lecturer
Ir. Tutug Dhanardono, MT

DIPLOMA OF METROLOGY AND INSTRUMENTATION
DEPARTMENT OF ENGINEERING PHYSICS
FACULTY OF INDUSTRIAL TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2016

RANCANG BANGUN SISTEM MONITORING KECEPATAN KAKI MANUSIA SAAT BERLARI DIATAS TREADMILL BERBASIS MIKROKONTROLLER

TUGAS AKHIR

Oleh :
RIZA FARIDA
NRP. 2413 031 028

Surabaya, 28 Juli 2016
Mengetahui / Menyetujui

Dosen Pembimbing

Ir. Tutug Dhanardono, MT
NIP.19520613 198103 1 004

Ketua Jurusan
Teknik Fisika FTI – ITS

Agus Muhamad Hatta, ST, MSi, Ph.D
NIP. 19780902 200312 1 002

Ketua Program Studi
D3 Metrologi dan Instrumentasi

Dr. Ir. Purwadi Agus Darwito, M.Sc.
NIP. 19620822 198803 1 001

TUGAS AKHIR

RANCANG BANGUN SISTEM MONITORING KECEPATAN KAKI MANUSIA SAAT BERLARI DI ATAS TREADMILL BERBASIS MIKROKONTROLLER

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Ahli Madya
pada
Program Studi D3 Metrologi dan Instrumentasi
Jurusan Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh :
RIZA FARIDA
NRP. 2413 031 028

Disetujui oleh Tim Penguji Tugas Akhir :

1. Ir. Tutug Dhanardono, MT ..... Pembimbing
2. Dr. Ir. Purwadi Agus Darwito, M.Sc ..... Ketua Penguji
3. Hendra Cordova, ST. MT ..... Dosen Penguji I
4. Ir. Yerri Susatio, MT ..... Dosen Penguji II
5. Murry Raditya, ST.MT ..... Dosen Penguji III

SURABAYA
JULI, 2016

RANCANG BANGUN SISTEM MONITORING KECEPATAN KAKI MANUSIA SAAT BERLARI DI ATAS TREADMILL BERBASIS MIKROKONTROLLER

Nama	**: Riza Farida**
NRP	**: 2413031028**
Jurusan	**: D3 Metrologi & Instrumentasi Teknik Fisika FTI-ITS**
Dosen Pembimbing	**: Ir. Tutug Dhanardono, MT**

Abstrak

Olahraga yang dilakukan secara teratur akan terhindar dari berbagai macam penyakit. Salah satunya berolahraga dengan menggunakan *treadmill*. Teknologi di dalamnya memberikan kemudahan bagi masyarakat untuk menggunakannya. Seperti dengan mengendalikan kecepatan motor agar dapat menyesuaikan dengan kemampuan berlari pengguna. Sensor yang digunakan adalah accelerometer untuk mendeteksi kecepatan kaki (m/s). Data kemudian dikirim melalui wireless ke mikrokontroller utama ATMega 128 pada treadmill. Program Sistem Monitoring dibuat pada software codevisionAVR. Bersamaan dengan data waktu yang dihasilkan oleh modul RTC, data kecepatan kaki akan ditampilkan ke display LCD dan disimpan ke memori MicroSD dalam format ".txt". Kemudian dilakukan pengujian pertama Modul RTC selama 10 hari dan waktu yang dihasilkan oleh modul RTC sama dengan waktu sebenarnya. Pengujian kedua yaitu pengujian modul MMC dengan tiga variasi nilai kapasitas microSD yang berbeda yaitu 1, 2 dan 4 GB serta variasi nilai interrupt timer 1,5,10, dan 30 detik sekali. Hasilnya adalah semakin besar kapasitas microSD dan nilai interrupt timernya maka semakin besar pula kapasitas penyimpannya. Data yang tersimpan dibandingkan dengan pembacaan stroboskop untuk mengukur kecepatan motor. Hasil perhitungan nilai error sebesar 1,21 % sehingga masih dalam batas toleransi.

Kata kunci : ATMega128, RTC, MicroSD, CodevisionAVR

DESIGN OF SYSTEM MONITORING SPEED OF THE HUMAN FOOT WHEN RUNNING ON A TREADMILL BASED MICROCONTROLLER

Name	***:***	**Riza Farida**
NRP	***:***	**2413 031 028**
Department	***:***	***Diploma Program of Metrology and Instrumentation, Engineering Physics FTI-ITS***
Counselor Lecturer	***:***	**Ir. Tutug Dhanardono, MT**

Abstract

Exercise done on a regular basis will be protected from various kinds of diseases. One of them exercising using a treadmill. Technology in it makes it easy for people to use. Such as by controlling the motor speed in order to adapt to the user running ability. Accelerometer sensor is used to detect the speed of the foot (m / s). The data is then sent via wireless to the main microcontroller ATMega 128 on a treadmill. Program Monitoring System created on CodeVisionAVR software. Along with the time data generated by the RTC module, foot speed data will be displayed to the LCD display and stored to memory MicroSD in ".txt" format. Then do the first test module RTC for 10 days and times generated by the RTC module is equal to the actual time. The second test is test MMC module with three different variations of capacity microSD ie 1, 2 and 4 GB as well as variations timer interrupt 1,5,10, and 30 seconds. The result is the larger capacity and greater microSD interrupt timer value, the greater the capacity stockpiles. Data stored compared with the reading stroboscopes to measure the speed of the motor. The comparison results have an error of 1.21%, so still within tolerance

***Keyword** : ATMega128, RTC, MicroSD, CodevisionAVR*

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT dan baginda besar rasulullah SAW atas segala rahmat dan anugerahnya sehingga penulis dapat menyelesaikan tugas akhir ini dengan judul :

"Rancang Bangun Sistem Monitoring Kecepatan Kaki Manusia Saat Berlari Di Atas Treadmill Berbasis Mikrokontroller"

Tugas akhir ini disusun guna memenuhi persyaratan bagi seorang mahasiswa untuk memperoleh geplar Ahli Madya dengan program studi D3 Metrologi dan Instrumentasi, Jurusan Teknik Fisika, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.

Penulis mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun secara tidak langsung dalam pengerjaan dan pembelajaran tugas akhir ini. Beberapa pihak tersebut antara lain:

1. Bapak Agus Muhamad Hatta, ST, MSi, Ph.D selaku Kepala Jurusan Teknik Fisika ITS Surabaya.
2. Kedua Orang tua saya yang selalu memberikan dukungan moril, dan doa, dan juga selalu memberikan semangat kepada penulis sehingga dapat menyelesaikan Tugas Akhir ini.
3. Kakak, adik, dan keluarga saya yang selalu memberikan semangat dan doa.
4. Bapak Dr. Ir. Purwadi Agus Darwito, M.Sc selaku Ketua Program Studi Diploma III Metrologi dan Teknik Instrumentasi, Institut Teknologi Sepuluh Nopember.
5. Bapak Ir. Tutug Dhanardono, MT selaku Dosen Pembimbing yang setia mendampingi, membimbing, mengkritisi, dan memotivasi pengerjaan tugas akhir ini.

6. Ibu Ir. Ronny Dwi Noryati, M. Kes. selaku Dosen Wali dan yang telah memberikan arahan selama menjalani masa perkuliahan hingga menyelesaikan tugas akhir ini.
7. Kawan-kawan TEAM Treadmill yang telah bersama dan berjuang mengerjakan Tugas Akhir ini.
8. Teman – Teman Pejuang tangguh angkatan 2013 yang telah membantu dan selalu memberi motivasi.
9. Teman teman D3 Metrologi dan Instrumentasi angkatan 2011, 2012, 2013, 2014 dan para senior yang selalu mendukung dalam segala hal.
10. Cindy, Dila ,Oliv, Irma, Nay, Hana yang telah memberi semangat.
11. Semua Pihak yang tidak dapat kami sebutkan satu persatu, yang telah banyak membantu dalam mengerjakan tugas akhir ini sampai selesai.

Penulis menyadari bahwa tugas akhir ini tidaklah sempurna, tetapi penulis berharap ini dapat memberikan kontribusi yang berarti dan dapat menambah wawasan bagi pembaca. Semoga awal dari permulaan yang panjang ini dapat membawa manfaat dan hikmat bagi kita semua dan juga semoga hari esok lebih baik dari hari ini.

Surabaya, 27 Juli 2016

Penulis

DAFTAR ISI

HALAMAN JUDUL I.....i
HALAMAN JUDUL II.....ii
LEMBAR PENGESAHAN I.....iii
LEMBAR PENGESAHAN II.....iv
ABSTRAK.....v
ABSTRACT.....*vi*
KATA PENGANTAR.....vii
DAFTAR ISI.....ix
DAFTAR GAMBAR.....xi
DAFTAR TABEL.....xii

BAB I PENDAHULUAN
1.1 Latar Belakang.....1
1.2 Rumusan Permasalahan.....2
1.3 Batasan Masalah.....2
1.4 Tujuan.....2
1.5 Materi.....3

BAB II TEORI PENUNJANG
2.1 Treadmill.....5
2.2 Sensor.....6
2.3 Mikrokontroller ATMega 128.....7
2.4 Real Time Clock (RTC) DS1307.....8
2.5 Media Penyimpanan Data.....11
2.6 LCD 20x4.....14
2.7 Software CodevisionAVR.....15
2.8 Stroboscope.....16

BAB III PERANCANGAN DAN PEMBUATAN ALAT
3.1 Studio Literatur.....20
3.2 Perancangan sistem.....20
3.3 Perancangan Hardware.....21
3.4 Perancangan Software.....22

3.5 Integrasi.....22
3.6 Uji Performansi.....23
3.7 Analisa Data dan Pembahasan.....23

BAB IV ANALISA DATA DAN PEMBAHASAN
4.1 Pengujian Modul RTC.....23
4.2 Pengujian Modul MMC.....28
4.3 Pengujian alat secara keseluruhan.....32
4.4 Pembahasan.....36

BAB V PENUTUP
5.1 Kesimpulan.....39
5.2 Saran.....39

DAFTAR PUSTAKA
LAMPIRAN
BIODATA PENULIS

DAFTAR GAMBAR

Gambar 2. 1 Treadmill. 5
Gambar 2. 2 Konfigurasi Pin ATmega128 8
Gambar 2. 3 Bentuk fisik DS1307 9
Gambar 2. 4 Format pengiriman data jam ke DS1307 10
Gambar 2. 5 Format pengiriman data tanggal ke DS1307 10
Gambar 2. 6 Format pembacaan data dari DS1307 11
Gambar 2.7 Register penyimpanan informasi kalender di dalam DS1307 11
Gambar 2.8 Bentuk fisik dan dimensi SD Card, Mini SD dan MicroSD 12
Gambar 2.9 Perhitungan tipe FAT 13
Gambar 2.10 LCD 20x4 15
Gambar 2.11 Software CodevisionAVR 17
Gambar 2.12 Stroboskop 18
Gambar 3.1 *Flowchart* metodologi penelitian 19
Gambar 3.2 Skema Kerja *Smart Treadmill* 20
Gambar 3.3 Skema kerja sistem monitoring kecepatan langkah kaki 21
Gambar 3. 4 Rangkaian Modul RTC DS1307 21
Gambar 3. 5 Rangkaian MMC dan LCD 20x4 22
Gambar 4. 1 Modul RTC 25
Gambar 4.2 Hasil Pengujian Modul RTC 27
Gambar 4.3 Modul MMC 28
Gambar 4.4 Hasil Pengujian Modul MMC 28
Gambar 4.5 Struktur disk pada microSD 30
Gambar 4.6 Grafik perbandingan data MMC dengan Stroboscope 36

DAFTAR TABEL

Tabel 4.1 Data Perbandingan Waktu antara modul RTC dan waktu sebenarnya..... 27
Tabel 4.2 Perhitungan sector size dan sector count pada mmc . 30
Tabel 4.3 Perhitungan Batas Waktu Peenyimpanan *MicroSD*.. 31
Tabel 4.4 Ketentuan nilai untuk kecepatan motor..... 32
Tabel 4.5 Data Pengujian MMC..... 33
Tabel 4.6 Perbandingan data MMC dengan stroboscope..... 34
Tabel 4.7 Perhitungan error data kecepatan kaki dengan pembacaan stroboscope 35

BAB I
PENDAHULUAN

1.1 Latar Belakang

Di perkotaan yang hanya memiliki sedikit lahan terbuka ,banyak orang memilih pusat kebugaran sebagai tempat olahraga. Pusat kebugaran menyediakan fasilitas olahraga yang biasanya lengkap seperti *cycle ergometry, treadmill* dan banyak macam alat lain. Di masyarakat yang pola gaya hidupnya sudah sangat berubah, terutama dikarenakan gaya hidup modern, kegiatan olahraga sudah menjadi kebutuhan yang mutlak harus dilakukan. Kegiatan masyarakat modern yang cenderung mengurangi gerak fisik dan ditambah tingkat stres yang tinggi serta perubahan pola makan, menyebabkan orang berusaha mencari solusi tepat untuk segala ancaman yang membahayakan kesehatannya .Alasan terbanyak tentu untuk mempertahankan kebugaran dan kesehatan jasmani. Olahraga yang dilakukan secara teratur akan terhindar dari berbagai macam penyakit. Salah satunya berolahraga dengan menggunakan *treadmill*. *Treadmill* adalah sebuah alat olahraga yang banyak digunakan oleh masyarakat modern saat ini yang lebih menyukai berolahraga di rumah. Kecanggihan teknologi di dalamnya memberikan kemudahan bagi masyarakat untuk menggunakannya. Seperti halnya dengan mengendalikan kecepatan motor agar dapat menyesuaikan dengan kemampuan berlari pengguna. Oleh karena itu, pengukuran kecepatan berlari pada manusia pada *treadmill* ini sangat diperlukan sebagai sistem monitoring yang diharapkan dapat memberikan informasi dan kemudahan dalam mengoperasikannya. Sistem monitoring ini akan ditampilkan pada sebuah display dan perekam data selama *treadmill* tersebut dijalankan. Dari latar belakang itulah, Tugas Akhir dengan judul "Rancang bangun sistem monitoring kecepatan kaki manusia saat berlari di atas *treadmill* berbasis mikrokotroller" ini diharapkan dapat memberikan sebuah inovasi terbaru pada sebuat alat olahraga *treadmill* dan sebagai solusi dari masalah tersebut.

1.2 Rumusan Permasalahan

Dari latar belakang diatas maka permasalahan yang diangkat dalam Tugas Akhir ini adalah sebagai berikut :

- Bagaimana membuat *real time clock* modul pada rancang bangun *treadmill*?
- Bagaimana cara menampilkan data kecepatan kaki manusia pada *display treadmill*?
- Bagaimana cara merekam data kecepatan kaki manusia secara *real time* ke dalam *micro*SD?

1.3 Batasan Masalah

Adapun batasan masalah dalam rancangan tugas akhir ini dengan tujuan keterarahan konsep , adalah ebagai berikut:

- Cakupan dari system monitoring ini dimulai dari penerimaan data hasil dari pembacaan sensor, pengolahan data, tampilan pada LCD serta rekam data pada *micro*SD.
- Perancangan sistem monitoring ini hanya menggunakan mikrokontroller ATMega 128
- Sistem akan bekerja apabila semua elemen sudah dalam keadaan aktif dan terintegrasi meliputi *wireless*, modul RTC dan MMC.

1.4 Tujuan

Tujuan utama dari penelitian ini adalah untuk memenuhi mata kuliah tugas akhir sebagai syarat kelulusan dari program studi diploma 3 metrologi dan instrumentasi, sebagai penerapan ilmu instrumentasi, Teknik otomasi , dan Mikrokontroler. Adapun tujuan lainnya adalah seabagi berikut:

- Merancang *real time clock* modul pada rancang bangun *treadmill*.
- Mengetahui cara menampilkan data kecepatan kaki manusia pada *display treadmill.*
- Mengetahui cara merekam data kecepatan kaki secara *real time* ke dalam memori *micro*SD.

1.5 Manfaat

Manfaat yang diperoleh dari penyelesaian tugas akhir ini adalah terciptanya sistem monitoring kecepatan kaki manusia ketika berlari di atas *treadmill* sehingga pengguna dapat mengetahui kondisi dirinya saat berolahraga dengan *treadmill*.

(Halaman ini sengaja dikosongkan)

BAB II
TEORI PENUNJANG

2.1 Treadmill

Treadmil adalah alat olah raga yang sering digunakan di pusat-pusat kebugaran seperti Gym. Namun banyak orang yang akhirnya membeli peralatan oleh raga sendiri di dalam rumahnya termasuk treadmil. Treadmill adalah alat yang digunakan untuk berjalan atau berlari di tempat yang sama. Alat fitness treadmill bisa digunakan untuk berjalan dan juga jogging. Biasanya alat ini memiliki banyak jenis, mulai dari yang hanya memiliki satu fungsi hingga yang memiliki banyak fungsi. Penggunaannya juga bermaca-macam, mulai dari cara manual, magnetik, sampai elektrik.

Gambar 2.1 Treadmill

Treadmill mejadi salah satu pilihan dari beberapa alat olah raga lainnya karena memili beberapa kelebihn diantaranya:

- Alat treadmill sangat ideal digunakan bagi orang yang ingin menurunkan berat badan yang berlebih dan juga dapat meningkatkan kepadatan tulang.
- Treadmill bisa digunakan oleh semua orang dengan berbagai kondisi tubuh termasuk orang yang pernah

mengalami cedera, misalnya; sakit punggung, cedera lutut, pergelangan kaki dan lutut lemah. Hal ini karena treadmill mengurangi tekanan pada sendi tidak seperti ketika berolah raga di jalan

- Treadmill juga dilengkapi dengan alat untuk memonitor detak jantung, mengetahui kalori yang terbakar, kecepatan berjalan/ berlari, dan juga jarak tempuh yang sudah dilalui.

2.2 Sensor

Sensor dan transduser merupakan peralatan atau komponen yang mempunyai peranan penting dalam sebuah sistem pengaturan otomatis. ketepatan dan kesesuaian dalam memilih sebuah sensor akan sangat menentukan kinerja dari sistem pengaturan secara otomatis. sensor adalah perangkat yang digunakan untuk mendeteksi fenomena fisik menjadi sinyal elektronik. sensor merupakan peralatan atau komponen yang mempunyai peranan penting dalam sebuah system pengaturan otomatis. ketepatan dan kesesuaian dalam memilih sebuah sensor akan sangat menentukan kinerja dari sistem pengaturan secara otomatis Sensor tidak dapat berdiri sendiri karena membutuhkan pengkondisi sinyal yang harus kompatibel dengan perangkat keras pengukuran. pengkondisi sinyal dapat terdiri atas penguat (untuk sensor yang menghasilkan sinyal dengan level yang sangat rendah), filter (untuk membatasi noise pada sinyal), isolasi (untuk melindungi perangkat pengukuran dari masukan yang berbahaya), serta rangkaian lain yang dibutuhkan untuk menyesuaikan sensor terhadap perangkat pengukuran.Selain itu impedansi masukan pengkondisi sinyal harus dibuat setinggi mungkin agar tidak terjadi jatuh tegangan pada keluaran sensor sehingga tidak mengurangi akurasi pengukuran. Sinyal listrik yang dihasilkan sensor harus dioptimasi untuk rentang masukan yang sesuai dengan perangkat akuisisi data.

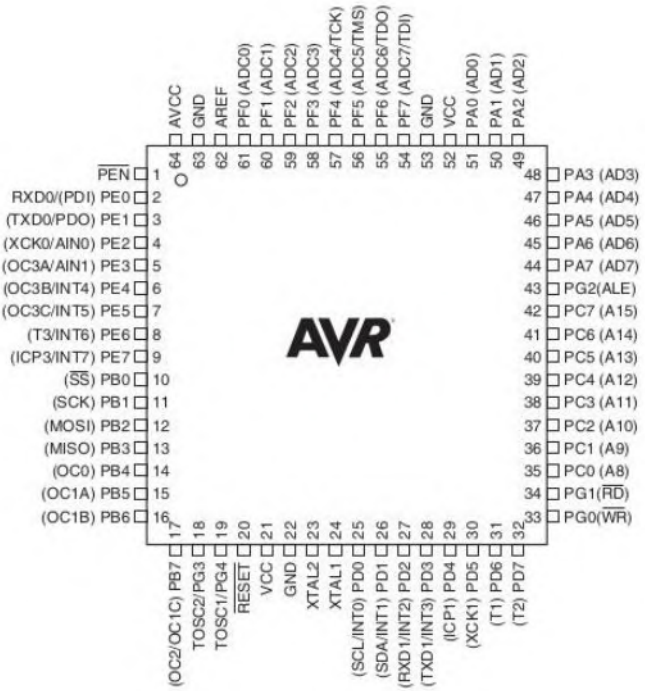
Sensor yang digunakan pada perancangan alat kali ini adalah sensor Accelerometer Sensor ini akan dirancang bersama dengan mikronkontroller yang akan diletakkan di pergelangan kaki. Sensor akan membaca pergerakan kaki

dengan membaca titik koordinat kaki. Sensor ini nantinya kan mengirim data melalui wire less dan akan di terima oleh mikrokontroller utama untuk pengolahan data

2.3 Mikrokontroller ATMega 128

Mikrokontroler dapat dianalogikan sebagai sebuah sistem komputer yang dikemas dalam sebuah chip. Artinya bahwa di dalam sebuah IC mikrokontroler sebenarnya sudah terdapat kebutuhan minimal agar mikrokontroler bisa bekerja, yaitu meliputi mikroprosesor, ROM, RAM, I/O dan clock seperti yang dimiliki oleh sebuah PC [7]. Mikrokontroler AVR dikelompokkan menjadi empat kelas yaitu keluarga ATtiny, AT90Sxx, ATmega, AT86RFxx. Perbedaan pada masing-masing keluarga AVR adalah kapasitas memori, peripheral, dan fungsi. Mikrokontroler ATmega128 merupakan salah satu varian dari mikrokontroler AVR 8-bit yang diproduksi oleh Atmel. Salah satu jenis mikrokontroler dari keluarga AVR adalah ATmega128. Mikrokontroler ini memiliki 64 pin yang masing-masing pin memiliki fungsi terendiri. ATmega 128 merupakan mikrokontroler 8-bit teknologi CMOS dengan konsumsi daya rendah yang berbasis arsitektur enhanced RISC AVR. Dengan eksekusi instruksi yang sebagian besar hanya menggunakan satu siklus clock, ATmega 128 mencapai throughput sekitar 1 MIPS per MHz . Port I/O sebanyak 32 bit, yang dikelompokkan dalam Port A, Port B, Port C, dan Port D. SRAM sebesar512 byte. Memori Flash sebesar 8 Kbyte dan EEPROM sebesar 512 byte [2,3].

ATmega 128 merupakan mikrokontroler keluarga AVR yang mempunyai kapasitas flash memori 128KB. AVR (Alf and Vegard’s Risc Processor) merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (Reduced Instruction Set Computer). Dengan mengeksekusi instruksi kuat dalam satu siklus clock tunggal, ATmega128 mencapai throughput mendekati 1 MIPS per MHz yang memungkinkan perancang sistem untuk mengoptimalkan konsumsi daya melawan kecepatan proses.



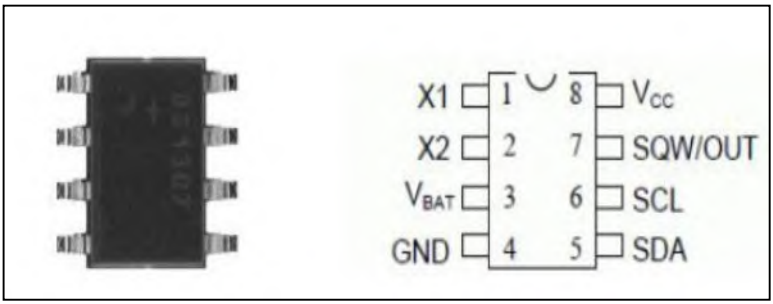
Gambar 2.2 Konfigurasi Pin ATmega128

Datasheet Manual DT-COMBO BASIC BASE BOARD
(*Terlampir pada lampiran 1*)

2.4 *Real Time Clock* (RTC) DS1307

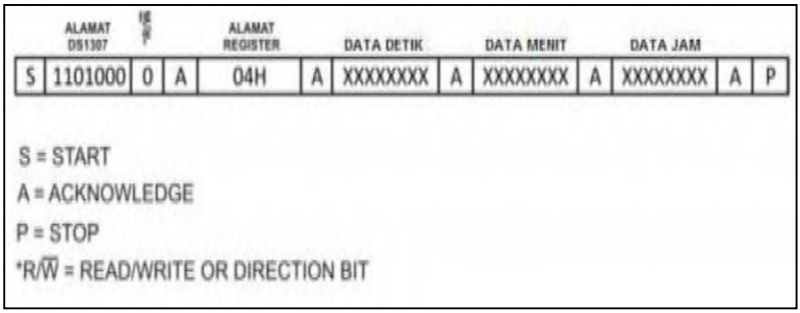
Agar pencatatan pada Micro SD teridentifikasi dengan baik, pada setiap pencatatan diperlukan waktu lokal pencatatan. Waktu lokal pencatatan dibangun menggunakan IC DS1307. DS1307 merupakan RTC buatan Dallas-Maxim Semiconductor®. Tanggal pada DS1307 menyediakan informasi detik, menit, jam, tanggal, bulan dan tahun (Pracoyo, 2008).

Akhir dari tanggal otomatis diatur untuk bulan dengan kurang daripada 31 hari, termasuk koreksi tahun kabisat. *Clock* dapat mengoperasikan dalam format 24 jam atau 12 jam dengan indikator AM/PM (Jarin, 2008).



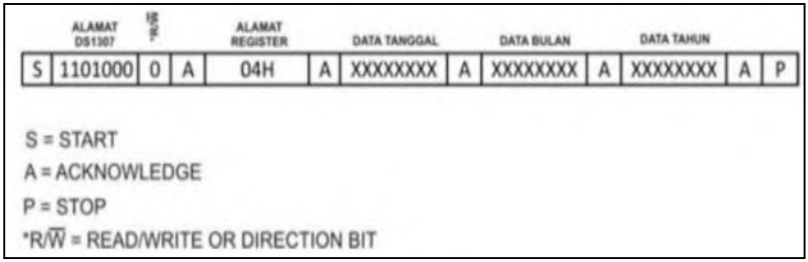
Gambar 2.3 Bentuk fisik DS1307

DS1307 memiliki sistem *nonvolatile* (permanen) SRAM 56 bytes dimana alamat dan data dikirimkan secara serial perbit dengan menggunakan sistem *Inter Integrated Circuit* (I2C) yang dikembangkan oleh Philips Semikonduktor® hampir 20 tahun silam untuk memudahkan komunikasi antar komponen dalam sebuah papan PCB (Susilo, 2010). I2C adalah komunikasi serial antar rangkaian terintegrasi. Komunikasi serial I2C selalu diawali dengan kondisi *start* dan diakhiri *stop*. Kondisi *start* adalah ketika terjadi perubahan kondisi dari *high* ke *low* pada SDA ketika SCL pada kondisi *high*, sedangkan kondisi *stop* adalah ketika terjadi perubahan kondisi dari *low* ke *high* pada SDA ketika SCL pada kondisi *high* (Budiharto dan Rizal, 2007). Pada komunikasi I2C antara DS1307 dan mikrokontroler, data-data jam dan yang lain dikirim dalam format *full binary-coded-decimal* (BCD). Data dikirim per-*byte* dari mikrokontroler ke DS1307 secara serial. Setelah menerima satu *byte* data, DS1307 sebagai *slave* akan mengirimkan sinyal *acknowledge* ke mikrokontroler. Sinyal *acknowledge* adalah tanda bahwa satu *byte* data telah diterima oleh DS1307. Format pengiriman data waktu ke DS1307 ditunjukkan pada Gambar 2.4



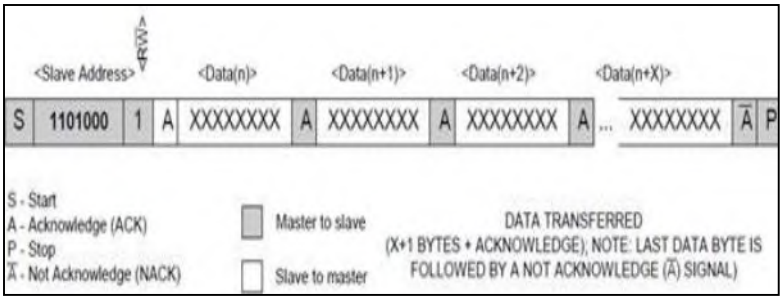
Gambar 2.4 Format pengiriman data jam ke DS1307

Data tanggal ditulis mulai alamat 04H, untuk pengaturan hari tidak dilakukan karena DS1307 otomatis akan menyesuaikan hari berdasarkan tanggal yang telah diatur. Untuk format pengiriman data tanggal ditunjukkan pada Gambar 2.5.



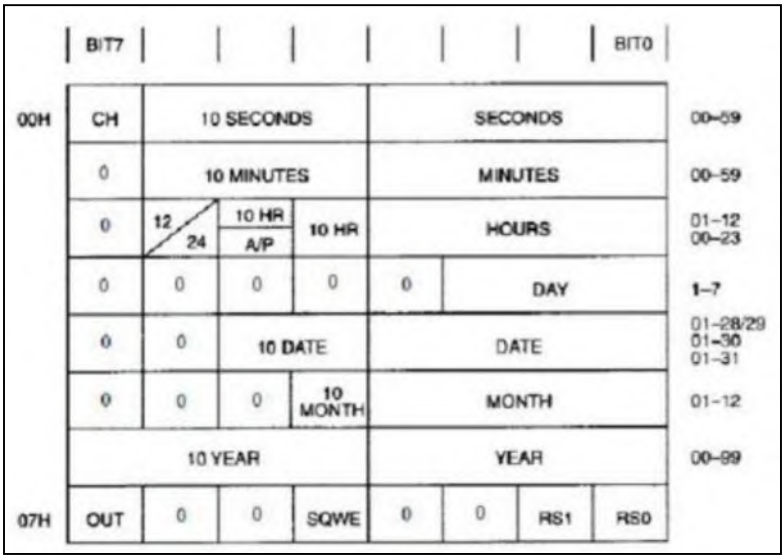
Gambar 2.5 Format pengiriman data tanggal ke DS1307.

Awal komunikasi, mikrokontroler akan membangkitkan sinyal start I2C. Kemudian mikrokontroler akan mengirimkan alamat grup dari DS1307 yaitu 1101 dan kode untuk menulis ke DS1307 yaitu 0 pada LSB-nya. Kode tulis ini diberikan ke DS1307 karena mikrokontroler akan menuliskan terlebih dahulu alamat awal pembacaan register DS1307 yaitu 00H. Setelah itu, proses pembacaan I2C dimulai. Format pembacaan data dari DS1307 oleh mikrokontroler ditunjukkan pada Gambar 2.6



Gambar 2.6 Format pembacaan data dari DS1307

Mikrokontroler mengirimkan alamat DS1307 beserta kode instruksi baca. Proses pembacaan dilakukan per-byte dimulai dari register detik sampai register tahun, alamat masing-masing register tersebut ditunjukkan pada Gambar 9.13



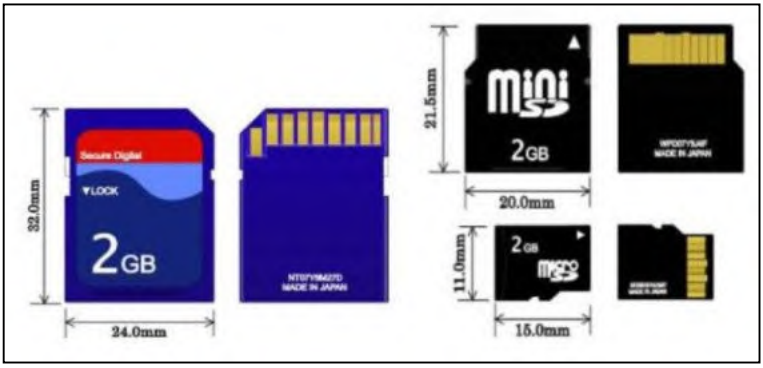
Gambar 2.7 Register penyimpanan informasi kalender di dalam DS1307

Setelah menerima data 1-*byte* utuh mikrokontroler akan mengirimkan sinyal *acknowledge* ke DS1307. Pada akhir pembacaan register tahun, akan dikirim sinyal *notacknowledge* dan diakhiri dengan sinyal stop (Putra, 2010).

Datasheet DS1307
(*Terlampir pada lampiran 2*)

2.5 Media Penyimpanan Data

Micro SD seringkali digunakan sebagai sarana penyimpan data pada *Personal Digital Assistant* (PDA), kamera digital, dan telepon seluler (ponsel). SD card memiliki dimensi 32 mm x 24 mm x 2,1 mm (panjang x lebar x tebal). Pengembangan lebih lanjut dari media penyimpanan ini menghasilkan dimensi yang lebih kecil dan kompak seiring dengan perkembangan zaman yang berupa Mini SD dan Micro SD seperti yang ditunjukan Gambar 2.5

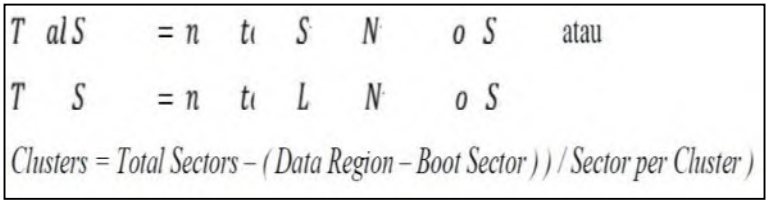


Gambar 2.8 Bentuk fisik dan dimensi SD Card, Mini SD, dan MicroSD.

Ada tiga macam cara berkomunikasi dengan Micro SD, yaitu: 1. *One-bit SDmode*; 2. *Four-bit SD mode*; 3. *Serial Peripheral Interface* (SPI) *mode*. Cara komunikasi yang terakhir merupakan cara termudah karena protokolnya mudah dipelajari, tersedia dokumentasi, dan berlisensi gratis. Sehingga komunikasi yang umum digunakan menggunakan mikrokontroler adalah SPI

(Sunardi dkk, 2009). Format data pada Micro SD umumnya menggunakan format *File Alocation Table* (FAT). FAT pertama kali dikembangkan oleh Bill Gates dan Marc McDonald, pada tahun 1976-1977. FAT *file systems* adalah sebuah sistem berkas yang menggunakan struktur tabel alokasi berkas sebagai cara dirinya beroperasi. Dengan adanya FAT *file systems* memungkinkan data disimpan dalam file yang dapat langsung dikenali oleh sistem operasi dengan format ekstensi *text document* (*.txt) sehingga data hasil rekaman dapat langsung dibaca dan dipindahkan ke dalam komputer . Nama file dan direktori di file sistem FAT berbentuk 8 titik 3 (maksimal delapan kareakter sebuah titik dan minimal tiga karakter), semua karakter selalu disimpan dalam huruf besar. Struktur direktori sangat sederana yaitu file diberikan slot pertama yang dapat dipakai, ada delapan atribut dasar pada suatu file yang dapat dipakai untuk menandakanapakan suatau file hanya dapat dibaca (read only), tersembunyi (hidden), sistem atau telah diubah sejak backup yang terakhir (archive)

FAT12 digunakan untuk kapasitas 16 MB ke bawah, FAT16 digunakan untuk kapasitas 32 MB hingga 2 GB, dan FAT32 umumnya digunakan untuk kapasitas di atas 2 GB (Haryono dan Surmayono, 2006). Dilakukan perhitungan sebagai berikut untuk mengetahui tipe FAT yang digunakan:



Gambar 2.9 perhitungan tipe FAT

Cluster adalah satuan sistem FAT yang terkecil dari penyimpanan data. Satu cluster terdiri atas satu nomor tertentu dari disk sector , catatan-catatan FAT berisi cluster yang dipakai dan tidak terpakai dan file-file diletakan di dalam cluster. File

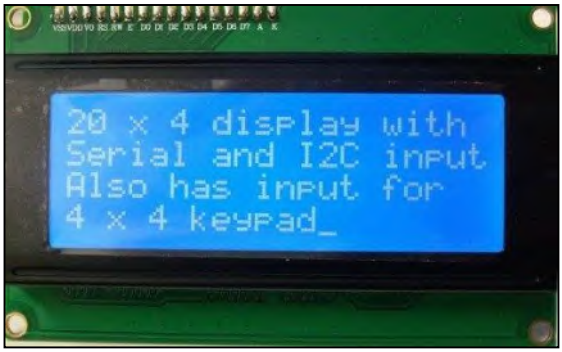
sistem FAT mendukung ukuran partisi sampai dengan 2 GB, tetapi hanya menyediakan maksimal 65.525 cluster. Oleh karena itu, apapun ukuran harddisk atau partisi nomor sector dalam satu cluster haruslah cukup besar. Sehingga semua ruang yang ada dapat dimasuki sampai 65.525 cluster. Pada umumnya, cluster yang besar cenderung lebih banyak membuang tempat dai pada cluster yang kecil. File sistem FAT digunakan juga pada sumber direktori, direktori ini maksimal membolehkan nomor yang masuk dan mungkin terletak pada tempat tertentu pada disk atau partisi. Sistem operasi yang digunakan file sistem FAT menunjukan sumber direktori dengan karakter slash terbalik (\). Bagian awal direktori ini menunjukan boot-up

Jika clusters lebih kecil dari 4085 berarti FAT bertipe FAT12. Jika clusters lebih kecil dari 65525 berarti FAT bertipe FAT16. Di atas itu, FAT yang digunakan adalah FAT32. Format FAT16 menyerupai FAT12. Perbedaan yang paling jelas diantara keduanya adalah bahwa FAT16 menggunakan tabel FAT sebanyak 16 bit per alamat sedangkan FAT12 menggunakan 12 bit (Anonimous, 2011 B).Nilai *cluster* awal menunjuk ke nomor *cluster* awal dari sebuah data. Jika entri berupa direktori, maka *cluster* awal menunjuk ke *cluster* yang mengandung awal sub-direktori. Jika entri berupa file, maka cluster awal menunjuk ke cluster yang mengandung awal file. Ukuran file adalah ukuran total file dalam satuan byte. Ukurannya tidak boleh melebihi 4 GB untuk satu file. Untuk entri selain berupa file, nilainya harus 0. Isi dari file dituliskan pada alamat *First Sector of Cluster N* dengan cluster N adalah nilai cluster awal. Jika ukuran file melebihi 1 cluster, maka isi selanjutnya dapat dituliskan pada cluster lain. Tabel FAT pun harus disesuaikan dengan perubahan ini agar sistem dapat mengetahui isi file secara berurutan. Jika ingin melakukan penambahan isi file, maka parameter yang harus dicari atau dihitung tidak hanya alamat cluster awal dan *First Sector* namun juga Ukuran File, *Bytes per Sector*, dan *Sector per Cluster*. Dari ukuran file dan *Bytes per Sector* didapat jumlah sektor yang ditempati file. Dari nilai tersebut dan *Sector per Cluster* didapat jumlah cluster yang ditempati file .

2.6 LCD 20 X 4

Display elektronik adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suatu data, baik karakter, huruf ataupun grafik. LCD (Liquid Cristal Display) 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. LCD (Liquid Cristal Display) berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik.

Material LCD (Liquid Cristal Display) LCD adalah 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.



Gambar 2.10 LCD 20x4

2.7 Software CodevisionAVR

CodeVisionAVR pada dasarnya merupakan perangkat lunak pemrograman microcontroller keluarga AVR berbasis bahasa C. Ada tiga komponen penting yang telah diintegrasikan dalam perangkat lunak ini: Compiler C, IDE dan Program generator. Berdasarkan spesifikasi yang dikeluarkan oleh perusahaan pengembangnya, Compiler C yang digunakan hampir mengimplementasikan semua komponen standar yang ada pada bahasa C standar ANSI (seperti struktur program, jenis tipe data, jenis operator, dan library fungsi standar-berikut penamaannya). Tetapi walaupun demikian, dibandingkan bahasa C untuk aplikasi komputer, compiler C untuk microcontroller ini memiliki sedikit perbedaan yang disesuaikan dengan arsitektur AVR tempat program C tersebut ditanamkan (embedded). Khusus untuk library fungsi, disamping library standar (seperti fungsi-fungsi matematik, manipulasi String, pengaksesan memori dan sebagainya), CodeVisionAVR juga menyediakan fungsi-fungsi tambahan yang sangat bermanfaat dalam pemrograman antarmuka AVR dengan perangkat luar yang umum digunakan dalam aplikasi kontrol. Beberapa fungsi library yang penting diantaranya adalah fungsi-fungsi untuk pengaksesan LCD, komunikasi I2C, IC RTC (Real time Clock), sensor suhu LM75, SPI (Serial Peripheral Interface) dan lain sebagainya. Untuk memudahkan pengembangan program aplikasi, CodeVisionAVR juga dilengkapi IDE yang sangat user friendly . CodeVisionAVR ini telah mengintegrasikan perangkat lunak downloader (in system programmer) yang dapat digunakan untuk mentransfer kode mesin hasil kompilasi kedalam sistem memori microcontroller AVR yang sedang deprogram.. CVAVR ini dapat mengimplematasikan hampir semua interuksi bahasa C yang sesuai dengan arsitektur AVR, bahkan terdapat beberapa keunggulan tambahan untuk memenuhi keunggulan spesifikasi dari CodeVisionAVR yaitu hasil kompilasi studio debugger dari ATMEL. Integrated Development Environtment (IDE) telah diadaptasikan pada chip AVR yaitu In-System Programmer software, memungkinkan programmer untuk mentransfer

program ke chip mikrokontroler secara otomatis setelah proses assembly/kompilasi berhasil. In-System Programmer software didesign untuk bekerja dan dapat berjalan dengan perangkat lunak lain seperti AVR Dragon, AVRISP, Atmel STK500, dan lain sebagainya.

Gambar 2.11 Software CodevisionAVR

2.8 Stroboscope

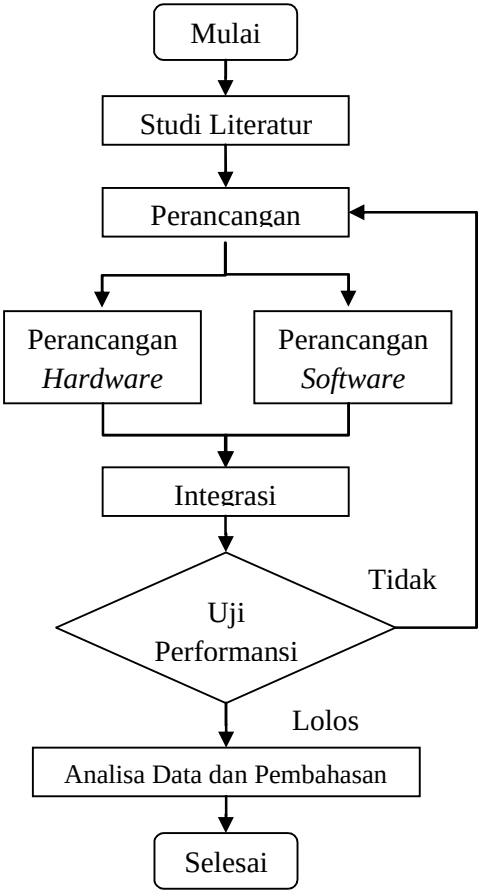
Stroboscope merupakan salah satu alat yang lebih canggih dan aman untuk mengukur kecepatan dengan cara tanpa kontak. Stroboscope menggunakan sumber sinar cahaya yang dapat disinkronisasi dengan setiap kecepatan dan pengulangan gerakan sehingga benda yan berpindah sangat cepat terlihat tidak bergerak atau berpindah perlahan Contoh penggunaan dari stroboscope ini dapat dilakukan dengan memasang sebuah disket putih dengan titik hitam terpasang pada poros (as) dari motor 1800 rpm. Bila disket berputar pada 1800 rpm; tidak mungkin terlihat oleh mata gambaran tunggal dan titik akan tampak menjadi lingkaran

kabur. Bila diterangi oleh cahaya stroboscope, maka cahaya akan disinkronkan dengan setiap putaran disket. sebagai contoh bila titik berada pada jam tiga, titik akan terlihat pada posisi ini pada kecepatan 1800 kali untuk setiap menit. Oleh karena itu, titik akan Nampak diam pada posisi tadi. Jika laju sinar dari stroboscope diperlambat menjadi 1799 sinar per menit, titik akan teriluminasi pada posisi cahaya yang berbeda.

Gambar 2.11 Stroboskop

BAB III
PERANCANGAN DAN PEMBUATAN ALAT

Adapun langkah-langkah yang akan dilakukan untuk merancang dan membuat smart treadmill adalah sebagai berikut :



Gambar 3.1 *Flowchart* metodologi penelitian

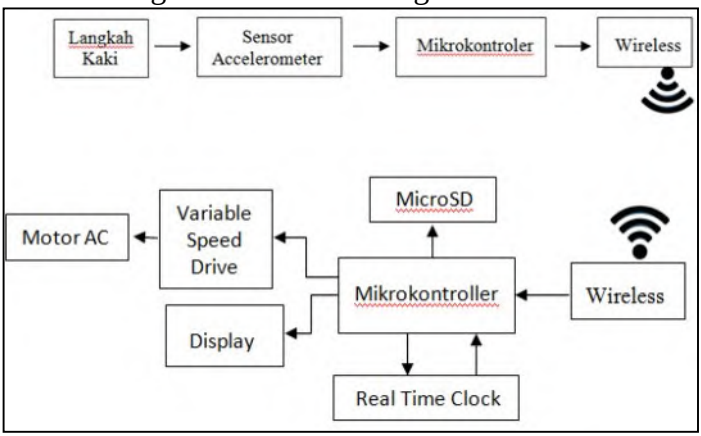
Adapun penjelasan dari flowchart metodologi penelitian mengenai “Rancang bangun sistem monitoring detak jantung dan kecepatan kaki manusia saat berlari di atas treadmill berbasis mikrokotroller” adalah sebagai berikut :

3.1 Studi Literatur

Studi literatur sebagai upaya pehaman materi untung menunjang kegiatan tugas akhir ini yaitu mengenai “Rancang bangun sistem monitoring detak jantung dan kecepatan kaki manusia saat berlari di atas treadmill berbasis mikrokotroller” studi literatur dilakukan dengan mencari dan mempelajari informasi dari seluruh *e-book* dan *manual book* yang berkaitan dengan tugas akhir ini

3.2 Perancangan Sistem

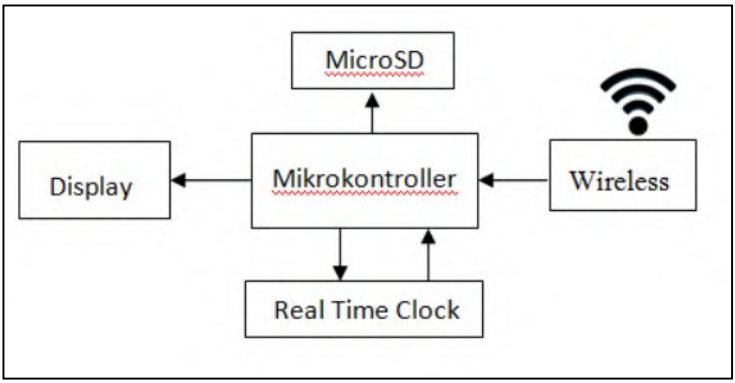
Setelah melakukan studi literatur, dilanjutkan dengan melakukan perancangan sistem, yang meliputi perancangan *hardware* yaitu menentukan *hardware* apa saja yang akan dipakai dan perancangan *software* yaitu menentukan *software* untuk menampilkan data yang telah diperoleh dari *hardware* sebagai sistem monitoring.



Gambar 3.2 Skema Kerja *Smart Treadmill*

3.3 Perancangan Hardware

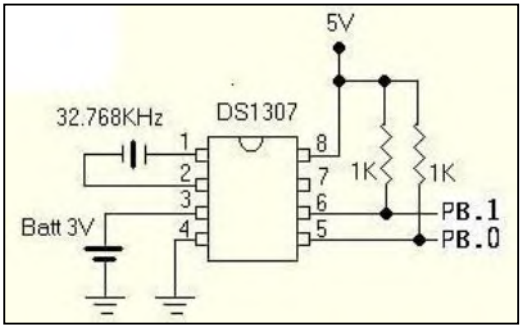
Adapun skema kerja dari sitem monitoring kecepatan kaki dan detak jantung adalah sebagai berikut:



Gambar 3.3 Skema kerja sistem monitoring kecepatanlangkah kaki

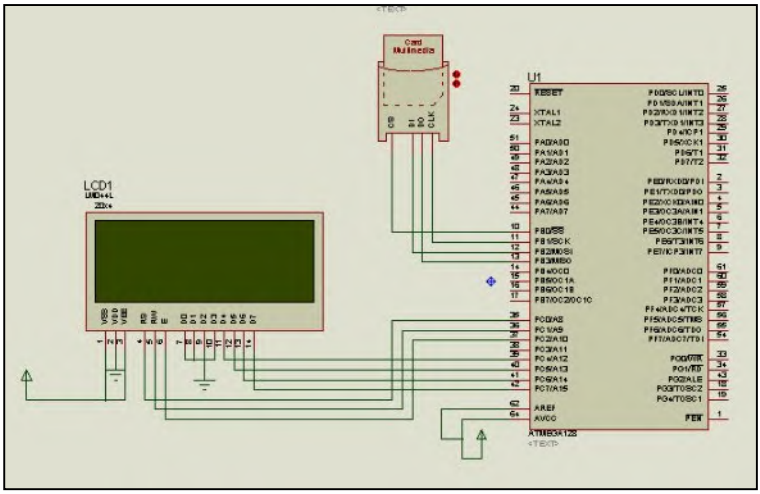
Sistem monitoring yang akan dibuat yaitu menggunakan beberapa hardware sebagai berikut :

- Wireless RF KYL 200
- Mikrokontroller ATMega 128
- Modul *Real Time Clock* (RTC) DS1307



Gambar 3.4 Rangkaian Modul RTC DS1307

- Modul Penyimpanan Data *(microSD)*
- *Display (LCD 20x4)*



Gambar 3.5 Rangkaian MMC dan LCD 20x4

3.4 Perancangan Software

Adapun rancangan program yang sudah dibuat pada software CodevisionAVR adalah sebagai berikut:

(*Terlampir pada lampiran 3*)

3.5 Integrasi

Perancangan ini juga mengintegrasikan sistem komunikasi wireless dimana data yang terbaca oleh sensor akan dikirim ke mikrokontroller untuk diolah. Pengolahan data ini terbagi menjadi dua yaitu :

- Akuisisi data atau Logging data (*data logging*)

  Sistim akuisisi data dapat didefinisikan sebagai suatu sistem yang berfungsi untuk mengambil, mengumpulkan dan menyiapkan data, hingga memprosesnya untuk menghasilkan data yang dikehendaki

- *Data logger* (perekam data)

  Data logger (perekam data) adalah sebuah alat elektronik yang mencatat data dari waktu ke waktu baik yang terintegrasi dengan sensor dan instrumen didalamnya maupun ekternal sensor dan instrumen. Atau secara singkat data logger adalah alat untuk melakukan data logging.

3.6 Uji Performansi

Uji performansi ini yaitu pengujian respon time dilakukan dengan menganalisis waktu yang dibutuhkan sistem untuk melakukan tugasnya masing masing dengan menganalisa respon actuator ketika mendapatkan hasil pengolahan data logger.

3.7 Analisa Data dan Pembahasan

Setelah pembuatan rancangan telah selesai dengan hasil yang sesuai dengan yang di inginkan, kemudian dilakukan analisa data dari hasil uji performansi dan menarik kesimpulan dari data yang telah dihasilkan.

(Halaman ini sengaja dikosongkan)

BAB IV
ANALISA DATA DAN PEMBAHASAN

4.1. Pengujian Modul RTC

Adapun rangkaian modul RTC yang sudah dibuat adalah seperti yang ditunjukan pada gambar 4.1

Gambar 4.1 Modul RTC

Modul RTC ini menggunakan IC DS1307 dan protocol 12C (Inter IC Bus) yang sesuai dengan protocol TWI (2 wire serial interfaces) pada mikronkontroller AVR. Penggunaan IC DS1307 dilakukan dua proses yakni menulis data ke IC dan yang kedua yaitu membaca data waktu dari IC. Berikut ini adalah program pada codevisionAVR untuk men-setting Real Time Clock

```
....
void settimertc(void)
{
rtc_set_time(11,59,0);
rtc_set_date(3,23,06,16);
}
```

```
void gettimertc(void)
{
rtc_get_time(&hour,&minute,&second);  // mengakses jam
rtc_get_date(&week,&day,&month,&year);  //mengakses tanggal
}

void rtc_lcd ()
{
 lcd_gotoxy(0,0);
 sprintf(temp,"%02u:%02u:%02u",hour,minute,second);
 lcd_puts(temp);

 lcd_gotoxy(10,0);
 sprintf(temp,"%02u-%02u-%02u",day,month,year);
 lcd_puts(temp);
  }

void main (void)
...
lcd_init(20);
lcd_clear();
settimertc();

.....

while (1)
        {
        gettimertc();
        rtc_lcd();
        }
```

Program tersebut akan mensetting nilai awal pada modul RTC. Data akan disimpan pada IC DS1307 . program akan terus berjalan tanpa harus men-download program lagi ke

mikrokontroller hal ini disebabkan karena DS1307 mendapatkan supply secara terus menerus meskipun mikrokontroller dalam keadaan off. Program diatas hanya untuk memberikan nilai awal. Kemudian program “settimertc()” yang ada pada void utama dihapus dan program di download ulang ke dalam mikrokontroller sehingga setiap download program baru, nilai awal RTC tidak kembali ke nilai awal RTC yang sudah disetting sebelumnya.

Berikut adalah hasil dari rangkaian modul yang sudah dibuat seperti yang ditunjukkan pada gambar 4.2

Gambar 4.2 Hasil Pengujian Modul RTC

Tabel 4.1 Data Perbandingan Waktu antara modul RTC dan waktu sebenarnya

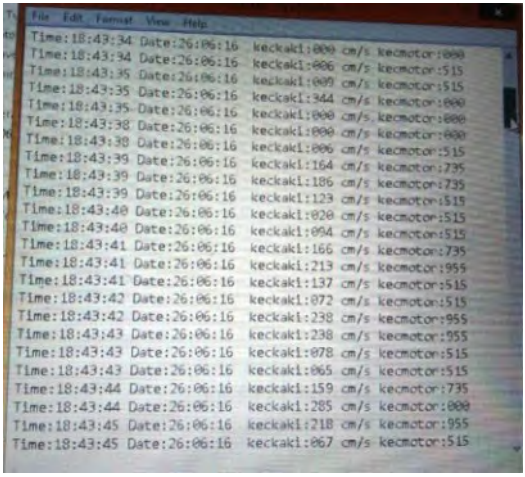
Pengujian Ke-	Pembacaan	
	Real Time	Hasil RTC Modul
1	09:30:25 29-05-2016	09:30:25 29-05-2016
2	10:05:45 30-05-2016	10:05:45 30-05-2016
3	09:54:25 31-05-2016	09:54:25 31-05-2016
4	08:40:13 01-06-2016	08:40:13 01-06-2016
5	09:30:36 02-06-2016	09:30:36 02-06-2016
6	10:36:47 03-06-2016	10:36:47 03-06-2016
7	08:32:53 04-06-2016	08:32:53 04-06-2016
8	09:31:25 05-06-2016	09:31:25 05-06-2016
9	09:28:30 06-06-2016	09:28:30 06-06-2016
10	10:09:51 07-06-2016	10:09:51 07-06-2016

4.2 Pengujian Modul MMC

Adapun rangkaian modul MMC yang sudah dibuat adalah seperti yang ditunjukan pada gambar 4.3

Gambar 4.3 Modul MMC

Pengujian modul penyimpanan data (data logger) dengan menggunakan MMC delakukan dengan mengkoneksikan ke modul mikrokontroller. Langkah berikutnya dengan menuliskan data ke mmc. Pengujian dilakukan dengan membuat file dan mengisis file di MMC. Berikut ini hasil pengujiaan MMC seperti yang ditunjukkan pada gambar 4.4



Gambar 4.4 Hasil Pengujian Modul MMC

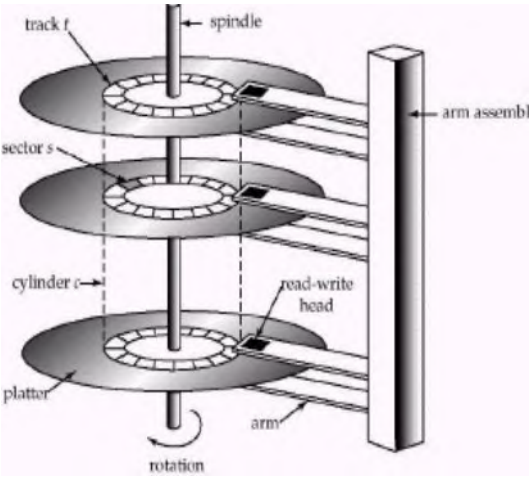
Berdasarkan hasil pengujian MMC bisa menyimpan data yang dikrimkan oleh sensor accelerometer melalui wireless sehingga dapat disimpulkan modul MMC dapat bekerja dengan baik.

4.2.1 Proses Penyimpanan

Disk umumnya terdiri dari beberapa *plate*. Pada setiap plate terdapat dua permukaan (surface). Setiap permukaan ini dilapisi dengan lapisan magnetis. Setiap surface dibagi menjadi *track-track*. Kumpulan track pada semua permukaan yang terletak pada posisi yang sama membentuk *silinder*. Setiap track dibagi menjadi *sector-sector*. Semua sector ini mempunyai ukuran yang sama. Umumnya ukuran satu sector adalah 512 bytes. Sector merupakan unit penyimpanan data terkecil dalam disk (secondary storage).

Pada setiap permukaan terdapat *head*, yang berfungsi untuk membaca dan menulis data pada **sector** tertentu. Setiap head ini ditempelkan pada *disk arm*, yang berfungsi untuk memindahkan head ke posisi track yang dinginkan. Semua arm ini bergerak bersamaan ke posisi silinder yang diinginkan. Ketika terjadi pengaksesan file, disk arm memindahkan head ke track yang diinginkan, kemudian head akan menunggu sector yang tepat untuk diakses. Setelah menemukan sector yang tepat head mengakses data yang terdapat pada sector tersebut. Waktu yang dibutuhkan untuk mengambil/menyimpan data pada disk ditentukan oleh dua besaran waktu, yaitu Access time dan data transfer rate. Access time terdiri dari dua bagian, yaitu seek time (waktu yang dibutuhkan untuk memindahkan head ke posisi track yang tepat) dan rotational latency, yaitu waktu yang dibutuhkan oleh sector yang akan diakses untuk sampai ke posisi head. Data transfer rate adalah waktu yang dibutuhkan untuk melakukan transfer data dari head ke sector yang diinginkan.

Satu data yang tersimpan pada microSD berukuran 1byte. satu byte sama dengan delapan bit. 1024 byte sama dengan satu kilobyte. 1024 kilobyte sama dengan satu megabyte . 1024 megabyte sama dengan satu gigabyte. Satu data yang dimaksudkan adalah satu karakter, misal “a” atau “1” saja.



Gambar 4.5 Struktur disk pada microSD

4.2.2 Perhitungan Kapsitas Memori

Percobaan dilakukan dengan menggunkan tiga *microSD* dengan kapasitas yang berbeda. Maka dihasilkan data sebagai berikut ;

Tabel 4.2. Perhitungan sector size dan sector count pada mmc

kapasitas memori (GB)	byte (M)	Sector size (byte)	sector count (M/512)
1	1073741824	512	2097152
2	2147483648	512	4194304
4	4294967296	512	8388608

Inisialisasi *MicroSD* pada ATMega 128 menghasilkan data sector size dan sector count dari *microSD* yang digunakan. Nilai dari sector size adalah 512 byte karena menggunakan FAT 32 8 bit. Sedangkan nilai sector count tergantung dari kapasitas *microSD* yang terlebih dahulu diubah ke satuan byte lalu dibagi

dengan nilai sector size. Semakin besar kapasitas *microSD* maka semakin besar pula nilai sector count.

Berikut merupakan formula perhitungan untuk kapasitas penyimpanan waktu data pada MMC. Hal ini dimaksudkan bahwa penggunaan MMC berkapasitas Megabyte saja dapat menampung sejumlah data dengan waktu yang cukup lama.

$$? ? \frac{?}{???} ? ?? \frac{??}{?} ? ????$$

Dimana ,

x = Kapasitas waktu untuk menyimpanan data pada MMC
M = kapasitas memori pada MMC
512 = kapasitas per sector (addres) pada MMC (byte)
I = Nilai Interrupt timer yang diberikan pada sistem
60 = Jumlah detik per menit
60 = jumlah menit per jam

Tabel 4.3 Perhitungan Batas Waktu Peenyimpanan *MicroSD*

Ket waktu	**Interrupt Timer**			
	1 dtk sekali	**5 dtk sekali**	**10 dtk sekali**	**30 sekali**
Jam	582.5422222	2912.71111	5825.42222	17476.27
Hari	24.27259259	121.362963	242.725926	728.1778
Jam	1165.084444	5825.42222	11650.8444	34952.53
Hari	48.54518519	242.725926	485.451852	1456.356
Jam	2330.168889	11650.8444	23301.6889	69905.07
hari	97.09037037	485.451852	970.903704	2912.711

Pada rancang bangun sistem monitoring yng telah dibuat menggunakan *microSD* berukuran 1 GB. Interrupt timer yang digunakan adalah 1 detik yang artinya data disimpan ke *microSD* setiap satu detik sekali. Dari perhitungan tabel 4.3 dihasilkan

bahwa kapasitas *microSD* 1 GB dengan nilai interrupt timer satu detik dapat menyimpan data hingga 582 jam atau 24 hari.

4.3 Pengujian alat secara keseluruhan

4.3.1 Perbadingan Data Tersimpan MMC dengan Kecepatan Motor

Pengujian alat keseluruhan dilakukan dengan membandingkan data hasil perekaman dan data kecepatan motor AC yang ada pada treadmill. Data yang berhasil disimpan dalam sebuah microSD akan dibandingkan oleh alat ukur kecepatan motor yaitu stroboscope yang dianggap sebagai data referensi. Berikut adalah hasil perbandingan data kecepatan motor dengan pembacaan sensor yang terekam pada MMC. Adapun range nilai yang diberikan untuk menghasilkan kecepatan motor yang berbeda yaitu :

Tabel 4. 4 Ketentuan nilai untuk kecepatan motor

Range nilai (cm/s)	**nilai yang sudah dikehendaki (cm/s)**	**Kategori**	**RPM yang dikehendaki**
0	0	Diam	0
1-140	140	Pelan	515
141-200	200	sedang	735
201-260	260	Cepat	955
>260	0	berbahaya	0

Berdasarkan nilai pada tabel diatas, maka program untuk menggerakkan motor pada treadmill dapat ditentukan. Adapun hasil pengujian data yang terekam dengan kecepatan actuator adalah sebagai berikut:

Tabel 4.5 Data Pengujian MMC

Data MicroSD (cm/s)	**Data Pembacaan Sensor (cm/s)**	**Berdasarkan nilai yang sudah ditetapkan sebelumnya (tabel 4.2)**		
		(cm/s)	**rpm**	?
79	79	140	515	53.56
53	53	140	515	53.56
80	80	140	515	53.56
123	123	140	515	53.56
145	145	200	735	76.44
130	130	140	515	53.56
0	0	0	0	0
183	183	200	735	76.44
190	190	200	735	76.44
215	215	260	955	99.32
229	229	260	955	99.32
198	198	260	955	99.32
200	200	200	735	76.44
0	0	0	0	0
243	243	260	955	99.32
105	105	140	515	53.56
94	94	140	515	53.56
270	270	0	0	0
85	85	140	515	53.56
105	105	140	515	53.56

Tabel 4.5 menunjukkan bahwa data dalam MMC yang dibandingan dengan tampilan display hasil pembacaan sensor accelerometer sudah benar. Hal ini dikarenakan konversi data dari

mikro satu dan mikro dua sudah benar sehingga dapat diolah dengan baik.

Tabel 4.6 Perbandingan data MMC dengan stroboscope

Data MicroSD			**Pembacaan Stroboscope**			
cm/s	**RPM**	?	**RPM**	?	**v**	**cm/s**
140	515	???̈??	526.15	??'??	1.423	142.27
140	515	???̈??	526.89	??'???	1.425	142.47
140	515	???̈??	525.91	??'???	1.422	142.21
140	515	??'??	527.67	??'???	1.427	142.68
200	735	???̈??	744.89	??'???	2.014	201.42
140	515	??'??	528.76	??'???	1.43	142.98
0	0	?	0	?	0	0
200	735	???̈??	742.82	??'???	2.009	200.86
200	735	??'??	748.45	??'???	2.024	202.38
260	955	??'??	969.16	???'??	2.621	262.06
260	955	??'??	972.78	???'??	2.63	263.04
260	955	??'??	974.86	???'??	2.636	263.6
200	735	??'??	752.36	??'???	2.034	203.44
0	0	?	0	?	0	0
260	955	??'??	968.62	???'??	2.619	261.91
140	515	??'??	525.94	??'???	1.422	142.21
140	515	??'??	527.87	??'???	1.427	142.74
0	0	?	0	?	0	0
140	515	??'??	526.23	??'???	1.423	142.29
140	515	??'??	529.87	??'???	1.433	143.28

Perhitungan error dilakukan dari data yang tersimpan pada MMC dengan Kecepatan motor yang diukur dengan stroboskop. Data yang dihasilkan stroboskop adalah rpm. Sehingga dengan menggunakan persamaan:

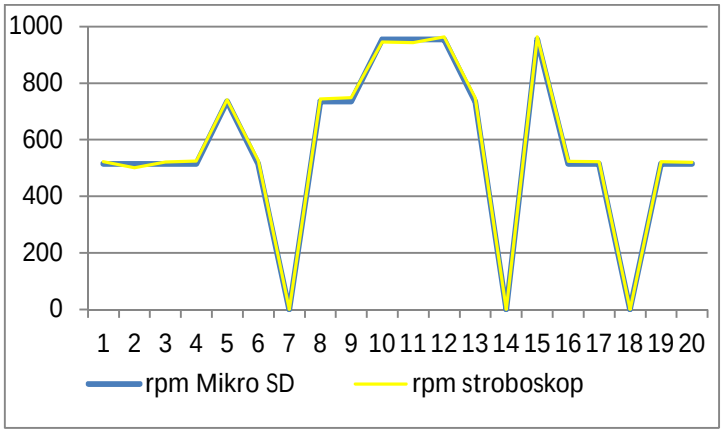
? ? ? ? …… (4.1)

Dimana :
V = kecepatan m/s
? ? ? ???????? ??????????
R = Jari jari m
1 rpm = 1x 2π/60 rad/s

Tabel 4.7 Perhitungan error data kecepatan kaki dengan pembacaan stroboscope

MikroSD cm/s	**Stroboscope cm/s**	**selisih**	**Error**	**% error**
140	142.27	2.27	0.016	1.6
140	142.47	2.47	0.017	1.73
140	142.21	2.20	0.016	1.55
140	142.68	2.68	0.019	1.88
200	201.42	1.41	0.007	0.7
140	142.98	2.97	0.021	2.08
0	0	0	0	0
200	200.86	0.85	0.004	0.43
200	202.38	2.38	0.012	1.18
260	262.06	2.06	0.008	0.79
260	263.04	3.04	0.012	1.16
260	263.6	3.60	0.014	1.37
200	203.44	3.43	0.017	1.69
0	0	0	0	0
260	261.91	1.91	0.007	0.73
e140	142.21	2.21	0.016	1.56
140	142.74	2.73	0.019	1.92
0	0	0	0	0
140	142.29	2.29	0.016	1.61
140	143.28	3.27	0.023	2.29
Rata-rata error			0.012	1.21

Dari perhitungan itulah pembacaan scroboscope dapat dikonversi menjadi data m/s dan didapatkan bahwa hasil kecepatan motor dengan data yang tersimpan pada mmc mempunyai error sebesar 1,33 % . Hal ini dikarenakan nilai yang diberikan program untuk menjalankan motor adalah nilai berbeda dengan pembacaan sensor. Pembacaan sensor yang terkadang tidak menunjukkan keakuratannya sehingga motor juga demikian.



Gambar 4.6 Grafik perbandingan data MMC dengan Stroboscope

4.4 Pembahasan

Sistem monitoring yang dibuat pada tugas akhir kali ini bertujuan untuk memonitoring kecepatan kaki secara real time yang disimpan pada sebuah *microSD* dengan format ".txt". modul RTC dibuat dengan rangkaian IC DS1307 yang dihubungkan dengan port D pada mikrokontroller ATMega 128 dan mendapatkan supply dari baterai 3v. Program dibuat pada software codevisionAVR untuk memberikan nilai awal jam, menit, detik, tanggal, bulan dan tahun. Setelah program dijalankan data akan tersimpan pada memori DS1307 dan tetap berjalan meskipun mikrokontroller dalam keadaan "OFF". Data yang dihasilkan ditampilkan pada LCD 20x4 dan duji coba

selama 10 hari berturut-turut. Data dibandingkan dengan waktu asli dan didapatkan hasil bahwa modul dapat berjalan dengan baik sesuai dengan waktu real tanpa menghasilkan selisih detik pun. Modul RTC ini juga digunakan untuk menghasilkan data waktu pada *microSD* bersamaan dengan data kecepatan kaki yang diterima dari mikrokontroller sesnsor accelerometer ke mikrokontroller utama pada *Treadmill*. Uji Modul MMC/*MicroSD* dilakukan dengan menginisialisasikan kapasitas *microSD* setelah itu membuat file dengan sistem FAT (File Allocation Table) dan menghasikan data berformat ".txt". Data notepad tersebut berisi data waktu meliputi detik , menit, jam, hari, bulan , tahun dan data kecepatan kaki. Sehingga dapat diketahui kecepatan kaki pada saat waktu tertertu. Selanjutnya dilakukan percobaan untuk mengetahui kapasitas waktu penyimpanan. Percobaan dilakukan menggunakan tiga *microSD* berkapasitas 1, 2 dan 4 GB. Ketiganya memiliki nilai sector size 512 byte karena menggunakan FAT 32 8 bit. Semakin besar nilai kapsitas *microSD* maka sector count semakin besar pula. Pada sistem monitoring ini menggunakan *microSD* berukuran 1 GB dengan variasi nilai interrupt timer 1, 5, 10, dan 30 detik sekali. Maka dihasilkan bahwa penyimpanan dapat dilakukan selama 582 jam yaitu maksimal selama 24 hari.

(Halaman ini sengaja dikosongkan)

LAMPIRAN 1
Datasheet Manual DT-COMBO BASIC BASE BOARD

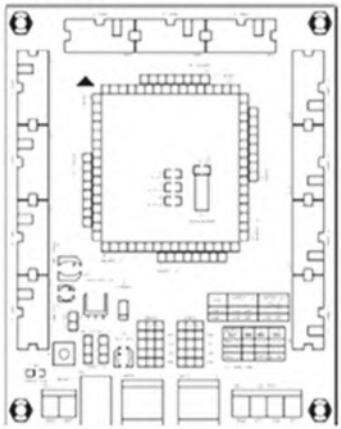
DT-COMBO
BASIC BASE BOARD

DT-COMBO BASIC BASE BOARD merupakan modul development base board untuk CPU MODULE Series yang memiliki kemampuan dan konektor untuk fungsi-fungsi dasar seperti input/output, serial UART, I²C, dan ADC. Modul ini dapat dihubungkan ke CPU MODULE Series dan digunakan untuk aplikasi sederhana hingga aplikasi yang cukup kompleks.

Spesifikasi

1. Mendukung CPU MODULE Series yang memiliki jalur Input/Output hingga 88 pin.
2. Tersedia 2 konektor RJ11 untuk komunikasi UART dengan 2 pilihan level tegangan, yaitu UART RS-232 atau UART TTL.
3. Tersedia terminal untuk komunikasi I²C.
4. Tersedia terminal untuk tegangan referensi ADC.
5. Tersedia tactile switch sebagai tombol reset.
6. Pilihan catu daya (VIN) 3,3 - 5,5 Volt DC (tanpa melalui regulator) atau 6,5 - 12 Volt DC (melalui regulator).
7. Tersedia voltage regulator 800 mA dengan pilihan output 3,3 Volt DC atau 5 Volt DC.
8. Kompatibel dengan CPU MODULE Series.

Tata Letak Komponen



DC Jack (J33) berfungsi sebagai jalur masuk catu daya 6,5 - 12 Volt (VIN).

Polaritas DC Jack J33

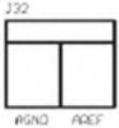
Jumper SSLCT1 (J34), SSLCT2 (J35), dan SSLCT3 (J31) berfungsi untuk mengatur konfigurasi tegangan input dan tegangan output.

	SSLCT1	SSLCT2	SSLCT3
Tanpa regulator	1 - 2	1 - 2	Don't care
Regulator 5V	2 - 3	2 - 3	OPEN
Regulator 3,3V	2 - 3	2 - 3	CLOSE

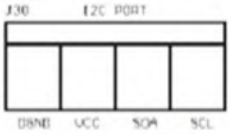
Penting!

- Perhatikan pengaturan jumper SSLCT1, SSLCT2, dan SSLCT3 sebelum menghubungkan sumber catu daya.

Terminal J32 merupakan konektor untuk tegangan referensi ADC (AREF).

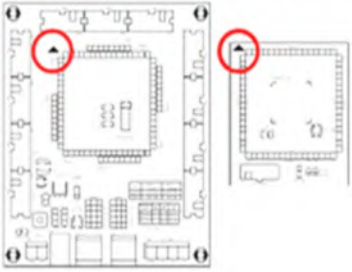


Terminal I2C PORT (J30) merupakan konektor untuk antarmuka I²C.



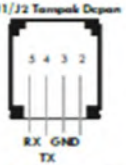
SCL dan SDA juga terhubung ke header PORT D.

Konektor SOCKET 1 (J7), SOCKET 2 (J8), SOCKET 3 (J9), SOCKET 4 (J10), SOCKET 5 (J3), SOCKET 6 (J4), SOCKET 7 (J5), dan SOCKET 8 (J6) merupakan jalur catu daya dan input/output yang dapat dihubungkan ke HEADER 1 - HEADER 8 pada CPU MODULE Series. Gunakan tanda segitiga pada kedua modul sebagai referensi. Kedua tanda segitiga harus berada pada posisi dan arah yang sama. Perhatikan juga posisi kaki CPU MODULE agar pemasangan persis sejajar antar pin (tidak bergeser).



Konektor RJ11 UART0 (J1) dan UART1 (J2) adalah konektor untuk jalur komunikasi UART.

Nama	Fungsi
GND	Referensi Ground
TX	Jalur data masuk ke board
RX	Jalur data keluar dari board



Level tegangan UART0 yang akan digunakan pada konektor RJ11 (J1) diatur melalui jumper J22 - J25.
Level tegangan UART1 yang akan digunakan pada konektor RJ11 (J2) diatur melalui jumper J26 - J29.

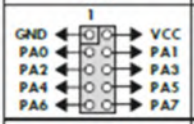
Level Tegangan	UART0 Posisi J22 - J25	UART1 Posisi J26 - J29
TTL	1 - 2	1 - 2
RS-232	2 - 3	2 - 3

Penting!

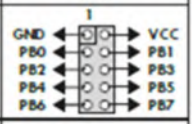
- Jika menggunakan header PORT E (J15) pin 3 dan 4 sebagai jalur I/O atau UART TTL, maka jumper J22 dan J24 harus dilepas.
- Jika menggunakan header PORT D (J14) pin 5 dan 6 sebagai jalur I/O atau UART TTL, maka jumper J26 dan J28 harus dilepas.
- Jika menggunakan RJ11 (J1) sebagai jalur komunikasi UART0, PORT E (J15) pin 3 dan 4 tidak boleh terhubung ke rangkaian lain.
- Jika menggunakan RJ11 (J2) sebagai jalur komunikasi UART1, PORT D (J14) pin 5 dan 6 tidak boleh terhubung ke rangkaian lain.
- Perhatikan pengaturan jumper sebelum menghubungkan device lain melalui RJ11.

Header PORT A (J11), PORT B (J12), PORT C (J13), PORT D (J14), PORT E (J15), PORT F (J16), PORT G (J17), PORT H (J18), PORT J (J19), PORT K (J20), dan PORT L (J21) berfungsi sebagai jalur I/O dari pin I/O mikrokontroler.

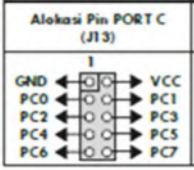
Alokasi Pin PORT A (J11)



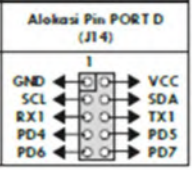
Alokasi Pin PORT B (J12)



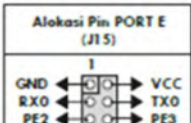
Alokasi Pin PORT C (J13)



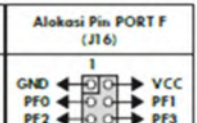
Alokasi Pin PORT D (J14)



Alokasi Pin PORT E (J15)



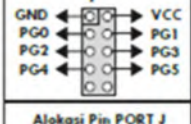
Alokasi Pin PORT F (J16)



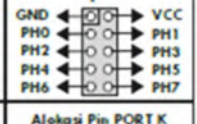
Alokasi Pin PORT G (J17)

Alokasi Pin PORT H (J18)

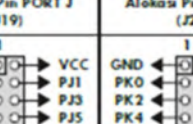
Alokasi Pin PORT J (J19)



Alokasi Pin PORT K (J20)



Alokasi Pin PORT L (J21)



Isi CD/DVD

1. CodeVisionAVR® versi evaluation.
2. Datasheet.
3. Manual DT-COMBO BASIC BASE BOARD.
4. Skema DT-COMBO BASIC BASE BOARD.
5. Prosedur Uji DT-COMBO BASIC BASE BOARD.
6. Website Innovative Electronics.

Prosedur Pengujian

Prosedur pengujian menggunakan DT-AVR ATMEGA64/128 CPU MODULE terdapat pada **Prosedur Uji DT-COMBO BASIC BASE BOARD.PDF** atau gunakan prosedur uji pada masing-masing CPU MODULE Series.

Trademark & Copyright

AVR is registered trademark of Atmel Corporation.
CodeVisionAVR is copyright by Pavel Haiduc, HP InfoTech s.r.l.

- *Terima Kasih atas kepercayaan Anda menggunakan produk kami. Bila ada kesulitan, pertanyaan atau saran mengenai produk ini, silakan menghubungi technical support kami:*

support@innovativeelectronics.com

LAMPIRAN 2
Datasheet DS1307

DALLAS SEMICONDUCTOR

DS1307
64 x 8 Serial Real-Time Clock

www.maxim-ic.com

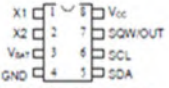
FEATURES

- Real-time clock (RTC) counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap-year compensation valid up to 2100
- 56-byte, battery-backed, nonvolatile (NV) RAM for data storage
- Two-wire serial interface
- Programmable squarewave output signal
- Automatic power-fail detect and switch circuitry
- Consumes less than 500nA in battery backup mode with oscillator running
- Optional industrial temperature range: -40°C to +85°C
- Available in 8-pin DIP or SOIC
- Underwriters Laboratory (UL) recognized

ORDERING INFORMATION

DS1307	8-Pin DIP (300-mil)
DS1307Z	8-Pin SOIC (150-mil)
DS1307N	8-Pin DIP (Industrial)
DS1307ZN	8-Pin SOIC (Industrial)

PIN ASSIGNMENT



DS1307 8-Pin DIP (300-mil)

DS1307 8-Pin SOIC (150-mil)

PIN DESCRIPTION

V_{CC}	- Primary Power Supply
X1, X2	- 32.768kHz Crystal Connection
V_{BAT}	- +3V Battery Input
GND	- Ground
SDA	- Serial Data
SCL	- Serial Clock
SQW/OUT	- Square Wave/Output Driver

DESCRIPTION

The DS1307 Serial Real-Time Clock is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially via a 2-wire, bi-directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power sense circuit that detects power failures and automatically switches to the battery supply.

LAMPIRAN 3
Program Sistem Monitoring Pada CodevisionAVR

```
/*****************************************************
This program was produced by the
CodeWizardAVR V2.05.3 Standard
Automatic Program Generator
© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 5/2/2016
Author  : Lukman
Company :
Comments:


Chip type               : ATmega128L
Program type            : Application
AVR Core Clock frequency: 7.372800 MHz
Memory model            : Small
External RAM size       : 0
Data Stack size         : 1024
*****************************************************/

#include <mega128.h>
#include <stdio.h>
#include <delay.h>
#include <stdlib.h>
#include <i2c.h>
#include <ds1307.h>
#include <alcd.h>
#include <sdcard.h>
#include <ff.h>
#include <spi.h>


/* 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;

/* 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 (0x10000L-(_MCU_CLOCK_FREQUENCY_/(T1_PRESC*T1_OVF_FREQ)))

#define xtal 8000000L
#define baud 9600

#define TOMBOL1 PINF.0


// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
TCNT1H=T1_INIT>>8;
TCNT1L=T1_INIT&0xFF;
/* card access low level timing function */
disk_timerproc();
}


flash char * flash error_msg[]=
{
"",
"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"
};

char buffer[100];
char filename[30];
unsigned int panjangdatanme;
unsigned char status,status_tulis;
unsigned int sector_size;
unsigned long int sector_count;
unsigned char hour,minute,second,week,day,month,year,temp [16] ;
char display_buffer[64];
unsigned char counter_detik ;
char input;
```

```
100
101 void settimertc(void)
102 {
103 rtc_set_time(8,28,0);
104 rtc_set_date(1,01,07,16);
105 }
106
107 void gettimertc(void)
108 {
109 rtc_get_time(&hour,&minute,&second); // menyekses jam
110 rtc_get_date(&week,&day,&month,&year); //menyekses tanggal
111
112 }
113
114 void rtc_lcd ()
115 {
116 lcd_gotoxy(0,0);
117 sprintf(temp,"%02u:%02u:%02u",hour,minute,second);
118 lcd_puts(temp);
119
120 lcd_gotoxy(10,0);
121 sprintf(temp,"%02u-%02u-%02u",day,month,year);
122 lcd_puts(temp);
123 }
124
125 void init_mmc (void)
126 {
127 status=disk_initialize(0);
128 status=0;
129     lcd_puts("Init OK/");
130     delay_ms(2000);
131     lcd_puts("Insert card");
132     delay_ms(2000);
133     /* clear the LCD */
134     lcd_clear();
135     /* get the sector size */
136     if (disk_ioctl(0,GET_SECTOR_SIZE,&sector_size)==RES_OK)
137        {
138        /* sector size read OK, display it */
139        sprintf(display_buffer,"sectorsize=%lu",sector_size);
140        lcd_gotoxy(0,1);
141        lcd_puts(display_buffer);
142        /* wait 2 seconds */
143        /* clear the LCD */
144        /* get the sector count */
145        if (disk_ioctl(0,GET_SECTOR_COUNT,&sector_count)==RES_OK)
146           {
147           /* sector count read OK, display it */
148           sprintf(display_buffer,"sectorcount=%lu",sector_count);
149           lcd_gotoxy(0,2);
150           lcd_puts(display_buffer);
151           }
152        else
153        lcd_puts("Error reading\nsector count");
154        }
155     else lcd_puts("Error reading\nsector size");
156     /* wait 2 seconds*/
157 delay_ms(2000);
158 lcd_clear();
159
160 }
161
162 void timer_mmc()
163 {
164     TCCR1A=0x00;
165     TCCR1B=0x05;
166     TCNT1H=TI_INIT>>8;
167     TCNT1L=TI_INIT&0xFF;
168     TIMSK=1<<TOIE1;
169 }
170
171 void stop_timer_mmc()
172 {
173     TCCR1A=0x00;
174     TCCR1B=0;
175     TCNT1H=TI_INIT>>8;
176     TCNT1L=TI_INIT&0xFF;
177     TIMSK&=~(1<<TOIE1);
178 }
179
180 void error(FRESULT res)
181 {
182 if ((res>=FR_DISK_ERR) && (res<=FR_TIMEOUT))
183   printf("ERROR: %p\r\n",error_msg[res]);
184 }
185
186 void display_status(char *file_name)
187 {
188 if ((res=f_stat(file_name,&finfo))==FR_OK)
189 {
190 }
191 }
192
193 //================================================
194 void inisialisasi_fat(void)
195 {
196     timer_mmc();
197     for(;;)
198     {
199         if ((res=f_mount(0,&fat))==FR_OK)
200         {
201             break;
202         }
203         else
204         {
205             error(res);
206         }
```

```
    }
    stop_timer_mac();
}

void reset_filename(void)
{
    unsigned char i_reset;
    for(i_reset=0;i_reset<30;i_reset++)
    {
        filename[i_reset]=0;
    }
}

void file_data (void)
{
    filename[0]='0';
    filename[1]=':';
    filename[2]='r';
    filename[3]='i';
    filename[4]='m';
    filename[5]='m';
    filename[6]='.';
    filename[7]='t';
    filename[8]='x';
    filename[9]='t';
}
```

```
void file_baru(void)
{
    unsigned char ulang;
    status=0;
    for(ulang=0;ulang<10;ulang++)
    {
        reset_filename();
        file_data();
        timer_mac();

        if ((res=f_open(&file,filename,FA_CREATE_NEW))==FR_OK)status=1;
        else status=0;

        if(status==1)
        {
            if ((res=f_close(&file))==FR_OK)status=1;
            else status=0;
        }

        if(status==1)break;

    }
    stop_timer_mac();
}
```

```
void isi_file(void)
{
    unsigned ulang;
    status_tulis=0;

    reset_filename();
    file_data();
    display_status(filename);

    for(ulang=0;ulang<10;ulang++)
    {
            timer_mac();
            if ((res=f_open(&file,filename,FA_WRITE))==FR_OK)status_tulis=1;
            else status_tulis=0;

            if(status_tulis==1)
            {
                if ((res=f_lseek(&file,finfo.fsize))==FR_OK)status_tulis=1;
                else status_tulis=0;
            }

            if(status_tulis==1)
            {
                if ((res=f_write(&file,buffer,panjangdatamac+1,&bytes))==FR_OK)status_tulis=1;
                else status_tulis=0;
            }

            if(status_tulis==1)
            {
                if ((res=f_close(&file))==FR_OK)status_tulis=1;
                else status_tulis=0;
            }
            if(status_tulis==1)break;
            delay_ms(500);
    }
    stop_timer_mac();
}

void rtc_mac (void)
{
    panjangdatamac=0;
    buffer[panjangdatamac]='T';
    panjangdatamac++;
    buffer[panjangdatamac]='i';
    panjangdatamac++;
    buffer[panjangdatamac]='m';
    panjangdatamac++;
    buffer[panjangdatamac]='e';
    panjangdatamac++;
    buffer[panjangdatamac]=':';

  panjangdatamac++;
    buffer[panjangdatamac]=(hour / 10)   + 48;
    panjangdatamac++;
    buffer[panjangdatamac]=(hour % 10)   + 48;
    panjangdatamac++;
    buffer[panjangdatamac]=':';
    panjangdatamac++;
    buffer[panjangdatamac]=(minute / 10)   + 48;
    panjangdatamac++;
    buffer[panjangdatamac]=(minute % 10)   + 48;
    panjangdatamac++;
    buffer[panjangdatamac]=':';
    panjangdatamac++;
```

```
    buffer[panjangdatasmac]=((second / 10)   + 48);
    panjangdatasmac++;
    buffer[panjangdatasmac]=((second % 10)   + 48);
    panjangdatasmac++;
    buffer[panjangdatasmac]=' ';

    panjangdatasmac++;
    buffer[panjangdatasmac]='D';
    panjangdatasmac++;
    buffer[panjangdatasmac]='a';
    panjangdatasmac++;
    buffer[panjangdatasmac]='t';
    panjangdatasmac++;
    buffer[panjangdatasmac]='e';
    panjangdatasmac++;
    buffer[panjangdatasmac]=':';

    panjangdatasmac++;
    buffer[panjangdatasmac]=((day / 10)   + 48);
    panjangdatasmac++;
    buffer[panjangdatasmac]=((day % 10)   + 48);
    panjangdatasmac++;
    buffer[panjangdatasmac]='-';
    panjangdatasmac++;
    buffer[panjangdatasmac]=((month / 10)   + 48);
    panjangdatasmac++;
    buffer[panjangdatasmac]=((month % 10)   + 48);
    panjangdatasmac++;
    buffer[panjangdatasmac]='-';
    panjangdatasmac++;
    buffer[panjangdatasmac]=((year / 10)   + 48);
    panjangdatasmac++;
    buffer[panjangdatasmac]=((year % 10)   + 48);
    panjangdatasmac++;
    buffer[panjangdatasmac]=' ';
     panjangdatasmac++;
}

unsigned char buf_rx[3],x;
unsigned int hasilkonversi=0;

unsigned int strtoint (void)
{
for (x=0;x<3;x++)
{
input = getchar();
lcd_gotoxy(x,3);
lcd_putchar(input);
putchar(input);
buf_rx[x]= input - 0x30;
}
hasilkonversi = ((int)buf_rx[0]*100) + ((int)buf_rx[1]*10)+((int)buf_rx[2]);
return hasilkonversi;
}

int data_int;
int rpm;

void kec_mac (void)
{
 panjangdatasmac++;
 buffer[panjangdatasmac]='k';
 panjangdatasmac++;
 buffer[panjangdatasmac]='e';
 panjangdatasmac++;
 buffer[panjangdatasmac]='c';
 panjangdatasmac++;
 buffer[panjangdatasmac]='h';
 panjangdatasmac++;
 buffer[panjangdatasmac]='a';
 panjangdatasmac++;
 buffer[panjangdatasmac]='h';
 panjangdatasmac++;
 buffer[panjangdatasmac]='i';
 panjangdatasmac++;
 buffer[panjangdatasmac]=':';
 panjangdatasmac++;
 buffer[panjangdatasmac]=(data_int/100)+48;
 panjangdatasmac++;
 buffer[panjangdatasmac]=((data_int%100)/10)+48;
 panjangdatasmac++;
 buffer[panjangdatasmac]=(data_int%10)+48;
 panjangdatasmac++;
 buffer[panjangdatasmac]=' ';
 panjangdatasmac++;
 buffer[panjangdatasmac]='r';
 panjangdatasmac++;
 buffer[panjangdatasmac]='m';
 panjangdatasmac++;
 buffer[panjangdatasmac]='/';
 panjangdatasmac++;
 buffer[panjangdatasmac]='s';
 panjangdatasmac++;
 buffer[panjangdatasmac]=' ';
 }

void motor (void)
{
 if (data_int>=1 && data_int<=140)
{
 OCR0=154;
 lcd_gotoxy(11,3);
 lcd_putsf("516 rpm");
 rpm = 516;
 }

 else if (data_int>=141 && data_int<=200)
{
 OCR0=169;
 lcd_gotoxy(11,3);
 lcd_putsf("735 rpm");
 rpm = 735;
 }
```

```
else if (data_int>=201 && data_int<=260)
{
OCR0=102;
lcd_gotoxy(11,3);
lcd_putsf("955 rpm");
rpm = 955;
}

else
{
OCR0=0;
lcd_gotoxy(11,3);
lcd_putsf("0   rpm");
rpm=0;
}
}

void motor_mac (void)
{
//
panjangdatamac++;
 buffer[panjangdatamac]='k';
 panjangdatamac++;
 buffer[panjangdatamac]='e';
 panjangdatamac++;
 buffer[panjangdatamac]='c';
 panjangdatamac++;
 buffer[panjangdatamac]='m';
 panjangdatamac++;
 buffer[panjangdatamac]='o';
 panjangdatamac++;
 buffer[panjangdatamac]='b';
 panjangdatamac++;
 buffer[panjangdatamac]='o';
 panjangdatamac++;
 buffer[panjangdatamac]='r';
 panjangdatamac++;
 buffer[panjangdatamac]=':';
  panjangdatamac++;
 buffer[panjangdatamac]=(rpm/100)+48;
 panjangdatamac++;
 buffer[panjangdatamac]=((rpm%100)/10)+48;
 panjangdatamac++;
 buffer[panjangdatamac]=(rpm%10)+48;
 panjangdatamac++;
 buffer[panjangdatamac]=13;
 panjangdatamac++;
 buffer[panjangdatamac]=10;
}

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

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=Out Func1=Out Func0=Out
// State7=T State6=T State5=T State4=T State3=T State2=0 State1=0 State0=0
PORTB=0x00;
DDRB=0x17;

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

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

// Port E 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
PORTE=0x00;
DDRE=0x00;

// Port F 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
PORTF=0x00;
DDRF=0x00;

// Port G initialization
// Func4=In Func3=In Func2=In Func1=In Func0=In
// State4=T State3=T State2=T State1=T State0=T
PORTG=0x00;
DDRG=0x00;

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

```
TCCR0=0x71;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 7372.800 kHz
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// OC1C 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
// Compare C Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x05;
TCNT1H=T1_INIT>>8;
TCNT1L=T1_INIT&0xFF;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
OCR1CH=0x00;
OCR1CL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// Timer/Counter 3 initialization
// Clock source: System Clock
// Clock value: Timer3 Stopped
// Mode: Normal top=0xFFFF
// OC3A output: Discon.
// OC3B output: Discon.
// OC3C output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer3 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR3A=0x00;
TCCR3B=0x00;
TCNT3H=0x00;
TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;
OCR3CH=0x00;
OCR3CL=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
// INT3: Off
// INT4: Off
// INT5: Off
// INT6: Off
// INT7: Off
EICRA=0x00;
EICRB=0x00;
EIMSK=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x04;


ETIMSK=0x00;


UCSR0A=0x00;
UCSR0B=0x10;
UCSR0C=0x06;
UBRR0H=(xtal/16/baud-1) >> 8;
UBRR0L=(xtal/16/baud-1) & 0xFF;

// USART1 initialization
// USART1 disabled
UCSR1B=0x00;

// Analog Comparator initialization
```

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

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI Type: Master
// SPI Clock Rate: 1843.200 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: 0
rtc_init(0,0,0);

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 0
// 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: 20
lcd_init(20);
lcd_clear();
//settimertc();

// Global enable interrupts
#asm("sei")
init_mmc();
inisialisasi_fat();

lcd_gotoxy(0,2);
lcd_putsf("kaki:");
lcd_gotoxy(5,3);
lcd_putsf("cm/s");
lcd_gotoxy(11,2);
lcd_putsf("motor:");

file_baru();

while (1)
      {
      gettimertc();
      rtc_lcd();

      data_int= strtoint();
      motor();

      rtc_mmc();
      kec_mmc();
      motor_mmc();
       isi_file();

        }
       }
```

BAB V
PENUTUP

5.1 Kesimpulan

Adapun Kesimpulan yang dapat diambil dari pembuatan tugas akhir ini adalh sebagai berikut:

- Sistem monitoring kecepatan kaki ini menggunkan modul Real Time Clock dengan IC DS1307 yang dapat menghasilkan data waktu secara real yang terdiri dari detik, menit , jam, tanggal, bulan dan tahun.
- Tampilan data kecepatan kaki dan detak jantung manusia secara real time menggunakan modul LCD 20x4 yang dapat menampilkan angka berupa karakter sesuai dengan program yang sudah diatur sebelumnya
- data kecepatan kaki secara real time direkam oleh memori microSD disertai dengan waktunya dan dihasilkan bahwa hasil rekaman data dibanding kecepatan motor mempunyai error sebesar 1,21% dan masih dalam batas toleransi.

5.2 Saran

Beberapa hal yang perlu diperhatikan dalam kelanjutan plant ini adalah sebagai berikut :

- alat ini dapat disempurnakan dengan menggunakan beberapa metode agar respon sistem berjalan lebih cepat dari sebelumnya.
- desain treadmill dapat diperbaiki agar tingkat keamanan dan kenyamanan pengguna dapat diutamakan.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

Albert Paul, Malvino. 2003. "**Prinsip-Prinsip Elektronika Buku Satu**".Salemba Teknika,Jakarta.

Albert Paul, Malvino. 2003. "**Prinsip-Prinsip Elektronik Edisi Tiga**".Salemba Teknika Jakarta.

Andrianto, Heri. 2013. "**Pemrograman Mikrokontroller AVR Atmega 128**". Informatika, Bandung.

Bejo, Agu. 2008. "**C&AVR Rahasia Kemudahan Bahasa C dalam Mikrokontroller ATMega 8535**".Graha Ilmu, Yogyakarta.

Elektronika Dasar. 2011.Transistor Sebagai Saklar, ?transistor-sebagai-saklar/. Diakses 16 maret 2016.

Ikhsan, Maulana. 2013. RTC (Real Time Clock) DS1307 (Code AVR), http://elektronika dasar.web.id/teori elektronika Diakses 17 Maret 2016.

Darko, I Gd Pancev,2013. "**Implementasi Penggunaan *Sensor Accelerometer* ADXL335 Pada *Quadcopter Robot* Berbasis Atmega32**". Teknik Elektro, Universitas Udayana

Muslim, Abadi. 2010."**Rancang Bangun Alat Pengukur Langkah Kaki dengan Sensor Accelerometer dan Fasilitas Komunikasi Wireless 2,4 GHz**". Teknik Telekomunikasi, PENS

BIODATA PENULIS

Nama Penulis Riza Farida, dilahirkan di Surabaya, 20 Maret 1995. Riwayat Pendidikan Penulis dimulai dari TK Putra Harapan dilanjutkan di SD Insan Mulya dan dilanjtukan lagi di SMPN 1 Driyorejo dan dilanjutkan lagi di SMAN 1 Driyorejo dan pada tahun 2013 masuk di prodi D3 Metrologi & Instrumentasi, Jurusan Teknik Fisika, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS) Surabaya dengan NRP : 24 13 031 028. Apabila terdapat pertanyaan tentang tugas akhir ini maka dapat menghubungi nomor telpon penulis yaitu : 085648084784, dan dapat juga melalui email penulis yaitu riza.fada@gmail.com