

II. TINJAUAN PUSTAKA

A. Sistem Akuisisi Data

Proses perubahan output yang belum diolah dari satu atau lebih sensor kedalam sinyal digital yang ekuivalen untuk dipakai pada proses lebih lanjut disebut dengan sistem akuisisi data. Mode akuisisi data merupakan tata cara pengiriman data dari suatu perangkat ke perangkat lainnya (Stallings, 2001). Terdapat dua cara dalam mode akuisisi yaitu dengan sinkron dan asinkron. Akuisisi sinkron adalah jenis akuisisi dimana kedua belah pihak, pengirim atau penerima berada pada waktu yang sinkron contoh dalam komputer. Sedangkan akuisisi asinkron merupakan akuisisi data dimana kedua belah pihak baik pengirim maupun penerima tidak perlu berada pada waktu yang sinkron. Seperti contoh prosesor kecepatannya lebih cepat dibanding dengan kecepatan mengetik seorang manusia (Ariyus dan Rumandri, 2008).

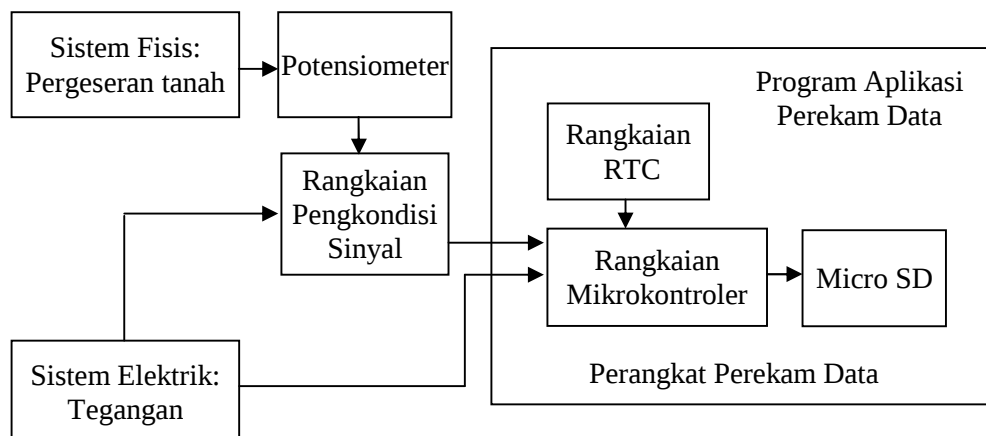
Berdasarkan arah pengirimannya akuisisi data dibedakan menjadi tiga (Bogart, 1992) yaitu:

1. *Simplex*: data dikirimkan hanya dalam satu arah saja. Data dari A (pengirim) dapat dikirimkan ke B (penerima), tetapi B tidak bisa mengirim data ke A. Seperti komunikasi pemancar TV dengan pesawat TV.
2. *Half duplex*: data dikirimkan dalam dua arah tetapi secara bersamaan. Pada saat A mengirim data B hanya menerima saja demikian juga sebaliknya. Contoh

komunikasi menggunakan *handy talkie* masing-masing perangkat yang digunakan untuk bisa bekerja secara bersamaan tetapi dilakukan secara bergantian.

3. *Full duplex*: data dikirimkan dalam dua arah secara bersamaan. Pada saat bersamaan antara A dan B dapat saling mengirim dan menerima data. Contoh yang seperti pesawat telepon yang dapat melakukan komunikasi dua arah secara bersamaan.

Sistem akuisisi data yang berbasis Micro SD secara umum dijelaskan pada Gambar 2.1.



Gambar 2.1 Akuisisi data pada sistem perekaman data.

Dari Gambar 2.1 dapat dijelaskan bahwa sebuah besaran fisis di alam dapat dirasakan oleh sebuah sensor atau transduser, dimana sinyal dari sensor tersebut diperkuat oleh pengkondisi sinyal, selanjutnya sinyal diubah menjadi sinyal data digital di port ADC pada mikrokontroler ATmega32 dengan tujuan agar sinyal tersebut dapat diolah untuk mengetahui hasilnya (Komarudin dkk, 2008) di dalam Micro SD yang merupakan salah satu instrumen yang bekerja dengan menggunakan data digital (Sunardi dkk, 2009).

B. Perekam Data

Perekam data merupakan alat berukuran kecil yang dapat dihubungkan dengan sejumlah sensor, yang dapat mencuplik sinyal sensor, mengubah sinyal tersebut dari bentuk analog ke dalam bentuk digital, mengolah sinyal digital berdasarkan kemauan pengguna, menyimpan data pada waktu yang telah ditentukan atau tergantung perintah eksternal serta mengirim data ke perangkat lain (Brock and Richardson, 2001). Perekam data dapat melakukan monitoring sekaligus penyimpanan data pengamatan dari suatu objek tertentu, dalam kurun waktu tertentu, dan dengan sampling data tertentu dari suatu *plant* (Sumiharto, 2010). Selain itu perekam data juga berfungsi untuk mengolah data input serta memberikan output berupa file data yang disimpan di dalam suatu memori (Santoso, 2012).

Perekam data secara umum terdiri dari mikrokontroler, sensor, dan media penyimpanan. Perekaman data dari sensor memiliki tujuan untuk pengarsipan atau tujuan analisis. Secara umum, perekam data ukuran fisiknya kecil, bertenaga baterai, portabel, perekam data biasanya dilengkapi dengan RTC didalamnya sehingga dapat ditampilkan waktu dan tanggal sampling untuk memastikan bahwa setiap data yang dicatat sesuai dengan tanggal dan waktu akuisisi dan dilengkapi dengan memori eksternal untuk menyimpan data dari sensor. Salah satu keuntungan menggunakan perekam data adalah kemampuannya secara otomatis mengumpulkan data (Withamana, 2009). Setelah diaktifkan, sistem perekam data dapat ditinggalkan untuk merekam informasi sehingga didapatkan gambaran yang komprehensif tentang kondisi lingkungan yang dipantau (Anonymous, 2012 A).

C. Sensor dan Extensometer

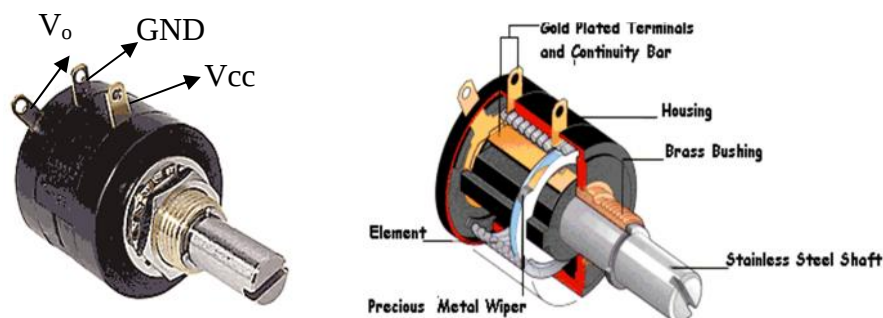
Sensor dan transduser merupakan peralatan atau komponen yang mempunyai peranan penting dalam sebuah sistem pengaturan otomatis. Ketepatan dan kesesuaian dalam memilih sebuah sensor akan sangat menentukan kinerja dari sistem pengaturan secara otomatis. Sensor adalah perangkat yang digunakan untuk mendeteksi fenomena fisik menjadi sinyal elektronik. Sensor merupakan peralatan atau komponen yang mempunyai peranan penting dalam sebuah sistem pengaturan otomatis. Ketepatan dan kesesuaian dalam memilih sebuah sensor akan sangat menentukan kinerja dari sistem pengaturan secara otomatis (Kenny, 2005).

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 (Taylor, 1997). 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 (Komarudin dkk, 2008).

Besaran masukan pada kebanyakan sistem kendali adalah bukan besaran listrik, seperti besaran fisika, kimia, mekanis dan sebagainya. Untuk memakaikan

besaran listrik pada sistem pengukuran, atau sistem manipulasi atau sistem pengontrolan, maka biasanya besaran yang bukan listrik diubah terlebih dahulu menjadi suatu sinyal listrik melalui sebuah alat yang disebut transduser. Transduser adalah sebuah alat yang bila digerakan oleh suatu energi di dalam sebuah sistem transmisi, akan menyalurkan energi tersebut dalam bentuk yang sama atau dalam bentuk yang berlainan ke sistem transmisi berikutnya. Transmisi energi ini bisa berupa listrik, mekanik, kimia, optik (radiasi) atau thermal (panas). Contoh: generator adalah transduser yang merubah energi mekanik menjadi energi listrik, motor adalah transduser yang merubah energi listrik menjadi energi mekanik, dan sebagainya (Talman, 1983).

Dalam penelitian ini sensor yang digunakan adalah potensiometer multiturn. Potensiometer masuk ke dalam kategori sensor mekanis karena mendeteksi perubahan gerak mekanis dan transduser daya dari luar (*external power transducer*) karena memerlukan sejumlah energi dari luar untuk menghasilkan suatu keluaran. Potensiometer multiturn memiliki keakuratan cukup baik terhadap perubahan hambatan serta dengan harga yang terjangkau memungkinkan untuk direalisasikan sensor pergeseran tanah yang bekerja secara digital (Kanchi and Gosala, 2010). Bentuk fisik dari potensiometer ditunjukkan oleh Gambar 2.2.



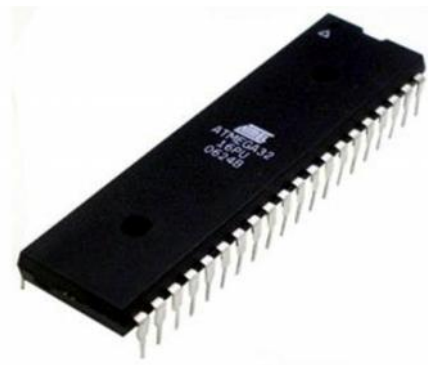
Gambar 2.2 Bentuk fisik potensiometer.

Potensiometer merupakan komponen resistor dengan tiga terminal yang tidak memiliki batas putaran pada kedua arahnya seperti terlihat pada Gambar 2.2. Jika ketiga terminal digunakan, potensiometer berfungsi sebagai rangkaian pembagi tegangan. Namun jika hanya dua terminal (terminal bagian tengah dan salah satu terminal bagian tepi) yang digunakan, potensiometer berfungsi sebagai variabel resistor atau rheostat (Chandra dan Arifianto, 2010). Potensiometer digunakan untuk mengubah gerak translasi atau anguler ke dalam suatu perubahan resistansi yang dapat langsung diubah menjadi sinyal tegangan atau arus listrik. Potensiometer terdiri dari sebuah kontak yang dapat menyapu pada hambatan lilitan kawat. Pergeseran kontak inilah yang menyebabkan terjadinya perubahan hambatan pada terminal-terminal kontak. Jika pada potensiometer dihubungkan dengan sebuah sumber tegangan maka perubahan hambatan tersebut menghasilkan perubahan tegangan keluaran (Tompkins and Webster, 1988).

Extensometer adalah sebuah perangkat yang digunakan untuk mendeteksi besar kecil pergeseran permukaan tanah. Extensometer berfungsi sebagai alat pendeteksi dan pengukur adanya pergerakan ataupun pergeseran permukaan tanah dalam orde millimeter. Hal ini berguna untuk pengukuran regangan dan pengujian tarik. Namanya berasal dari "**meter ekstensi**". Hal ini ditemukan oleh Dr Charles Huston yang dijelaskan dalam sebuah artikel di *Journal of Franklin Institute* pada tahun 1879. Huston kemudian memberikan hak untuk Fairbanks & Ewing, produsen utama mesin pengujian dan skala. Alat ini memanfaatkan perubahan resistansi dari potensiometer multiturn sehingga extensometer dapat digunakan untuk mengukur perpindahan yang sangat kecil akibat pembengkokan (*tensile stress*) atau peregangan (*tensile strain*).

D. Mikrokontroler ATmega32

Mikrokontroler adalah suatu chip yang dapat digunakan sebagai pengontrol utama sistem elektronika, di dalam chip tersebut sudah ada unit pemrosesan memori *Read Only Memory* (ROM), *Random Access Memory* (RAM), *Input-Output*, dan fasilitas pendukung lainnya (Budiharto, 2004) sehingga sangat memungkinkan untuk membentuk suatu sistem yang hanya terdiri dari *single chip* (keping tunggal) (Wardhana, 2006). Pada penelitian ini digunakan mikrokontroler ATmega32 yang merupakan mikrokontroler dengan arsitektur *Reduced Instruction Set Computing* (RISC) dengan lebar bus data 8 bit. Bentuk fisik mikrokontroler ATmega32 dapat dilihat pada Gambar 2.3.



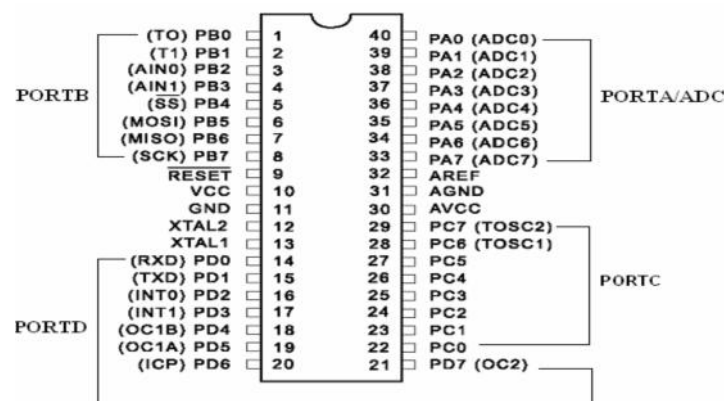
Gambar 2.3 Mikrokontroler ATmega32.

Frekuensi kerja mikrokontroler ini pada dasarnya sama dengan frekuensi osilator sehingga hal tersebut menyebabkan kecepatan kerja untuk frekuensi osilator yang sama akan dua belas kali lebih cepat dibandingkan dengan mikrokontroler keluarga AT89S51/52. ATmega 32 memiliki fitur ADC 10 bit, yang terhubung dengan 8 saluran Analog Multiplexer, sehingga memungkinkan untuk membangun sistem elektronika yang kompak. ADC mempunyai pin tegangan catu yang terpisah, yaitu AVCC. Referensi tegangan internal 2,56 volt atau AVCC

disediakan di dalam chip. ADC mengkonversi tegangan masukan analog ke nilai digital 10 bit melalui *successive approximation*. Nilai minimum adalah GND dan nilai maksimumnya adalah tegangan pada pin AREF dikurangi 1 LSB. Pada pin ADC terdapat rangkaian *Sample and Hold*, dimana tegangan input ADC ditahan dalam tingkat yang konstan pada saat konversi berlangsung. Kecepatan konversinya sekitar 65-260 μ s (Susilo, 2010).

1. Konfigurasi pin mikrokontroler ATmega32

Mikrokontroler memiliki beberapa port yang dapat digunakan sebagai *input/output* (I/O). Susunan kaki standar 40 pin DIP mikrokontroler AVR ATmega32 seperti Gambar 2.4.



Gambar 2.4 Susunan kaki ATmega32.

Pin pada mikrokontroler memiliki fungsi masing-masing yaitu:

- VCC merupakan pin masukan positif catu daya.
- GND sebagai pin Ground.
- AVCC sebagai pin masukan tegangan untuk ADC.
- AREF sebagai pin masukan tegangan referensi.
- Reset merupakan pin yang digunakan untuk me-reset mikrokontroler.

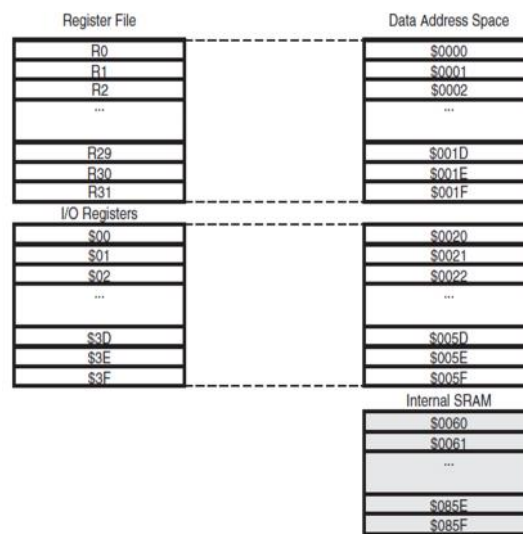
- f. Port A (PA0-PA7) merupakan pin I/O dua arah dan dapat diprogram sebagai pin masukan ADC.
- g. Port B (PB0-PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu Timer/Counter, komparator analog dan SPI.
- h. Port C (PC0-PC7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu TWI, komparator analog, dan *Timer Osilator*.
- i. Port D (PD0-PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analog, interupsi eksternal dan komunikasi serial.
- j. XTAL 1 dan XTAL 2 sebagai pin masukan clock eksternal. Sumber detak (*clock*) dibutuhkan oleh mikrokontroler agar dapat mengeksekusi instruksi yang ada di memori. Semakin tinggi nilai kristalnya, maka semakin cepat kerja mikrokontroler tersebut (Budiharto dan Rizal, 2007).

2. Peta memory ATmega32

Untuk penyimpanan data, mikrokontroler AVR menyediakan dua jenis memori yang berbeda, yaitu *Electrically Erasable Programmable Read Only Memory* (EEPROM) dan *Static Random Access Memory* (SRAM). EEPROM umumnya digunakan untuk menyimpan data-data program yang bersifat permanen, sedangkan SRAM digunakan untuk menyimpan data variabel yang dimungkinkan setiap saatnya. ATmega32 berisi 1024 byte memori data EEPROM atau memori yang dapat dapat ditulis dan dihapus secara elektrik. Memori ini diorganisasikan agar dapat diakses baca dan ditulis dalam satu byte.

SRAM adalah *space* kosong yang dapat kita gunakan sebagai tempat penyimpanan variabel, data, stack, dan keperluan lainnya. SRAM ini tidak

terhubung pada ALU sehingga untuk operasi yang menggunakan data pada SRAM harus melalui register umum R0-R31. Akan tetapi data pada SRAM dapat diakses secara *direct* (langsung) maupun *indirect* (melalui Pointer Register). Alamat \$085F adalah akhir dari alamat SRAM internal atau biasa disebut RAMEND. Peta memori ATmega32 seperti pada Gambar 2.5.



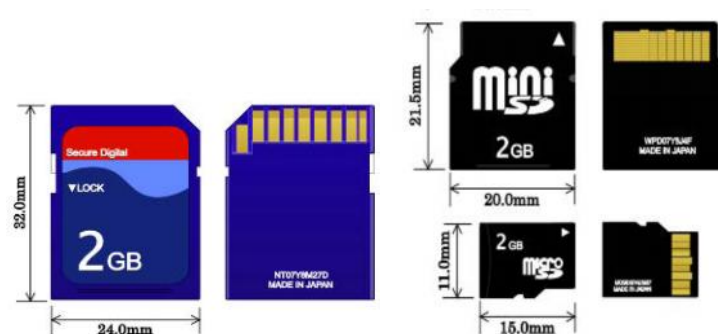
Gambar 2.5 Peta memori ATmega32.

Memori yang ada di dalam mikrokontroler ATmega32 ada beberapa macam, misalnya *General Working Register* (register kegunaan umum) sebanyak 32 byte. Register tersebut dinamai R1 sampai dengan R31 dari alamat \$0000 sampai \$001F. Untuk menyimpan data temporer, variabel lokal, dan alamat memori setelah terjadi interupsi dan subroutine biasanya digunakan Stack Pointer Register. Alamat Stack Pointer harus dimulai di atas \$0060. Register I/O, yaitu register yang berfungsi mengatur modul-modul pada mikrokontroler, menempati 64 alamat mulai dari \$0020 sampai dengan \$005F alamat berikutnya, yaitu \$0060 sampai \$085F sebesar 2 Kilobyte berfungsi sebagai SRAM internal. Selain itu ATmega32 memiliki *flash* PEROM, memori ini mempunyai kegunaan

menyimpan kode-kode instruksi dan merupakan memori dengan kapasitas terbesar di antara memori yang ada di dalam sebuah chip mikrokontroler. Memori program yang terletak pada memori jenis ini tersusun dalam 1 word atau 2 byte dengan lebar kode instruksi sebesar 16 byte atau 32 bit. ATmega32 memiliki 32 Kilobyte x 16 bit dengan alamat dari \$000 sampai dengan \$3FFF. Mode pengalamatan memori ini ditangani oleh Program Counter (PC) sebesar 12 bit. Untuk keamanan perangkat lunak, lahan memori Flash PEROM dibagi menjadi 2 bagian, bagian Boot Program dan bagian *Application Program* (Susilo, 2010).

E. 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 ditunjukkan Gambar 2.6.



Gambar 2.6 Bentuk fisik dan dimensi SD Card, Mini SD, dan MicroSD.

Ada tiga macam cara berkomunikasi dengan Micro SD, yaitu: 1. *One-bit SD mode*; 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 Allocation 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 (Sumiharto, 2010). 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:

$$\begin{aligned} \text{Total Sectors} &= \text{nt} \text{ Sectors} \text{ No S} \quad \text{atau} \\ \text{Total Sectors} &= \text{nt} \text{ L} \text{ No S} \\ \text{Clusters} &= \text{Total Sectors} - (\text{Data Region} - \text{Boot Sector}) / \text{Sector per Cluster} \end{aligned}$$

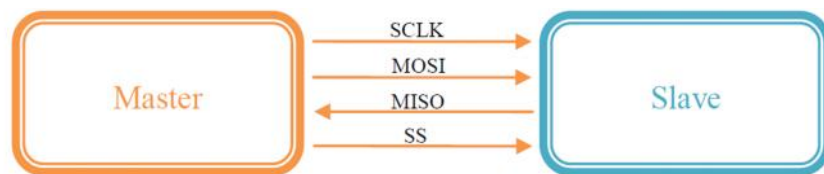
Jika clusters lebih kecil dari 4085 berarti FAT bertipe FAT12. Jika clusters lebih kecil dari 65525 berarti FAT bertipe FAT16. Di atas itu, FAT yang digunakan adalah FAT32. Format FAT16 menyerupai FAT12. Perbedaan yang paling jelas diantara keduanya adalah bahwa FAT16 menggunakan tabel FAT sebanyak 16 bit per alamat sedangkan FAT12 menggunakan 12 bit (Anonymous, 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 (Anonymous, 2011 C).

F. Serial Peripheral Interface (SPI)

SPI merupakan hubungan data serial yang standar untuk mikroprosesor, mikrokontroler dan peripheral yang dikeluarkan oleh perusahaan Motorola. Hubungan dalam SPI merupakan hubungan data serial yang *full-duplex*, *synchronous*. Sistem SPI cukup fleksibel sebagai antarmuka secara langsung dengan banyak *peripheral* yang tersedia. SPI dipakai untuk menyediakan komunikasi antara pengontrol dengan piranti peripheral. Komunikasi antara mikroprosesor dan peripheral atau *inter-processor* dapat dilakukan dengan SPI. Piranti SPI tersedia dari *shift register* sederhana untuk ADC, DAC, dan chip

memori. Pengontrol yang terintegrasi dengan port SPI menyediakan hubungan ke piranti peripheral dengan port SPI. SPI mampu mengirim data hingga kecepatan 3 MHz (Susilo, 2010). Skema SPI terdiri dari SPI *Master* dan SPI *Slave* seperti yang ditunjukkan Gambar 2.7.



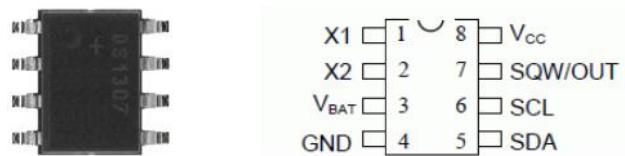
Gambar 2.7 Skema antarmuka komunikasi SPI.

Mode pengiriman data dengan SPI memerlukan 4 pin jalur data, yaitu *Serial Clock* (SCLK), *Master Output/Slave Input* (MOSI/SIMO), *Master Input/Slave Output* (MISO/SOMI) dan *Slave Select* (SS). *Device* yang dikomunikasikan menggunakan SPI dibedakan dalam master dan slave mode. Pada alat perekam data ini, SPI *Master* adalah Mikrokontroler dan SPI *Slave* adalah Micro SD. (Withamana, 2009).

G. 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 (Sumiharto, 2010). 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).

DS1307 tidak memiliki keterbatasan dalam penulisan ulang, tetapi DS1307 membutuhkan baterai untuk menyimpan data dan menjalankan jam. Apabila baterai tidak dipasang, maka semua data yang ada pada DS1307 akan hilang (Khoswanto dkk, 2004). Secara fisik DS1307 seperti pada Gambar 2.8.



Gambar 2.8 Bentuk fisik DS1307 beserta keterangan kaki-kaknya.

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 2.9.



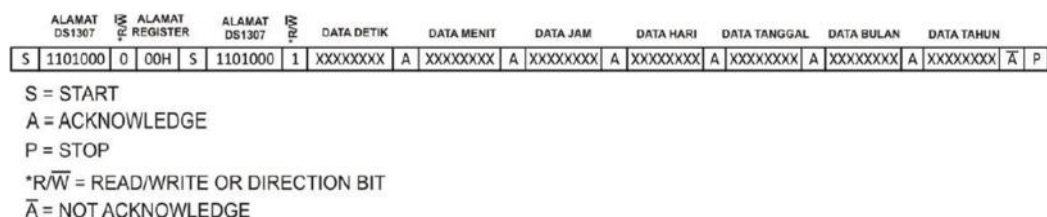
Gambar 2.9 Format pengiriman data jam ke DS1307.

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 2.10.



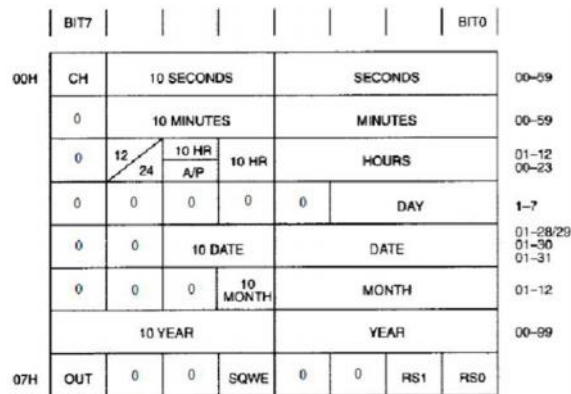
Gambar 2.10 Format pengiriman data tanggal ke DS1307.

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 2.11.



Gambar 2.11 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 2.12.



Gambar 2.12 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).

H. Liquid Crystal Display (LCD)

LCD adalah kristal cair pada layar yang digunakan sebagai tampilan dengan memanfaatkan listrik untuk mengubah-ubah bentuk kristal-kristal cairnya sehingga membentuk tampilan angka dan atau huruf pada layar. LCD sebagai modul penampil yang banyak digunakan karena tampilannya menarik. M1632 merupakan Modul LCD Matrix dengan konfigurasi 16 karakter dan 2 baris dengan setiap karakternya dibentuk oleh 8 baris pixel dan 5 kolom pixel (1 baris terakhir adalah kursor). Pada LCD tersebut terdapat Register Data dan Register Perintah. Proses akses data ke atau dari Register Data akan mengakses ke CGRAM, DDRAM, atau CGROM, bergantung pada kondisi *Address Counter* sedangkan

proses akses data ke atau dari Register Perintah akan mengakses ke *Instruction Decoder* dan menentukan perintah-perintah yang dilakukan LCD (Nalwan, 2004).

Untuk menampilkan karakter atau string ke LCD sangat mudah karena didukung pustaka yang telah disediakan oleh CodeVision AVR. Susunan alamat pada LCD seperti pada Gambar 2.13.

Display	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	...
Line 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	...
Line 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	...

Gambar 2.13 Susunan alamat pada LCD.

Penampil LCD sangat membantu dalam memprogram karena kita tidak menggunakan program debug. Kita perlu menampilkan hasil perhitungan, isi variabel atau keperluan debug lain ke LCD untuk mengetahui proses program yang kita buat LCD nantinya juga bisa untuk menampilkan hasil pengambilan data dari sensor, bahkan bisa untuk interaksi antara mikrokontroler dengan manusia (Heryanto dan Adi, 2008).

Rangkaian *interfacing* untuk LCD tidak banyak memerlukan komponen pendukung. Hanya diperlukan satu resistor dan satu variabel resistor untuk memberi tegangan kontras pada matriks LCD. Modul LCD ini membutuhkan daya yang kecil dan dilengkapi dengan panel LCD dengan tingkat kontras yang cukup tinggi, serta pengendali LCD CMOS yang terpasang dalam modul tersebut. Pengendali mempunyai pembangkit karakter ROM /RAM dan *display* data RAM. Semua fungsi *display* diatur oleh instruksi-instruksi, sehingga modul LCD ini dapat dengan mudah dihubungkan dengan unit mikroprosesor (Wardhana, 2006).

Display karakter pada LCD diatur oleh pin EN, RS dan RW:

1. Jalur EN dinamakan Enable. Jalur ini digunakan untuk memberitahu LCD bahwa anda sedang mengirimkan sebuah data. Untuk mengirimkan data ke LCD, maka melalui program EN harus dibuat logika low “0” dan set pada dua jalur kontrol yang lain RS dan RW. Ketika dua jalur yang lain telah siap, set EN dengan logika “1” dan tunggu untuk sejumlah waktu tertentu (sesuai dengan datasheet dari LCD tersebut) dan display karakter pada LCD diatur oleh pin EN, RS dan RW berikutnya set EN ke logika low “0” lagi.
2. Jalur RS adalah jalur Register Select. Ketika RS berlogika low “0”, data akan dianggap sebagai sebuah perintah atau instruksi khusus (seperti *clear screen*, posisi kursor dan lain-lain). Ketika RS berlogika high “1”, data yang dikirim adalah data text yang akan ditampilkan pada display LCD.
3. Jalur RW adalah jalur kontrol *Read/Write*. Ketika RW berlogika *low* (0), maka informasi pada bus data akan dituliskan pada layar LCD. Ketika RW berlogika *high* ”1”, maka program akan melakukan pembacaan memori dari LCD. Sedangkan pada aplikasi umum pin RW selalu diberi logika low ”0”. Pada akhirnya, bus data terdiri dari 4 atau 8 jalur (bergantung pada mode operasi yang dipilih oleh *user*). Pada kasus bus data 8 bit, jalur diacukan sebagai DB0 s/d DB7 (Budiharto dan Rizal, 2007).

I. Perangkat Lunak

Perangkat lunak merupakan faktor penting dalam tahap perancangan sistem perekam data. Perangkat lunak ini merupakan algoritma dalam bentuk *listing* program yang ditanamkan kedalam mikrokontroler.

1. Pemrograman bahasa C

Bahasa pemrograman C merupakan bahasa level atas yang paling banyak dipakai untuk mikrokontroler. Hal tersebut karena Bahasa C mempunyai kemampuan lebih dari bahasa pemrograman yang lain. Bahasa C merupakan bahasa yang sifatnya portabel yaitu dengan sedikit (tanpa perubahan) satu program yang ditulis dengan bahasa C pada suatu komputer dapat dijalankan pada komputer lain. Keunggulan lain dari bahasa C adalah kecepatan prosesnya 50 kali lebih cepat dibandingkan dengan program BASIC. Kebaikan Bahasa C yang lain adalah program yang ditulis itu akan teratur dengan baik (Nugroho, 1994).

Dalam penulisan program C terbagi dua kategori yaitu deklarasi dan pernyataan (*statement*). Deklarasi adalah membuat dan memberitahu kepada *compiler* tentang sesuatu yang digunakan nanti dalam penulisan program agar digunakan semestinya dan tidak dianggap *error* atau asing. Sedangkan pernyataan adalah membuat instruksi-instruksi program dengan menggunakan *keyword* seperti instruksi operasi aritmatika, logika, operasi bit, atau instruksi percabangan dan *looping*, atau pembuatan fungsi (Winoto, 2010).

Berikut adalah penjelasan dasar-dasar dari pemrograman bahasa C:

a. Identifier

Identifier adalah nama yang dipakai untuk mengacu pada sesuatu (misalnya variabel, konstanta, fungsi, dan lain-lain). Nama yang kita pilih tidak ada artinya bagi komputer karena itu pilihlah nama yang ada artinya bagi kita agar mudah dimengerti. Nama identifier bersifat *case sensitive* sehingga Bandung berbeda dengan bandung.

b. Tipe data

Tipe data menunjukkan bagaimana data itu direpresentasikan serta diproses oleh komputer. Tipe data yang dasar terdiri dari int, float, char dan bool. Tipe-tipe data yang ada dalam bahasa C dan yang dikenali oleh Code Vision AVR dijelaskan dalam Tabel 2.1.

Tabel 2.1 Tipe-tipe data dalam bahasa C

No	Tipe Data	Ukuran	Jangkauan Nilai
1	Bit	1 bit	0 atau 1
2	Char	1 byte	-128 s/d 225
3	Unsigned Char	1 byte	0 s/d 225
4	Signed Char	1 byte	-128 s/d 127
5	Int	2 byte	-32.768 s/d 32.767
6	Short Int	2 byte	-32.768 s/d 32.767
7	Unsigned Int	2 byte	0 s/d 65.535
8	Signed Int	2 byte	-32.768 s/d 32.767
9	Long Int	4 byte	-2.147.483.648 s/d 2.147.483.647
10	Unsigned Long Int	4 byte	0 s/d 4.294.967.295
11	Signed Long Int	4 byte	-2.147.483.648 s/d 2.147.483.647
12	Float	4 byte	$1.2 \cdot 10^{-38}$ s/d $3.4 \cdot 10^{+38}$
13	Double	4 byte	$1.2 \cdot 10^{-38}$ s/d $3.4 \cdot 10^{+38}$

Pada Penelitian ini menggunakan tipe data float. Tipe data float dideklarasikan dengan menggunakan *keyword float*, *double* dan *long double*. Tipe data float dipakai untuk merepresentasikan bilangan riil (misalnya 19.0, 153.24). Bilangan riil direpresentasikan sebagai pendekatan oleh memori komputer. Alokasi memori yang digunakan tipe data float tergantung pada presisi yang digunakan yang terdiri dari:

- *Single precision (float)*: 4 byte dan 6 digit desimal
- *Double precision (double)*: 8 bytes and 16 digit desimal
- *Long double*: 10 byte and 20 digit desimal (Nugraha, 2010).

c. Komentar

Komentar digunakan untuk memberikan penjelasan, informasi ataupun keterangan-keterangan yang dapat membantu mempermudah dalam memahami kode program baik bagi pembuat program maupun bagi orang lain yang membacanya. Komentar yang hanya satu baris ditulis dengan diawali `'//'` sedangkan komentar yang lebih dari satu baris diawali dengan `'/*'` dan diakhiri dengan `'*/'`. Selain digunakan untuk memberikan keterangan program, komentar juga dapat digunakan untuk membantu dalam pengujian program yaitu dengan menon-aktifkan dan mengaktifkan kembali bagian program tertentu selama proses pengujian. Komentar tidak akan dikompilasi (diabaikan) oleh *compiler* (Bejo, 2008).

d. Pernyataan

Pernyataan merupakan satu buah instruksi lengkap yang dapat berdiri sendiri. Pernyataan If digunakan untuk melakukan pengambilan keputusan terhadap dua buah kemungkinan yaitu mengerjakan suatu blok pernyataan atau tidak.

Bentuk pernyataan If adalah :

```

if (kondisi)
{ // blok pernyataan yang akan dikerjakan
  // jika kondisi if terpenuhi
}

```

Contoh : **if** (PINA>0x80)

```

{
  Dataku=PINA;
  PORTC=0xFF;
}

```

Pernyataan if diatas akan mengecek apakah data yang terbaca pada Port A (PINA) nilainya lebih dari 0x80 atau tidak, jika ya maka variabel dataku diisi

dengan nilai PINA dan data 0xFF dikeluarkan ke PORT C. Apabila dalam blok pernyataan hanya terdapat satu pernyataan saja maka tanda { dan } dapat dihilangkan seperti:

```
if (PINA>0x80) PORTC=0xFF;
```

e. Konstanta dan Variabel

Variabel adalah tempat untuk menyimpan dan mengakses data yang mewakili memori dalam mikrokontroler. Variabel harus dideklarasikan (memberitahu kompiler) dengan tipe data beserta nama variabel yang akan digunakan. Selain harus punya nama, variabel juga harus dideklarasikan sebelum dapat digunakan. Konstanta adalah identifier yang merepresentasikan lokasi di memori komputer dimana isi lokasi tersebut tidak dapat diubah (variabel yang mempunyai nilai tetap). Pendeklarasian konstanta dapat dilakukan dengan 2 cara. Cara yang pertama berasal dari bahasa C menggunakan preprocessor directive `#define`. Cara kedua adalah dimulai dengan keyword *const* dan tipe data lalu nama konstanta beserta harga konstanta tersebut (Joni dan Raharjo, 2011).

2. Code Vision AVR

Pada pembuatan program mikrokontroler, memerlukan suatu sistem program untuk menempatkan dan mengirim program dari PC ke mikrokontroler. Sistem program pendukung yang digunakan adalah Code Vision AVR. Code Vision AVR merupakan salah satu program bahasa C untuk AVR yang berbasis Windows, keuntungan menggunakan code vision AVR lebih besar dibandingkan menggunakan program yang lain yang under DOS. Kompiler ini cukup memadai

untuk belajar AVR, karena selain mudah penggunaannya juga didukung berbagai fitur yang sangat membantu dalam pembuatan *software* untuk keperluan pemrograman AVR. Code vision AVR menyediakan area kerja dan toolbar yang mudah untuk melakukan berbagai operasi. Tampilan Code Vision AVR seperti pada Gambar 2.14.



Gambar 2.14 Tampilan Code Vision AVR.

Code Vision AVR memiliki beberapa menu aplikasi windows yaitu meliputi *File, Project, Edit, Debug, View, Tool, Windows, Help, check syntak eror compile* dan *make*. Ada perbedaan pada menu Code Vision AVR yaitu, ketika belum ada satu pun project atau file yang dibuka, Code Vision AVR hanya memiliki enam menu yaitu File, Project, View, Tools, Debug dan Help, namun jika sebuah project atau file telah dibuka dan sedang dikerjakan, maka akan terdapat dua menu tambahan yaitu Edit dan Windows (Haryanto, 2005).

Code Vision AVR ini dapat mengimplementasikan hampir semua instruksi bahasa yang sesuai dengan arsitektur AVR, bahkan terdapat beberapa keunggulan tambahan untuk memenuhi keunggulan spesifik dari AVR. Hasil kompilasi objek Code Vision AVR bisa digunakan sebagai source debug dengan AVR studio debugger dari ATMEL. Selain pustaka standar bahasa, Code vision AVR juga

menyediakan pustaka tambahan yang sangat membantu pemrograman AVR, yaitu Alphanumeric LCD modules, Philips I2C bus, National Semiconductor LM75 Temperature Sensor, Philips PCF8583, Maxim/Dallas Semiconductor DS1302 and DS1307 Real Time Clocks, Maxim/Dallas Semiconductor 1 Wire protocol, Maxim/Dallas Semiconductor DS1820, DS18S20, DS18B20 Temperature Sensor, Maxim/Dallas Semiconductor DS1621 Thermometer/Thermostat, Maxim/Dallas Semiconductor DS2430 and DS2433 EEPROMs, SPI, MMC, Power management, Delays, dan Gray code conversion. Code Vision AVR juga memiliki program generator yang memungkinkan kita membuat program dengan cepat (Heryanto dan Adi, 2008).