BAB II
LANDASAN TEORI

Landasan teori ini memuat teori-teori pendukung dalam perancangan perangkat keras dan perangkat lunak yang terkait dengan pembuatan tugas akhir penulis.

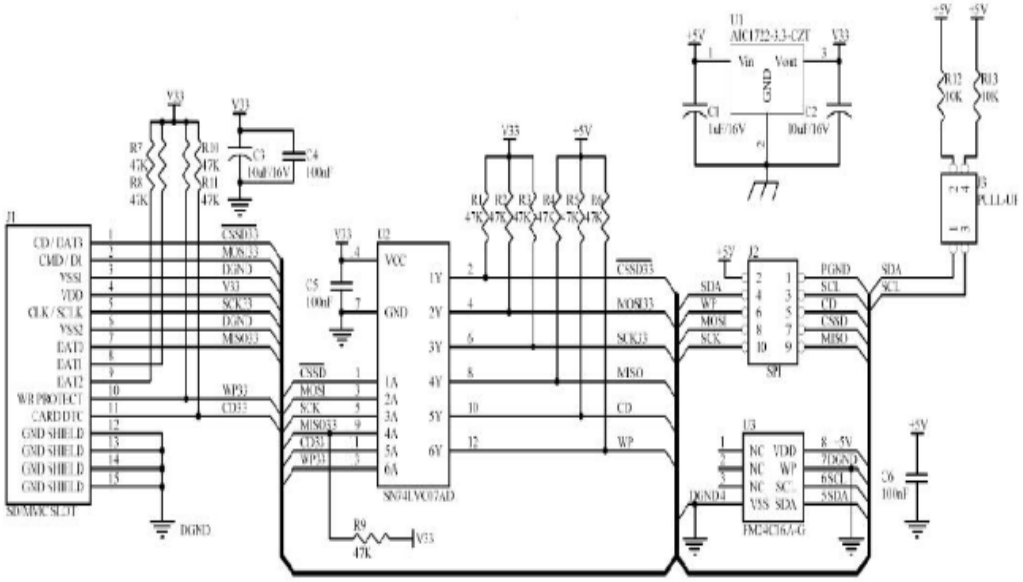
2.1 Perangkat Keras (*Hardware*)

Perangkat keras ini memuat teori-teori pendukung dalam pembuatan perangkat keras tugas akhir penulis yang menggunakan beberapa perangkat keras,

yaitu modul MMC (*Multi Media Card*) yang berfungsi sebagai antarmuka antara MMC dengan mikrokontroler, mikrokontroler ATMega32 berfungsi sebagai pengolah data, Tombol berfungsi sebagai masukan yang akan diolah oleh Mirokontroler, dan LCD (*Liquid Crystal Display*) berfungsi sebagai keluaran yang berupa text.

2.1.1 Modul MMC

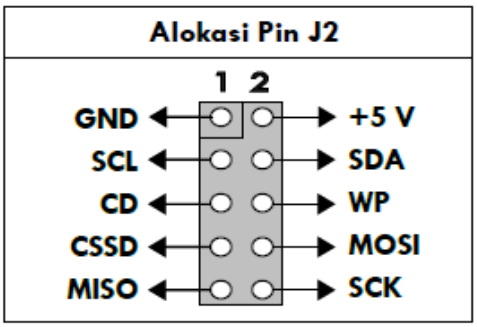
EMS (Embedded Module Series) SD/MMC/FRAM merupakan suatu modul untuk mempermudah antar muka antara MMC dan mikrokontroler dengan tegangan kerja +5 VDC. MMC dapat digunakan sebagai memori yang dapat diganti dengan mudah sehingga memudahkan dalam ekspansi ke kapasitas memori yang lebih besar. Tersedia *Ferroelectric Nonvolatile RAM* (FRAM) yang dapat digunakan sebagai *buffer* sementara dalam mengakses MMC atau sebagai tempat penyimpan data lain. Modul ini dapat digunakan antara lain sebagai penyimpan data pada sistem absensi, sistem antrian, atau aplikasi data *logging* lainnya. [1]



Gambar 2.1 Skema Rangkaian Module EMS SD/MMC/FRAM. [1]

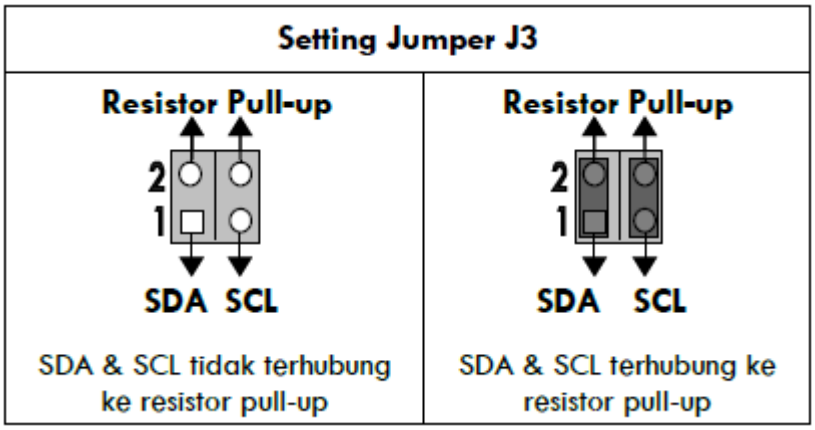
Pada gambar 2.1 merupakan skema rangkaian *EMS SD/MMC/FRAM* dimana modul ini bekerja pada tegangan supply +5 V DC. Jenis kartu yang didukung adalah MMC. Antarmuka MMC dengan mikrokontroler secara SPI dan tersedia 2 Kbyte *Ferroelectric Nonvolatile* RAM sedangkan Antarmuka FRAM dengan mikrokontroler secara *Two Wire Interface.*

Two Wire Interface atau sering disebut juga I2C (*Inter Integrated Cirkuit*) merupakan protokol atau aturan untuk komunikasi serial antar IC.



Gambar 2.2 Alokasi pin J2 pada module EMS SD/MMC/FRAM.[1]

Gambar 2.2 yaitu alokasi pin J2 pada *EMS SD/MMC/FRAM* berfungsi untuk akses data *EMS SD/MMC/FRAM* dengan mikrokontroler ATMega32 agar dapat berkomunikasi.



Gambar 2.3 Setting Jumper J3 pada module EMS SD/MMC/FRAM. [1]

Gambar 2.3 yaitu pengaturan jumper J3 yang digunakan untuk resistor *pull-up* SDA dan SCL. Apabila modul terhubung kejaringan *Two-Wire Interface*, maka dalam satu jaringan tersebut hanya perlu memasang *pull-up* pada salah satu modul saja.

Tabel 2.1 merupakan keterangan dan kegunaan masing-masing pin pada *EMS SD/MMC/FRAM*.

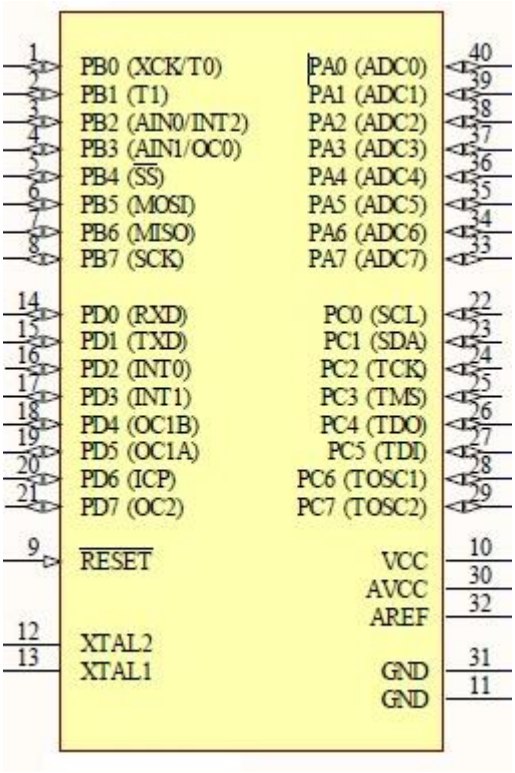
Tabel 2.1 Keterangan Pin Pada *Module EMS SD/MMC/FRAM. [1]*

Pin	**Nama**	**Fungsi Pada Module**	**Keterangan**
1	GND	*Input*	Referensi *Ground*
2	+5 V	*Input*	Terhubung ke Sumber Tegangan+5 VDC
3	SCL	*Input*	Serial *Clock* untuk akses *FRAM*
4	SDA	*Input /Output*	Serial Data untuk transaksi data dari / ke *FRAM*
5	CD	*Output*	*Card Detect*, berlogika 0 jika ada kartu yang dimasukkan, berlogika 1 jika tidak ada kartu
6	WP	*Output*	*Write Protect*, berlogika 0 jika saklar pada MMC tidak berada pada posisi dikunci, berlogika 1 jika *MMC* dalam posisi dikunci
7	CSSD	*Input*	*Chip Select*, diberi logika 0 untuk mengakses *MMC*, diberi logika 1 jika tidak mengakses *MMC*
8	MOSI	*Input*	Jalur data masuk ke *MMC*
9	MISO	*Output*	Jalur data keluar dari *MMC*
10	SCK	*Input*	Jalur *clock* dari mikrokontroler untuk mengakses *MMC*

2.1.2 Mikrokontroler ATMega32

Mikrokontroler ATMega32 mempunyai *spesifikasi* diantaranya ukuran memori *flash* 16 KB, SRAM sebesar 1 KB dan EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 512 *byte*, dan portal komunikasi serial USART (*Universal Asynchronous Receiver-Transmitter*) yang memudahkan untuk komunikasi serial. [2]

USART (*Universal Asynchrinous Receiver-Transmitter*) merupakan komunikasi yang memiliki *fleksibilitas* tinggi, yang dapat digunakan untuk melakukan pengiriman data baik antar mikrokontroler maupun dengan modul-modul *eksternal* termasuk PC.



Gambar 2.4 Mikrokontroler ATMega32.

Gambar 2.4 merupakan mikrokontroler ATMega32 yang berfungsi sebagai pengolah data dan digunakan untuk mengolah masukan dari tombol serta memberikan keluaran.

Tabel 2.2 di bawah ini adalah keterangan dan kegunaan dari masing-masing pin ATMega32.

Tabel 2.2 Keterangan Pin ATMega32. [2]

No. Pin	Nama	Fungsi
1	PB0(XCK/T0)	Port *B.0/Counter* 0/*clock eksternal* untuk *USART (XCK)*
2	PB1 (T1)	Port B.1/Counter 1
3	PB2 (INT2/AIN0)	Port B.2/Input (+) Analog Komparator (AIN0) dan interupsi *eksternal* 2 (INT2)
4	PB3 (0C0/AIN1)	Port *B.3/Input* (-)Analog Komparator (AIN1) dan *output* PWM0
5	PB4 (SS)	Port B.4/ SPI *Slave Select Input* (SS)
6	PB5 (MOSI)	Port B.5/ SPI bus *Master Out Slave Input*
7	PB6 (MISO)	Port B.6/ SPI bus *Master Input Slave Output*
8	PB7 (SCK)	Port B.7/ sinyal *clock serial* SPI
9	RESET	Me-reset mikrokontroller
10	VCC	Catu daya (+)
11	GND	Sinyal *Ground* terhadap Catu daya
12 - 13	XTAL 1-XTAL 2	Sinyal input *Clock eksternal* (kristal)
14	PD0 (RXD)	Port D.0/ Penerima data serial
15	PD1 (TXD)	Port D.l/ Pengirim data serial
16	PD2 (INT0)	Port D.2/ Interupsi *eksternal* 0
17	PD3 (INT1)	Port D.3/ interupsi *eksternal* 1
18	PD4 (0C1B)	Port D.4/ Pembanding *timer-counter* 1
19	PD5 (0C1A	Port D.5/ *Output* PWM 1A
20	PD6 (ICP1)	Port D.6/ *Timer-counter* 1 *input*
21	PD7 (0C2)	Port D.7/ *Output* PWM 2
22	PC0 (SCL)	Port C.0/ *Serial bus clock line*
23	PC1 (SDA)	Port C.1/ *Serial bus* data *input-output*
24 - 27	PC2-PC5	Port C.2 - C.5
28	PC6 (T0SC1)	Port C.6/ *Timer osilator* 1
29	PC7 (T0SC2)	Port C.7/ *Timer osilator* 2
30	AVCC	Tegangan ADC

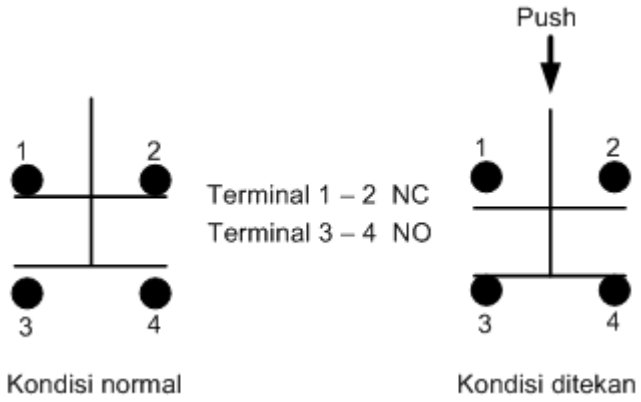
2.1.3 Tombol

Tombol (*Push button*) adalah perangkat/saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan *unlock* (tidak mengunci). Sistem kerja *unlock* disini berarti saklar akan bekerja sebagai perangkat penghubung atau pemutus aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan (dilepas), maka saklar akan kembali pada kondisi normal.



Gambar 2.5 Tombol.

Gambar 2.5 merupakan tampilan dari tombol. Sebagai perangkat penghubung atau pemutus, *push button switch* hanya memiliki 2 kondisi, yaitu *On* dan *Off* (1 dan 0). Istilah *On* dan *Off* ini menjadi sangat penting karena semua perangkat listrik yang memerlukan sumber energi listrik pasti membutuhkan kondisi *On* dan *Off*.



Gambar 2.6 prinsip kerja tombol

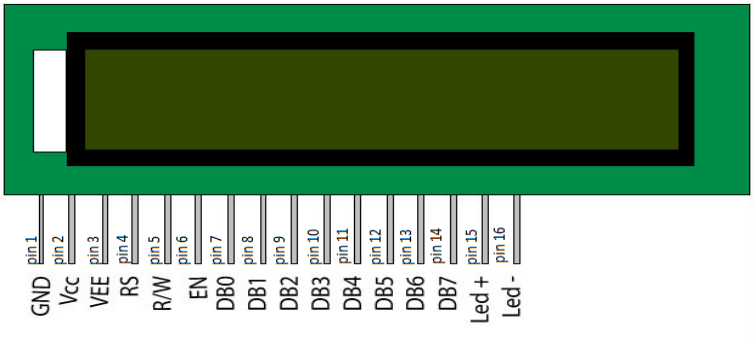
Gambar 2.6 di atas ini adalah prinsip kerja tombol. Berdasarkan fungsi kerjanya yang menghubungkan dan memutuskan, tombol mempunyai 2 tipe kontak yaitu NC *(Normally Close*) dan NO (*Normally Open*).

1. NO (*Normally Open*), merupakan kontak terminal dimana kondisi normalnya terbuka (aliran arus listrik tidak mengalir), Dan ketika tombol saklar ditekan, kontak yang NO ini akan menjadi menutup (*Close*) dan mengalirkan atau menghubungkan arus listrik. Kontak NO digunakan sebagai penghubung atau menyalakan sistem circuit (*Push Button ON*).

2. NC (Normally Close), merupakan kontak terminal dimana kondisi normalnya tertutup (mengalirkan arus litrik). Dan ketika tombol saklar push button ditekan, kontak NC ini akan menjadi membuka (*Open*), sehingga memutus aliran arus listrik. Kontak NC digunakan sebagai pemutus atau mematikan sistem circuit (*Push Button Off*). [3]

2.1.4 LCD (Liquid Crystal Display)

LCD (*Liquid Crystal Display*) adalah modul penampil yang banyak digunakan karena tampilannya menarik. LCD yang paling banyak digunakan saat ini adalah tipe LCD 16x2 karena hargaya yang cukup murah. LCD 16x2 merupakan modul dengan tampilan 2x16 (2 baris x 16 kolom) dengan menggunakan daya rendah. Modul tersebut dilengkapi dengan mikrokontroler yang didesain khusus untuk mengendalikan LCD.



Gambar 2.7 LCD.

Gambar 2.7 merupakan gambar LCD yang berfungsi sebagai penampil hasil masukan pengguna yang dapat digunakan untuk memeriksa apakah masukan yang diberikan benar atau salah. Memiliki jumlah penampil sebanyak 32 karakter dengan 16 tertampil pada setiap barisnya. Mikrokontroler mengakses LCD menggunakan mode pengaturan 4 jalur data, sesuai dengan pustaka yang tersedia pada CodeVisionAVR. Tabel 2.3 adalah operasi dasar pada LCD.

Tabel 2.3 Operasi Dasar LCD. [4]

RS	R/W	Operasi
0	0	Input Instruksi ke LCD
0	1	Membaca Status *flag* (DB7) dan alamat *counter* (DB0 ke DB6)
1	0	Menulis Data
1	1	Membaca Data

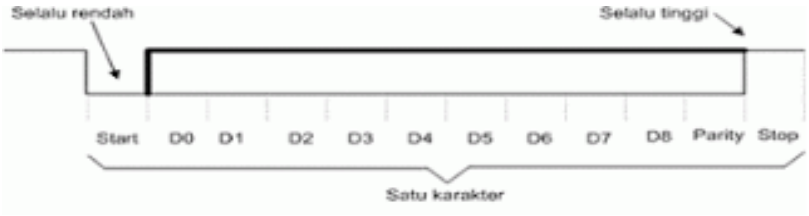
Tabel 2.4 di bawah ini adalah keterangan dan *konfigurasi* pada pin LCD.

Tabel 2.4 Konfigurai pin LCD. [4]

Pin No.	Keterangan	Konfigurasi Hubung
1	GND	Ground
2	VCC	Tegangan +5VDC
3	VEE	Ground
4	RS	Kendali RS
5	RW	Ground
6	E	Kendali E/Enable
7	D0	Bit 0
8	D1	Bit 1
9	D2	Bit 2
10	D3	Bit 3
11	D4	Bit 4
12	D5	Bit 5
13	D6	Bit 6
14	D7	Bit 7
15	A	Anoda (+5VDC)
16	K	Katoda (Ground)

2.1.5 Komunikasi Serial

Transmisi data serial dibedakan menjadi 2 macam, yaitu komunikasi data serial sinkron dan komunikasi data serial asinkron, perbedaan ini tergantung pada *clock* pendorong data. Dalam komunikasi serial data serial sinkron, *clock* untuk register geser ikut dikirimkan bersama dengan data serial. Sebaliknya dalam komunikasi data serial asinkron, *clock* pendorong register geser tidak ikut dikirim, rangkaian penerima data harus dilengkapi dengan rangkaian yang mampu membangkitkan *clock* yang di perlukan. Bagian yang terpenting dari komunikasi serial asinkron adalah upaya agar penerima data bisa membangkitkan clock yang bisa dipakai untuk mendorong register geser penerima. Untuk keperluan tersebut terlebih dulu ditentukan bahwa saat tidak ada pengiriman data, keadaan saluran adalah ‘1’. Saat akan mulai mengirim data 1 byte saluran dibuat menjadi ‘0’ dulu selama 1 periode *clock* pendorong, dalam 8 periode clock berikutnya dikirim data bit 0, bit 1, dan bit seterusnya sampai bit 8, dan pada periode clock yang ke 10 saluran dikembalikan menjadi ‘1’. Dengan demikian, data 8 bit yang dikirim diawali dengan bit start yang bernilai ‘0’ dan diakhiri dengan bit stop yang bernilai ‘1’ seperti yang terlihan pada gambar di bawah ini.



Gambar 2.8 Komunikasi Serial Asinkron.

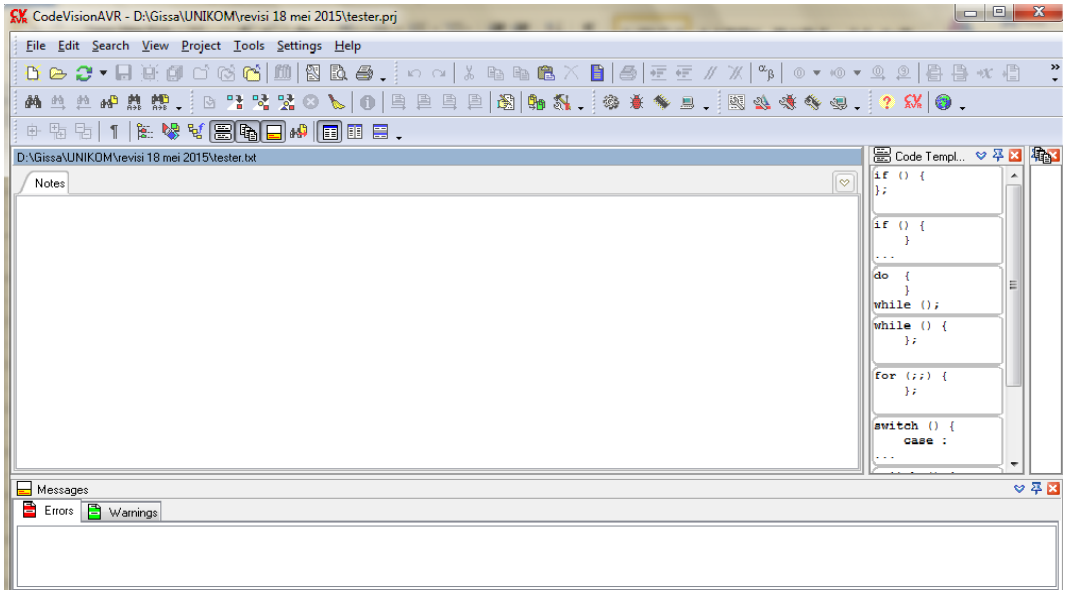
Gambar 2.8 merupakan kemasan data yang dimaksud agar rangkaian penerima bisa membangkitkan clock yang frekuensinya sama dengan clock pengirim dan fasanya di sinkronkan pada awal penerimaan data 1 *byte*.[5]

2.2 Perangkat Lunak (*software*)

Pengembangan sebuah sistem yang menggunakan mikrokontroler AVR buatan ATMEL, menggnakan perangkat lnak AVRStudio dan CodeVisionAVR. AVRStudio merupakan perangkat lunak khusus untuk bahasa *assembly* yang mempunyai fungsi sangat lengkap, yaitu digunakan untuk menulis program, kompilasi, simulasi dan *download* program ke IC mikrokontroler AVR.

CodeVisionAVR merupakan perangkat lunak *C-cross compiler*, dimana penulisan program dapat ditulis dalam bahasa C, CodeVisionAVR memiliki IDE (*Integrated Development Environment*) yang lengkap, dimana penulisan program, *compile*, *link*, pembuatan kode mesin (*assembler)* dan *download* program ke chip AVR dapat dilakukan pada CodeVisionAVR, selain itu ada fasilitas terminal, yaitu untuk melakukan komunikasi serial pada mikrokontroler yang sudah di program. Proses *download* program ke IC mikrokontroler AVR dapat menggunakan sistem *download* secara ISP (*In-System Programing*). *In-System Programmable Flash on chip* mengizinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI.[5]

Gambar 2.9 di bawah ini merupakan tampilan utama dari CodeVisionAVR.



Gambar 2.9 Tampilan dari CodeVisionAVR.

Tabel 2.5 di bawah ini merupakan instruksi-instruksi dasar yang diusung oleh kompiler C, baik itu untuk pemrograman mikrokontroler, maupun pemrograman *software* komputer.

Tabel 2.5 Instruksi dasar bahasa pemrograman C.

Instruksi	**Keterangan**
While (kondisi) { … }	Perulangan
If (kondisi) { … }	Percabangan
Switch(item) {case(kondisi) : … Break;}	Percabangan
Var = item;	Penugasan/assignment
getchar();	Input data
putchar();	Output data