

BAB 11

TINJAUAN PUSTAKA

2.1. *Radio Frequency Identification (RFID)*

RFID (*Radio Frequency Identification*) adalah proses identifikasi seseorang atau objek dengan menggunakan frekuensi transmisi radio dan teknologi wireless dengan frekuensi kerja antara 125 KHz sampai 2.5 GHz. RFID menggunakan frekuensi radio untuk membaca informasi dari sebuah alat kecil yang disebut *tag* atau *transponder* (*Transmitter+Responder*). *Tag* RFID akan mengenali diri sendiri ketika mendeteksi sinyal dari alat yang kompatibel, yaitu pembaca RFID (*RFID Reader*). RFID adalah teknologi identifikasi yang fleksibel, mudah digunakan, dan sangat cocok untuk operasi otomatis. RFID mengkombinasikan keunggulan yang tidak tersedia pada teknologi identifikasi yang lain. RFID dapat disediakan dalam perangkat yang hanya dapat dibaca saja (*Read Only*) atau dapat dibaca dan ditulis (*Read/Write*), tidak memerlukan kontak langsung maupun jalur cahaya untuk dapat beroperasi, dapat berfungsi pada berbagai variasi kondisi lingkungan, dan menyediakan tingkat integritas data yang tinggi. Sebagai tambahan, karena teknologi ini sulit untuk dipalsukan, maka RFID dapat menyediakan tingkat keamanan yang tinggi.

2.1.1. Sejarah RFID

Beberapa orang berpendapat bahwa device pertama ditemukan oleh Leon Theremin sebagai suatu *tool spionase* untuk pemerintahan Rusia

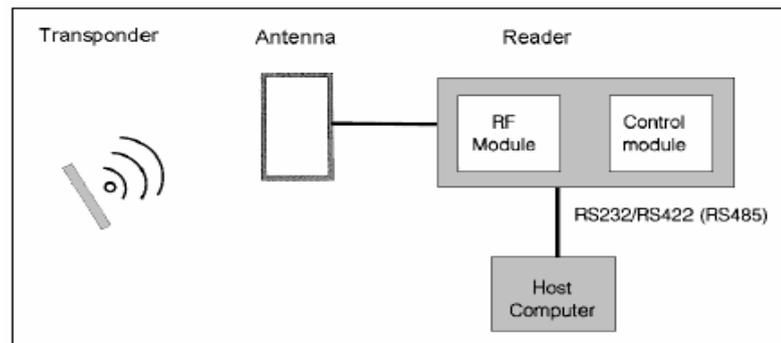
sekitar tahun 1945 adalah cikal bakal RFID. Hal ini tidak benar sepenuhnya karena alat Theremin ini sebenarnya suatu alat pendengar yang pasif dan bukan merupakan suatu *identification tag*. Cikal bakal teknologi RFID sudah ada sejak tahun 1920-an dan dikembangkan pada masa Perang Dunia II untuk mengenali pesawat udara musuh atau teman. Suatu teknologi yang lebih dekat dengan RFID, yang dinamakan IFF (*Identity Friend or Foe*) transponder, beroperasi pada tahun 1939 dan digunakan oleh Inggris pada Perang Dunia II untuk mengenali pesawat udara musuh atau teman. Ketika pesawat Inggris tertangkap oleh radar, maka pesawat itu akan memancarkan sinyal ke radar yang mengidentifikasi bahwa pesawat tersebut adalah teman. IFF ini sudah menggunakan prinsip dasar teknologi RFID.

2.1.2. Sistem RFID

Pada sistem RFID umumnya, *tag* atau *transponder* ditempelkan pada suatu objek. Setiap *tag* membawa dapat membawa informasi yang unik, di antaranya: serial number, model, warna, tempat perakitan, dan data lain dari objek tersebut. Ketika *tag* ini melalui medan yang dihasilkan oleh pembaca RFID yang kompatibel, *tag* akan mentransmisikan informasi yang ada pada *tag* kepada pembaca RFID, sehingga proses identifikasi objek dapat dilakukan.

Sistem RFID terdiri dari empat komponen, di antaranya seperti dapat dilihat pada gambar 2.1, yaitu :

1. *Tag*: Ini adalah alat yang menyimpan informasi untuk identifikasi objek. *Tag* RFID sering juga disebut sebagai *transponder*. Antena: untuk mentransmisikan sinyal frekuensi radio antara RFID *reader* dengan *tag* RFID.
2. RFID *reader* : adalah alat yang kompatibel dengan *tag* RFID yang akan berkomunikasi secara *wireless* dengan *tag*.
3. *Software* Aplikasi: adalah aplikasi pada sebuah workstation atau PC yang dapat membaca data dari *tag* melalui RFID *reader*. Baik *tag* dan RFID *reader* dilengkapi dengan antena sehingga dapat menerima dan memancarkan gelombang elektromagnetik.



Gambar 2.1. Sistem RFID

Sumber:Subari DGZ, 2008, *RFID (Radio Frequency Identification)*,

<http://Subari.blogspot.com>

2.1.3. *Tag* RFID

Tag RFID adalah alat yang dibuat dari rangkaian elektronika dan antena yang terintegrasi di dalam rangkaian tersebut. Rangkaian elektronik dari

tag RFID umumnya memiliki memori sehingga *tag* ini mempunyai kemampuan untuk menyimpan data. Memori pada *tag* secara dibagi menjadi sel-sel. Beberapa sel menyimpan data *Read Only*, misalnya *serial number* yang unik yang disimpan pada saat *tag* tersebut diproduksi. Selain pada RFID mungkin juga dapat ditulis dan dibaca secara berulang.

Berdasarkan catu dayanya, *tag* RFID dapat digolongkan menjadi 2:

2.1.3.1.Tag Aktif

Tag Aktif yaitu *tag* yang catu dayanya diperoleh dari batere, sehingga akan mengurangi daya yang diperlukan oleh pembaca RFID dan *tag* dapat mengirimkan informasi dalam jarak yang lebih jauh. Kelemahan dari tipe *tag* ini adalah harganya yang mahal dan ukurannya yang lebih besar karena lebih kompleks. Semakin banyak fungsi yang dapat dilakukan oleh *tag* RFID maka rangkaianannya akan semakin kompleks dan ukurannya akan semakin besar.

2.1.3.2. Tag Pasif

Tag Pasif yaitu *tag* yang catu dayanya diperoleh dari medan yang dihasilkan oleh pembaca RFID. Rangkaianannya lebih sederhana, harganya jauh lebih murah, ukurannya kecil, dan lebih ringan. Kelemahannya adalah *tag* hanya dapat mengirimkan informasi dalam jarak yang dekat dan pembaca RFID harus menyediakan daya tambahan untuk *tag* RFID. Metode pengiriman data *tag* RFID pasif ke piranti pembaca dapat dibagi menjadi 2 macam, yaitu:

Sumber:Subari DGZ, 2008, *RFID (Radio Frequency Identification)*,
<http://Subari.blogspot.com>

Pada setiap bagian *tag* RFID seperti pada gambar 2.2, terdiri dari :

1. *Silicon Mikroprosesor*

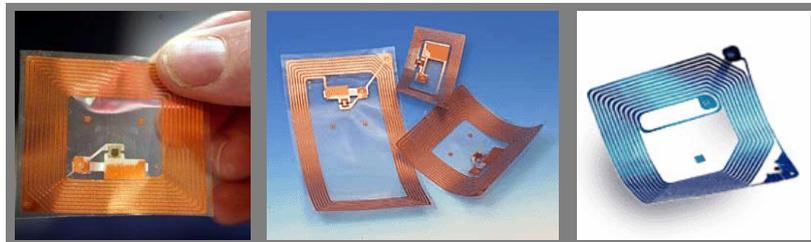
Ini adalah sebuah chip yang terletak dalam sebuah *tag* yang berfungsi sebagai penyimpan data.

2. *Metal Coil*

Sebuah komponen yang terbuat dari kawat tembaga atau aluminium yang berfungsi sebagai antena yang dapat beroperasi pada frekuensi 13,56 MHz. Jika sebuah *tag* masuk ke dalam jangkauan reader maka antena ini akan mengirimkan data yang ada pada *tag* kepada reader terdekat.

3. *Encapsulating Material*

Encapsulating Material adalah bahan yang membungkus *tag* yang terbuat dari bahan kaca.



Gambar 2.3. Bagian-bagian *tag* RFID

Sumber:Subari DGZ, 2008, *RFID (Radio Frequency Identification)*,
<http://Subari.blogspot.com>

2.1.4. Frekuensi Kerja RFID

Faktor penting yang harus diperhatikan dalam RFID adalah frekuensi kerja dari sistem RFID. Ini adalah frekuensi yang digunakan untuk komunikasi *wireless* antara pembaca RFID dengan *tag* RFID. Agar *tag* dengan RFID *reader* dapat berkomunikasi, maka *tag* RFID dan RFID *reader* harus di-*tune* pada frekuensi yang sama. Ada empat macam RFID *tag* yang sering digunakan bila dikategorikan berdasarkan frekuensi radio, yaitu:

1. *Low frequency tag* (antara 125 ke 134 kHz)
2. *High frequency tag* (13.56 MHz)
3. *UHF tag* (868 sampai 956 MHz)
4. *Microwave tag* (2.45 GHz)

UHF tag tidak bisa digunakan secara global, karena tidak ada peraturan global yang mengatur penggunaannya.

Pemilihan dari frekuensi kerja sistem RFID akan mempengaruhi jarak komunikasi, interferensi dengan frekuensi sistem radio lain, kecepatan komunikasi data, dan ukuran antena. Untuk frekuensi yang rendah umumnya digunakan *tag* pasif, dan untuk frekuensi tinggi digunakan *tag* aktif. Pada frekuensi rendah, *tag* pasif tidak dapat mentransmisikan data dengan jarak yang jauh, karena keterbatasan daya yang diperoleh dari medan elektromagnetik. Akan tetapi komunikasi tetap dapat dilakukan tanpa kontak langsung. Sedangkan pada frekuensi tinggi, jarak komunikasi antara *tag* aktif

dengan pembaca RFID dapat lebih jauh, tetapi masih terbatas oleh daya yang ada.

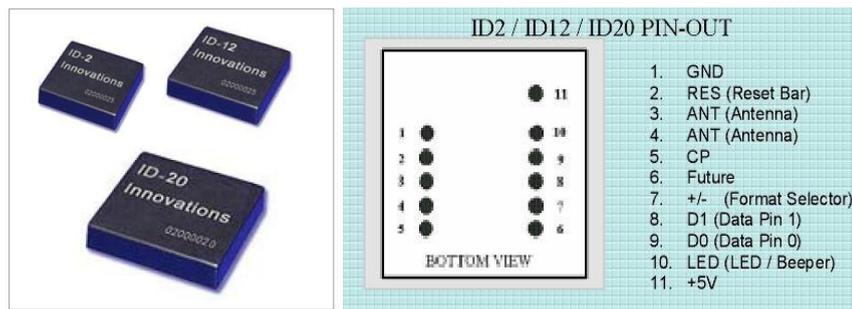
2.1.5. RFID Reader

Sebuah RFID *reader* harus menyelesaikan dua buah tugas, yaitu menerima perintah dari software aplikasi dan berkomunikasi dengan tag RFID. RFID *reader* merupakan penghubung antara *software* aplikasi dengan antena yang akan meradiasikan gelombang radio ke *tag* RFID. Gelombang radio yang diemisikan oleh antena berpropagasi pada ruangan di sekitarnya. Akibatnya data dapat berpindah secara wireless ke *tag* RFID yang berada berdekatan dengan antena.

RFID *reader* adalah alat yang kompatibel dengan *tag* RFID yang akan berkomunikasi menggunakan frekuensi radio dengan *tag*. RFID *reader* yang digunakan pada sistem ini adalah ID-12 yang mempunyai antena internal, frekuensi kerjanya 125 KHz dan dapat membaca *tag* pada jarak ± 12 cm. Konfigurasi *Pin* dari RFID *reader* (ID-12) adalah seperti gambar 2.2. *Pin* RFID *reader* yang digunakan pada sistem ini adalah sebagai berikut :

- a. *Pin* 1 dihubungkan ke *ground*.
- b. *Pin* 2 harus diberi kondisi high agar RFID *reader* tidak direset dan dihubungkan ke *VCC*.
- c. *Pin* 3 dan 4 merupakan *Pin reset* yang dihubungkan pada antena eksternal untuk mendapatkan jarak baca yang lebih jauh.

- d. *Pin 7* harus diberi kondisi *low* agar format data yang dikirim oleh *reader* adalah ASCII sedangkan kondisi *high* untuk format data *wiegand26*.
- e. *Pin 8* dan *9* merupakan *Pin output* data dari ID-12. *Pin 8* merupakan *Pin data 1* sedangkan *Pin 9* merupakan *Pin data 0*. Bila format data yang dipilih adalah ASCII, maka tegangan *output* data *Pin 8* adalah level CMOS, sedangkan tegangan *output* data *pin 9* adalah level TTL. Bila format data yang dipilih adalah *wiegand26*, maka *output* data pada *Pin 8* adalah *one output*. Sedangkan *output* data pada *pin 9* tersebut adalah *zero output*.
- f. *Pin 10* digunakan sebagai sinyal indikator bahwa data dari RFID *tag* telah terbaca, indikator yang digunakan berupa LED dan *buzzer*.
- g. *pin 11* dihubungkan ke *VCC*.



Gambar 2.4. Konfigurasi *Pin* ID12

Sumber: ID Innovations Advanced Digital Reader Technology, 2004, ID Series Datasheet ID2/12/20/2WR/12WR, www.id-innovations.com.

Tabel 2.1. Karakteristik Fisik dan Operasional ID12

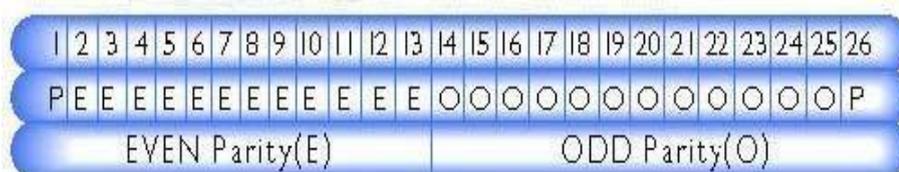
PARAMETERS	ID12
Read Range	12+ cm
Dimensions	26 mm x 25 mm x 7 mm
Frequency	125 kHz
Card Format	EM 4001 or compatible
Encoding	Manchester 64-bit, modulus 64
Power Requirement	5 VDC @ 30mA nominal
I/O Output Current	-
Voltage Supply Range	+4,6V though +5,4V

Sumber: ID Innovations Advanced Digital Reader Technology, 2004, ID Series Datasheet ID2/12/20/2WR/12WR, www.id-innovations.com



Gambar 2.5. Format data ASCII

Sumber: ID Innovations Advanced Digital Reader Technology, 2004, ID Series Datasheet ID2/12/20/2WR/12WR, www.id-innovations.com



P=Parity Start Bit and Stop Bit

Gambar 2.6. Format data *wiegand 26*

Sumber: ID Innovations Advanced Digital Reader Technology, 2004, ID Series Datasheet ID2/12/20/2WR/12WR, www.id-innovations.com

2.2. Mikrokontroler AVR ATmega16

AVR adalah mikrokontroler RISC (Reduce Instruction Set Compute) 8 bit berdasarkan arsitektur Harvart yang dibuat oleh Atmel pada tahun 1996. AVR mempunyai kepanjangan *Advanced Versatile RISC* atau *Alf and Vegard's Risc Prosesor* yang berasal dari nama dua mahasiswa Norwegian institute of Technology (NTH), yaitu Alf-Egil Bogen dan vegard Wollan.

AVR memiliki keunggulan dibandingkan dengan mikrokontroler lain, yaitu AVR memiliki kecepatan eksekusi program yang lebih cepat, karena sebagian besar intruksi dieksekusi dalam 1 siklus *clock*, lebih cepat dibandingkan dengan mikrokontroler MCS51 yang memiliki arsitektur CISC (*Complex Instruction Set Compute*) dimana mikrokontroler MCS51 membutuhkan 12 siklus *clock* untuk mengeksekusi 1 intruksi. Selain mikrokontroler AVR memiliki fitur yang lengkap antara lain :

1. ADC internal.
2. EEPROM internal.
3. *Timer/Counter*.
4. *Wanblog timer*.
5. *PWM, Port I/O*.
6. Komunikasi serial.
7. Komparator.
8. I2C, dll.

Sehingga mempermudah programmer dan desainer menggunakannya untuk berbagai aplikasi sistem elektronik. Secara umum mikrokontroler AVR dapat dikelompokkan menjadi 3 kelompok, yaitu keluarga AT90Sxx, ATmega, dan ATtiny.

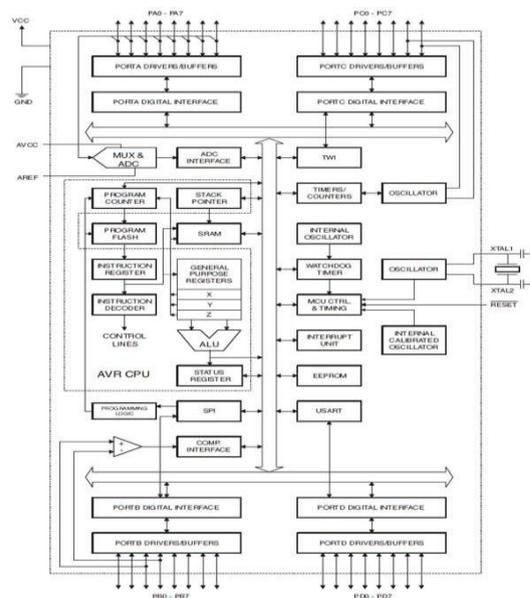
Tabel 2.2. Jenis mikrokontroler AVR

Mikrokontroler AVR		Memori		
Tipe	Jumlah <i>pin</i>	Flash	EEPROM	SRAM
TinyAVR	8–32	1–2K	64–128	0–128
AT90Sxx	20–44	1–8K	128–512	0–1K
ATmega	32–64	8-128K	512–4K	512–4K

Menggunakan Bahasa C (CodeVision AVR Sumber: Heri Andrianto, 2008

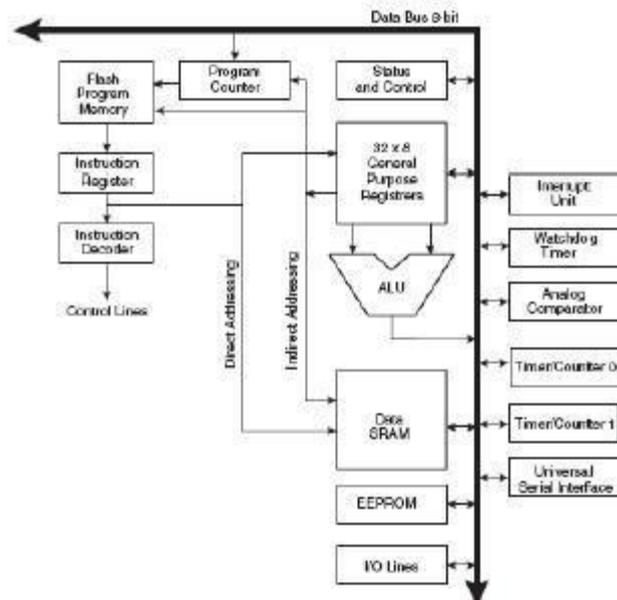
Pemrograman Mikrokontroler AVR ATMEGA16). Hal ,Informatika Bandung.

2.2.1. Blok Diagram



Gambar 2.7. Blok Diagram ATmega16

2.2.2. Arsitektur Mikrokontroler AVR RICS



Gambar 2.8. Arsitektur mikrokontroler AVR RICS

Dari gambar 2.8. AVR menggunakan arsitektur Harvard dengan memisahkan antara memori dan bus untuk program dan data sehingga memaksimalkan kemampuan dan kecepatan. Intruksi dalam memori program dieksekusi dengan *pipelining single level*. Dimana ketika satu intruksi dieksekusi, intruksi berikutnya diambil dari memori program. Konsep ini mengakibatkan intruksi dieksekusi setiap *clock cycle*. CPU terdiri dari 32x8-bit *general purpose register* yang dapat diakses dengan cepat dalam satu *clock cycle*, yang mengakibatkan operasi *Aritmatic Logic Unit* (ALU) dapat dilakukan dalam satu *cycle*. Pada operasi aritmatic dan logic pada ALU akan mengubah bit-bit yang terdapat pada suatu *Register* (SREG) secara paralel.

2.2.3. Fitur ATmega16

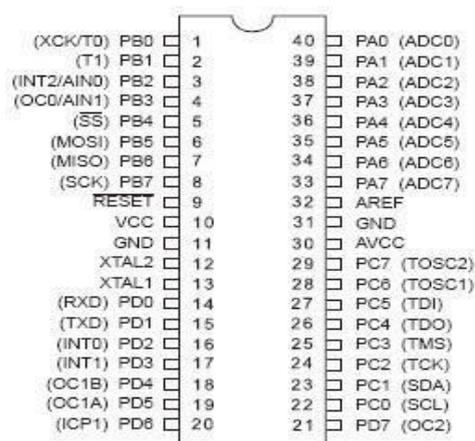
Fitur-fitur yang dimiliki ATmega16 adalah sebagai berikut :

1. Mikrokontroler AVR 8 bit yang memiliki kemampuan tinggi dengan daya rendah.
2. Arsitektur RISC dengan *throughput* mencapai 16 MIPS pada frekuensi 16MHz.
3. Memiliki kapasitas *Flash* memori 16 Kbyte, EEPROM 512 byte dan SRAM 1 Kbyte.
4. Saluran I/O sebanyak 32 buah. Yaitu *Port A*, *Port B*, *Port C*, dan *Port D*.
5. CPU yang terdiri atas 32 buah *register*.
6. Unit interupsi internal dan eksternal.
7. *Port* USART untuk komunikasi serial.
8. Fitur *Peripheral*
 - a. Tiga buah *Timer/Counter* dengan kemampuan pembandingan.
 - 2 buah *Timer/Counter* 8 bit dengan *prescaler* terpisah *Mode Compare*.
 - 1 buah *Timer/Counter* 16 bit dengan *Prescaler* terpisah, *Mode Compare*, dan *Mode capture*.
 - b. *Real Time Counter* dengan *Oscillator* tersendiri.
 - c. 4 channel PWM.
 - d. 8 channel, 10 bit ADC:

- 8 *Single-ended Channel*.
 - 7 *Differential Channel* hanya pada kemasan TQFP.
 - 2 *Differential Channel* dengan *Programmable Gain* 1x, 10x, atau 200x
- e. *Byte-oriented Two-wire Serial interface*.
 - f. *Programmable serial USART*. g. Antarmuka SPI.
 - h. *Watchdog Timer* dengan *oscillator internal*
 - i. *One-chip Analog Comperator*.

2.2.4. Konfigurasi *Pin* AVR ATmega16

Konfigurasi *pin* ATmega16 dengan kemasan 40 *pin* DIP (*Dual Inline Package*) dapat dilihat pada gambar 2.9.



Gambar 2.9. Konfigurasi kaki (*pin*) ATmega16

Dari gambar tersebut dapat dijelaskan fungsi dari masing-masing *pin* ATmega16 sebagai berikut :

1. VCC merupakan *pin* yang berfungsi masukan catu daya.
2. GND merupakan *pin Ground*.
3. *Port A* (PA0,PA7) merupakan pin input/output dua arah dan pin masukan ADC.
4. *Port B* (PB0,PB7), *Port D* (PD0,PD7) merupakan *Pin* input/output dua arah dan *Pin* fungsi khusus.
5. RESET merupakan *Pin* yang digunakan untuk me-reset mikrokontroler.
6. XTAL1 dan XTAL2 merupakan *Pin* masukan *clock* eksternal.
7. AVCC merupakan *Pin* masukan tegangan untuk ADC.
8. AREF merupakan *Pin* masukan tegangan referensi ADC

2.2.5. Port sebagai input/output digital

ATMega16 mempunyai empat buah *Port* yang bernama *PortA*, *PortB*, *PortC*, dan *PortD*. Keempat *Port* tersebut merupakan jalur bidirectional dengan pilihan internal pull-up. Tiap *Port* mempunyai tiga buah *register bit*, yaitu DDxn, PORTxn, dan PINxn. Huruf “x” mewakili nama huruf dari *Port* sebagian huruf “n” mewakili nomor *bit*. Bit DDxn terdapat pada I/O address PINx. Bila DDxn diset 1 maka Px berfungsi sebagai *pin* output. Bila DDxn diset 0 maka Px berfungsi sebagai *pin* input. Bila PORTxn diset 1 pada saat *pin* berfungsi sebagai pin input, maka resistir *pull-up* akan diaktifkan. Untuk mematikan resistor *pull-up*, PORTxn harus diset 0 atau *pin* dikonfigurasi sebagai *pin* output. *Pin*

port adalah *tri-state* setelah kondisi reset. Bila PORT_{xn} diset 1 pada saat *pin* terkonfigurasi sebagai *pin* output maka *pin port* akan berlogika 1. Dan bila PORT_{xn} diset 0 pada saat *pin* terkonfigurasi sebagai *pin* output maka *pin port* akan berlogika 0. Saat mengubah kondisi *port* dari kondisi *tri-state* (DD_{xn}=0,PORT_{xn}=0) ke kondisi *output high* (DD_{xn}=1,PORT_{xn}=1) maka harus ada kondisi peralihan apakah itu kondisi *pull-up enabled* (DD_{xn}=0, PORT_{xn}=1) atau kondisi *output low* (DD_{xn}=1, PORT_{xn}=0). Biasanya, kondisi *pull-up enable* dapat diterima sepenuhnya, selama lingkungan impedansi tinggi tidak memperhatikan perbedaan antara sebuah *strong high driver* dengan sebuah *puul- up*. Jika ini bukan suatu masalah, maka *bit* PUD pada *register* SFTOR dapat diset 1 untuk mematikan semua *pull-up* dalam semua *port*. Peralihan dari kondisi *input* dengan *pull-up* ke kondisi *output low* juga menimbulkan masalah yang sama. Kita harus menggunakan kondisi *tri-state* (DD_{xn}=0, PORT_{xn}=0) atau kondisi *output high* (DD_{xn}=1, PORT_{xn}=0) sebagai kondisi transisi.

Tabel 2.3. Konfigurasi Pin Port

		PUD		Pull –	
0	0	X	Input	No	Tri-state(Hi-Z)
					Pxn will source
0	1	1	Input	No	Tri-state(Hi-Z)
1	0	X	Output	No	Output Low(Sink)
1	1	X	Output	No	Output High(source)

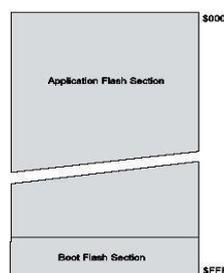
Sumber: <http://ilmu-computer.org/wp.content/uploads/2008/08/sholihul-atmega16.pdf...>

Bit 2-PUD *pull-up Disable*. Bila *bi* diset bernilai 1 maka *pull-up* pada *port* I/O akan dimatikan walaupun *register* DDxn dan PORTxn dikonfigurasi untuk menyalakan *pull-up* (DDxn=0, PORTxn=1).

2.2.6. Peta Memori AVR ATmega16

1. Memori Program

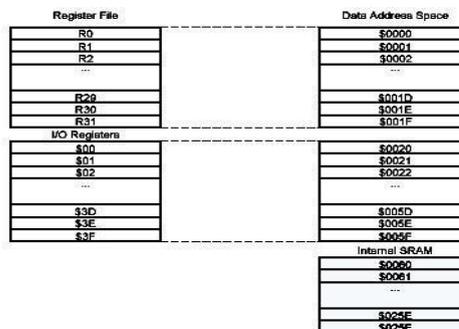
Arsitektur AVR mempunyai dua memori utama, yaitu memori data dan memori program. Selain itu, ATmega16 memiliki memori EEPROM untuk menyimpan data. ATmega16 memiliki 16k *byte on chip in-system Reprogrammable Flash Memory* untuk menyimpan program. Karena semua instruksi AVR memiliki format 16 dan 32 *bit*, *Flash* diatur dalam 8K x 16 *bit*. Untuk keamanan program, memori program dibagi dalam dua bagian, yaitu bagian program *Boot* dan aplikasi. *Bootloader* adalah program kecil yang bekerja pada saat *star up time* yang dapat memasukan seluruh program aplikasi dalam prosesor.



Gambar 2.10. Peta Memori Program AVR ATmega16

2. Memori Data (SRAM)

Memori data AVR ATmega16 terbagi menjadi 3 bagian, yaitu 32 buah *register* umum, 64 buah register I/O dan 1 Kbyte SRAM internal. *General purpose register* menempati alamat data terbawa, yaitu \$00 sampai \$1F. Sedangkan memori I/O menempati 64 alamat berikutnya mulai dari \$20 hingga \$5F. Memori I/O merupakan register yang khusus digunakan untuk mengatur fungsi terhadap berbagai peripheral mikrokontroler seperti control register, timer/counter, fungsi- fungsi I/O, dan sebagainya. 1024 alamat memori berikutnya mulai alamat \$60 hingga \$45 digunakan untuk SRAM internal.



Gambar 2.11. Peta Memori Data AVR ATmega16

3. Memori Data EEPROM

EEPROM (*electrically erasable programmable read only memory*) adalah memori yang masih dapat menyimpan data walaupun catu daya dimatikan. ATmega16 terdiri dari 512 *byte* memori data EEPROM 8 *bit*, data dapatditulis/dibaca dari memori ini, ketika catu daya dimatikan, data terakhir yang ditulis pada memori EEPROM masih tersimpan pada memori ini, atau dengan kata lain memori EEPROM bersifat *nonvolatile*. Alamat EEPROM mulai

\$000 sampai \$1FF. untuk mengakses EEPROM dilakukan dengan cara menentukan EEPROM Address Register, EEPROM data register dan EEPROM kontrol Register.

2.2.7. Interupsi

Interupsi adalah kondisi dimana pada saat program utama dieksekusi atau dikerjakan oleh CPU kemudian tiba-tiba terhenti untuk sementara waktu karena ada rutin lain yang harus ditangani terlebih dahulu oleh CPU dan setelah selesai mengerjakan rutin instruksi pada program utama. ATmega16 memiliki 21 sumber interupsi seperti pada tabel 2.4.

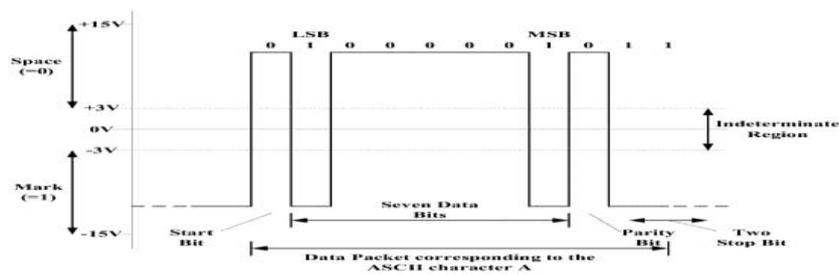
Table 2.4. Interrupt Vector Pada ATmega16

Vector No	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog Reset
2	0x0001	INT0	External Interrupt Request 0
3	0x0002	INT1	External Interrupt Request 1
4	0x0003	TIMER2 COMP	Timer/Counter2 Compare Match
5	0x0004	TIMER2 OVF	Timer/Counter2 Overflow
6	0x0005	TIMER1 CAPT	Timer/Counter1 Capture Event
7	0x0006	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	0x0007	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	0x0008	TIMER1 OVF	Timer/Counter1 Overflow
10	0x0009	TIMER0 OVF	Timer/Counter0 Overflow
11	0x000A	SPI, STC	Serial Transfer Complete
12	0x000B	USART, RXC	USART, Rx Complete
13	0x000C	USART, UDRE	USART Data Register Empty
14	0x000D	USART, TXC	USART, Tx Complete
15	0x000E	ADC	ADC Conversion Complete
16	0x000F	EE_RDY	EEPROM Ready
17	0x0010	ANA_COMP	Analog Comparator
18	0x0011	TWI	Two-wire Serial Interface
19	0x0012	INT2	External Interrupt Request 2
20	0x0013	TIMER0 COMP	Timer/Counter0 Compare Match
21	0x0014	SPM_RDY	Store Program Memory Ready

2.2.8. Komunikasi Serial USART

Komunikasi serial merupakan fitur yang penting dalam sistem *embended*, karena dengan komunikasi serial kita dapat dengan mudah menghubungkan

mikrokontroler dengan peralatan lainnya. Port serial pada mikrokontroler terdiri atas dua pin, yaitu RxD dan TxD. RxD berfungsi untuk menerima data dari komputer atau peralatan lainnya, sedangkan TxD berfungsi untuk mengirim data ke komputer atau peralatan lainnya. Pengiriman data serial dikirim satu per satu beserta format data serial yang umum.



Gambar 2.12. Format serial

Pada mikrokontroler ATmega16, pin PD0 dan PD1 digunakan untuk komunikasi serial USART (*Universal Synchronous Asynchronous Receiver/Transmitter*). USART pada ATmega16 memiliki beberapa keuntungan dibandingkan dengan sistem UART (*Universal Asynchronous Receiver-Transmitter*), yaitu :

1. Operasi *full duplex* (mempunyai register receiver dan transmit yang terpisah).
2. Mendukung komunikasi multiprocessor.
3. Mode kecepatan transmisi bermode Mbps

2.2.9. Inisialisasi USART

USART harus diinisialisasi sebelum komunikasi manapun dapat berlangsung. Untuk mengirim data serial menggunakan *Code VisionAVR*, untuk menggunakan fungsi *putcher, puts*, atau menggunakan I/O register UDR. Ada register yang perlu ditentukan nilainya, yaitu :

1. UBRR (USART Based Rate Register)

UBRR merupakan *register* 16 bit yang berfungsi untuk menentukan kecepatan transmisi data yang akan digunakan. UBRR dibagi menjadi dua yaitu UBRRH dan UBRRL. UBRR merupakan bit penyimpan konstanta kecepatan komunikasi serial. UBRRH menyimpan 4 bit tertinggi dan UBRRL menyimpan 8 bit sisanya. Data yang dimasukkan ke UBRRH dan UBRRL dihitung menggunakan rumus pada tabel 2.8.

Tabel 2.5. Rumus Perhitungan UBRR

Operating Mode	Equation for Calculating Baud Rate ⁽¹⁾	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{osc}}{16(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{osc}}{8(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{osc}}{2(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{2BAUD} - 1$

Sumber : Heri Adrianto, 2008, *Pemrograman Mikrokontroler AVR ATMEGA16 Menggunakan Bahasa C (codeVision AVR)*.Informatikabandung.

U2X adalah *bit* pada *register* UCRA yang berfungsi untuk menggandakan *transfery rate* menjadi dua kalinya. Hanya berlaku untuk mode asinkron sedangkan untuk mode sinkron *bit* ini diset 0.

2. UCSRA (USART Control And Status Register A)

Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

Gambar 2.13. UCSRA

Sumber: Heri Andrianto, 2008, *Pemrograman Mikrokontroler AVR ATMEGA16 Menggunakan Bahasa C (CodeVision AVR)*, Informatika Bandung

3. UCSRB (USART Control And Status Register B)

UCSRB merupakan register 8 bit pengatur aktivasi penerima dan pengirim USART, komposisinya sebagai berikut :

Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.14. UCSRB

Sumber: Heri Andrianto, 2008, *Pemrograman Mikrokontroler AVR ATMEGA16 Menggunakan Bahasa C (CodeVision AVR)*, Informatika Bandung

Penjelasan bit pada UCSRB

RXCIE : bit pengatur aktivasi interupsi penerimaan data serial.

TXCIE : bit pengatur aktivasi interupsi pengiriman data serial.

UDRIE : bit pengatur aktivasi interupsi yang berhubungan dengan kondisi bit UDRE pada UCSRA.

RXEN : bit pengatur aktivasi penerimaan serial ATmega16.

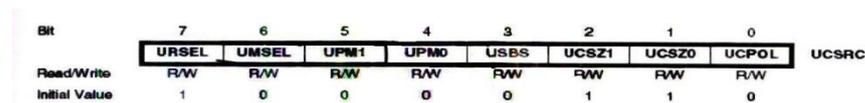
TXEN : *bit* pengatur aktivasi pengiriman serial ATmega16.

UCSZ2 bersama-sama dengan *bit* UCSZ1 dan UCSZ0 di *register*

UCSZ0 menentukan ukuran karakter serial yang dikirimkan.

4. UCSRC (USART Control And Status Register C)

UCSRC merupakan *register* 8 *bit* yang digunakan untuk mengatur mode dan kecepatan komunikasi serial yang dilakukan. Komposisinya seperti gambar di bawah ini :



Gambar 2.15. UCSRC

Sumber: Heri Andrianto, 2008, Pemrograman Mikrokontroler AVR ATMEGA16 Menggunakan Bahasa C (CodeVision AVR), Informatika Bandung.

Penjelasan *bit* pada UCSRC

URSEL : merupakan *bit* pemilih akses antara UCSRC dan UBRR.

UMSEL : merupakan *bit* pemilih mode komunikasi serial antara sinkron dan asinkron.

UPM : merupakan *bit* pengatur paritas.

USBS : merupakan *bit* pemilih ukuran *bit* stop.

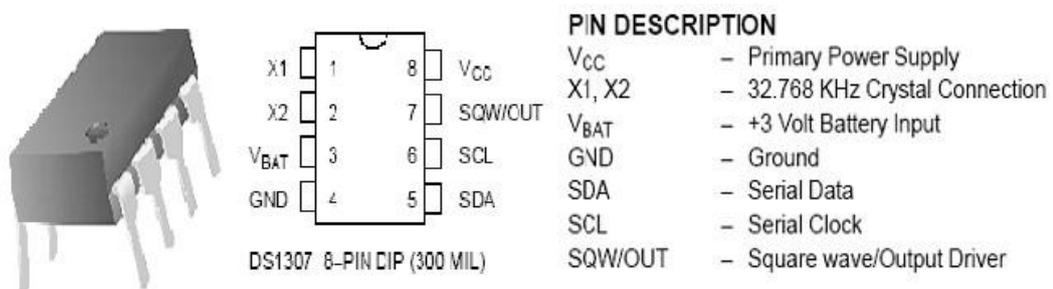
UCSZ2 dan UCSZ0 merupakan *bit* pengatur jumlah karakter serial.

UCPOL : merupakan *bit* pengatur hubungan antara perubahan data keluaran dan data masukan serial dengan *clock* sinkronisasi yang hanya berlaku untuk mode sinkron dan untuk mode asinkron *bit* ini diset 0.

2.3. Pengatur Informasi Waktu

Pada Gambar 2.16 pin RTC DS1307 ialah Pengambilan data yang akan dilakukan memiliki rentang waktu tertentu sesuai dengan pengaturan yang ditentukan. Pengaturan rentang waktu pengambilan data tersebut dapat dilakukan dengan sebuah chip (IC) yang sering dikenal dengan *Real Time Clock*. IC pengendali waktu yang digunakan dalam tugas akhir ini adalah RTC DS1307.

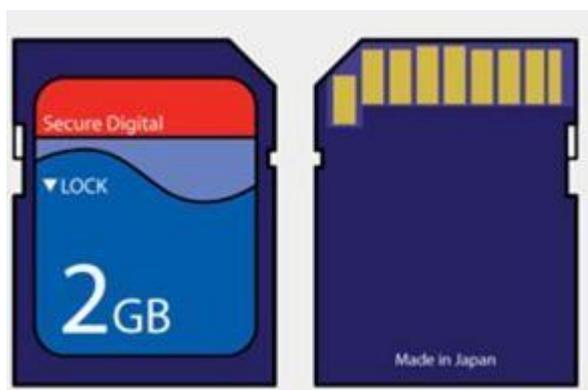
RTC DS1307 memiliki register yang dapat menunjukkan waktu dalam detik, menit, jam, tanggal, bulan dan tahun. RTC ini didesain dengan 128 lokasi RAM yang terdiri dari 15 *byte* untuk data waktu serta kontrol, dan 113 *byte* sebagai RAM yang dapat digunakan sebagai RAM pada umumnya. RTC DS 1307 menggunakan *bus* yang termultipleks sehingga bisa menghemat penggunaan pin pada mikrokontroler.



Gambar 2.16. Pin RTC DS1307

2.4. Memori Eksternal sebagai Penyimpan Data

Pada gambar 2.17 Memory MMC eksternal digunakan untuk sarana penyimpanan data pada alat elektronik seperti kamera digital dan handphone dan format data pada MMC umumnya menggunakan FAT. Secure Digital (SD) atau Multi Media Card (MMC) seringkali digunakan sebagai sarana penyimpan data pada Personal Digital Assistant (PDA), kamera digital, dan telepon seluler (ponsel). Beberapa perintah dasar untuk SD Card juga dapat digunakan untuk MMC sehingga kita dapat menggunakan SD atau MMC. Format data pada SD maupun MMC umumnya menggunakan format FAT. FAT12 digunakan untuk kapasitas 16 MB ke bawah. FAT16 digunakan untuk kapasitas 32 MB hingga 2 GB. FAT32. Memori yang akan digunakan dalam alat ini adalah SD/MMC memori.



Gambar 2.17. Memory MMC

2.5. Liquid Crystal Display (LCD) 16 x 2

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama. LCD sudah digunakan diberbagai bidang misalnya alat-alat elektronik seperti televisi, kalkulator, atau

pun layar komputer. Pada aplikasi LCD yang digunakan ialah LCD dot matrik dengan jumlah karakter 2 x 16. LCD sangat berfungsi sebagai penampil yang nantinya akan digunakan untuk menampilkan status kerja alat.

2.5.1. Fitur LCD 16 x 2

Adapun fitur yang disajikan dalam LCD ini adalah :

- a. Terdiri dari 16 karakter dan 2 baris.
- b. Mempunyai 192 karakter yang tersimpan
- c. Terdapat karakter generator terprogram
- d. Dapat dialamati dengan mode 4-bit dan 8-bit
- e. Dilengkapi dengan back light



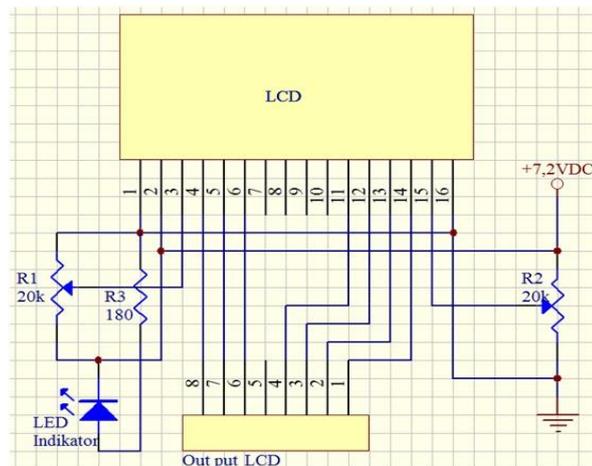
Gambar 2.18. Bentuk Fisik LCD 16 x 2

2.5.2. Spesifikasi Kaki LCD 16 x 2

LCD yang digunakan dalam penelitian ini adalah display LCD 2x16 yang mempunyai lebar display 2 baris dan 16 kolom atau biasa disebut sebagai LCD karakter 2x16, dan LCD tersebut mempunyai spesifikasi sebagai berikut:

Table 2.6. Spesifikasi LCD 16 x 2

Pin	Deskripsi
1	Ground
2	VCC
3	Pengatur Kontras
4	RS (Instruksi/Register Select)
5	R/W (Read/Write LCD Register)
6	EN (Enable)
7-14	Data I/O Pins
15	VCC
16	Ground



Gambar 2.19. Skematik LCD 16 x 2

2.6. Solenoid Door Lock

Pada Gambar 2.20 solenoid door lock adalah salah satu solenoid pengunci otomatis yang difungsikan khusus sebagai solenoid untuk pengunci pintu. Solenoid door lock yang digunakan dalam tugas akhir ini menggunakan normally close (NC) yang membutuhkan tegangan DC 12 Volt.



Gambar 2.20. Solenoid Door Lock

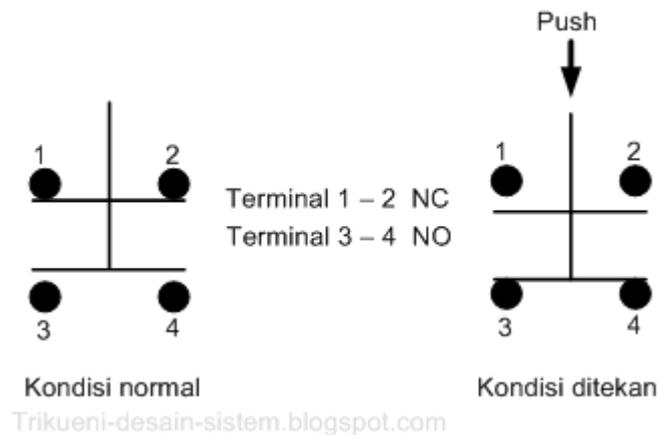
Cara kerja solenoid door lock:

1. Jika diberikan tegangan, electromagnet di dalamnya akan menarik lidah sehingga kunci akan terbuka (instan 1 detik dan bisa bertahan hingga 10 detik).
2. Jika tegangan listrik dihentikan, daya magnet nya hilang dan lidah kembali keluar sehingga terkunci kembali.

2.7. Push button

Pada Gambar 2.21 push button switch (saklar tombol tekan) 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 device penghubung atau pemutus aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan (dilepas), maka saklar akan kembali pada kondisi normal.

Sebagai device 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 2.21. Kondisi NC dan NO

Karena sistem kerjanya yang unlock dan langsung berhubungan dengan operator, push button switch menjadi device paling utama yang biasa digunakan untuk memulai dan mengakhiri kerja mesin di industri. Secanggih apapun sebuah mesin bisa dipastikan sistem kerjanya tidak terlepas dari keberadaan sebuah saklar seperti push button switch atau perangkat lain yang sejenis yang bekerja mengatur pengkondisian On dan Off.



Gambar 2.22. Push Button

Berdasarkan fungsi kerjanya yang menghubungkan dan memutuskan, push button switch 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 listrik). 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).