

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Sistem**

Sistem adalah sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Dari pengertian tersebut dapat dikatakan bahwa tidak semua system memiliki elemen yang sama, tetapi susunan dasar dari setiap sistem hampir sama. Setiap sistem memiliki masukan / *input* yang melalui sebuah *transformasi* / proses akan mengubah masukan tersebut menjadi keluaran / *output* yang dibutuhkan oleh pengguna sistem. Setiap masukan dan keluaran / *output* selalu ada mekanisme pengendaliannya dengan tujuan agar masukan yang akan diproses sesuai dengan standar yang ada sehingga keluaran / *output* yang dihasilkan dapat sesuai dengan yang dibutuhkan oleh pihak yang berinteraksi dengan sistem. (Munawaroh,2006) .

#### **2.2 Perlintasan kereta api**

Perlintasan kereta api adalah persilangan antara jalur kereta api dengan jalan, baik jalan raya atau pun jalan setapak kecil lainnya. Persilangan bisa terdapat di pedesaan ataupun perkotaan. Perlintasan terdiri dari perlintasan sebidang dan perlintasan tak sebidang. Perlintasan tak sebidang adalah persilangan antara jalur kereta api dengan jalan raya yang tidak pada satu bidang, misal dengan *flyover* atau *underpass*. Persyaratan pembuatan perlintasan tak sebidang :

1. Selang waktu antara kereta api satu dengan kereta api berikutnya yang melintas pada lokasi tersebut rata – rata sekurang – kurangnya 6 menit pada waktu sibuk
2. Jarak perlintasan yang satu dengan yang lainnya pada satu jalur kereta api tidak kurang dari 800 meter.

Perlintasan sebidang adalah persilangan antara jalur kereta dengan jalan raya pada satu bidang, yaitu di atas tanah. Persilangan ini banyak terdapat di pedesaan yang arus lalu lintas pada jalan tersebut masih relatif jarang.

### 2.3 Pengertian Komputer

Robert H. Blissmer dalam buku *Computer Annual* Komputer adalah suatu alat elektronik yg mampu melakukan beberapa tugas seperti menerima *input*, memproses *input* tadi sesuai dengan programnya, menyimpan perintah-perintah dan hasil pengolahan, serta menyediakan *output* dalam bentuk informasi. (Robert,2006).

### 2.4 Mikrokontroler

Mikrokontroler adalah sebuah sistem *microprocessor* dimana didalamnya sudah terdapat CPU, ROM, RAM I/O, clock dan peralatan *internal* lainya yang sudah saling terhubung dan terorganisasi (teralamat) dengan baik oleh pabrik pembuatnya dan dikemas dalam satu *chip* yang siap pakai. Sehingga kita tinggal memprogram isi ROM sesuai aturan penggunaan oleh pabrik yang membuatnya. (Winoto,2010).

### 2.5 Mikrokontroler AVR ATMega 16

Bascom-AVR merupakan seri mikrokontroler *Complementary Metal Oxide Semiconductor* (CMOS) 8-bit buatan *Atmel* berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Hampir semua instruksi pada program dieksekusi dalam satu siklus *clock*. AVR mempunyai 32 *register general-purpose*, *timer/counter* fleksibel dengan mode *compare*, *interupsi internal* dan *eksternal*, serial UART, *programmable Watchdog Timer*, *power saving mode*, ADC dan PWM. AVR pun mempunyai *In-System Programmable* (ISP) *Flash on-chip* yang mengijinkan memori program untuk diprogram ulang (*read/write*) dengan koneksi secara serial yang disebut *Serial Peripheral Inteface* (SPI).

*Bascom-AVR* memilki keunggulan dibandingkan dengan mikrokontroler lain, keunggulan mikrokontroler AVR yaitu memiliki kecepatan dalam mengeksekusi program yang lebih cepat, karena sebagian besar *instruksi* dieksekusi dalam 1 siklus *clock* (lebih cepat dibandingkan mikrokontroler keluarga MCS 51 yang memiliki arsitektur *Complex Intruksion Set Compute*). *ATMega16* mempunyai *throughput* mendekati 1 *Millions Instruction Per Second* (MIPS) per MHz,

sehingga membuat konsumsi daya menjadi rendah terhadap kecepatan proses eksekusi perintah.



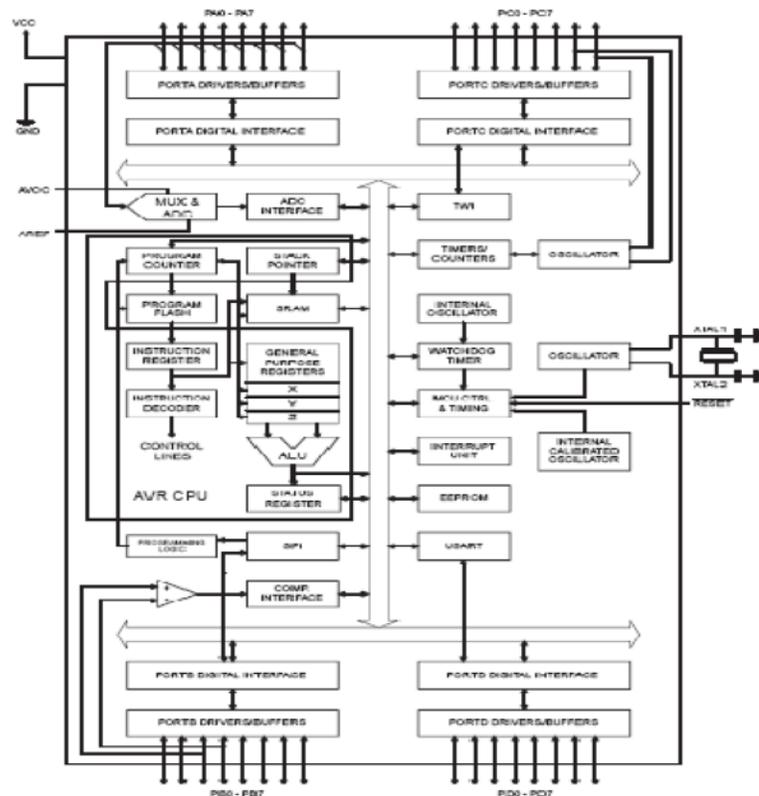
Gambar 2.1 Mikrokontroler ATmega 16

*ATmega16* mempunyai *throughput* mendekati 1 Millions Instruction Per Second (MIPS) per MHz, sehingga membuat konsumsi daya menjadi rendah terhadap kecepatan proses eksekusi perintah.

Beberapa keistimewaan dari AVR *ATmega16* antara lain:

1. Mikrokontroler AVR 8 bit yang memiliki kemampuan tinggi dengan konsumsi 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 dari 32 buah *register*.
6. Unit *interupsi* dan *eksternal*.
7. *Port* USART untuk komunikasi *serial*.
8. Fitur *peripheral*.
  - ✓ Tiga buah *Timer/Counter* dengan kemampuan perbandingan (*compare*)
    - Dua buah *Timer/Counter* 8 bit dengan *Prescaler* terpisah dan *Mode Compare*

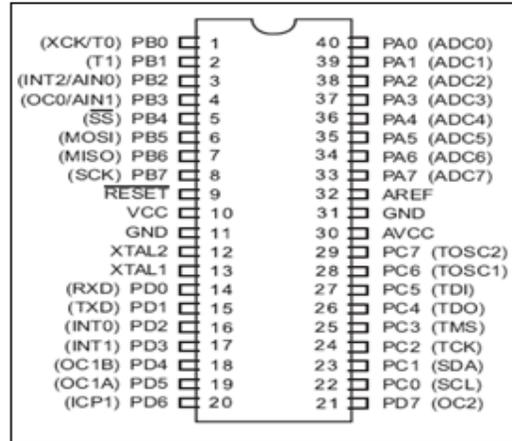
- Satu buah *Timer/Counter* 16 bit dengan *Prescaler* terpisah, *Mode Compare* dan *Mode Capture*
  - ✓ *Real Time Counter* dengan *Oscillator* tersendiri
  - ✓ Empat kanal PWM
  - ✓ 8 kanal ADC
    - 8 *Single-ended Channel* dengan keluaran hasil konversi 8 dan 10 resolusi (*register ADCH* dan *ADCL*)
    - 7 *Diferential Channel* hanya pada kemasan *Thin Quad Flat Pack (TQFP)*
    - 2 *Diferential Channel* dengan *Programmable Gain*
  - ✓ Antarmuka *Serial Peripheral Interface (SPI) Bus*
  - ✓ *Watchdog Timer* dengan *Oscillator Internal*
  - ✓ *On-chip Analog Comparator*
9. *Non-volatile program memory.*



Gambar 2.2 Blok diagram *ATmega16*

### 2.5.1 Konfigurasi Pin ATmega16

Mikrokontroler *ATmega16* memiliki 40 pin 32 pin diantaranya adalah *directional I/O* yang terbagi dalam 4 *port*. Dapat dilihat pada gambar dibawah ini.



Gambar 2.3 Konfigurasi Kaki Pin ATmega16

Konfigurasi *pin ATmega16* dengan kemasan 40 pin *Dual In-line Package* (DIP) dapat dilihat pada Gambar 2.13. dari gambar diatas dapat dijelaskan fungsi dari masing-masing *pin ATmega16* sebagai berikut.

1. VCC merupakan *pin* yang berfungsi sebagai masukan catu daya.
2. GND merupakan *pin Ground*.
3. *Port A* (PA0 – PA7) merupakan *pin input/output* dua arah (*full duplex*) dan selain itu merupakan *pin* masukan ADC.
4. *Port B* (PB0 – PB7) merupakan *pin input/output* dua arah (*full duplex*) dan selain itu merupakan *pin* khusus, seperti dapat dilihat pada tabel dibawa ini.

Tabel 2.1 Fungsi Khusus *Port B* (a.)

Pin	Fungsi Khusus
PB0	XCK (USART External Clock Input/Output) T0 (Timer/Counter0 External Counter Input)
PB1	T1 (Timer/Counter1 External Counter Input)
PB2	INT2 (External Interrupt 2 Input) AIN0 (Analog Comparator Negative Input)
PB3	OC0 (Timer/Counter0 Output Compare Match Output) AIN1 (Analog Comparator Negative Input)

Tabel 2.1 Fungsi Khusus *Port B* (b.)

PB4	SPI <i>Slave Select Input</i> )
PB5	MOSI (SPI <i>Bus Master Output /Slave Input</i> )
PB6	MISO (SPI <i>Bus Master Input/Slave Output</i> )
PB7	SCK (SPI <i>Bus Serial Clock</i> )

5. *Port A* (PC0 – PC7) merupakan *pin input/output* dua arah (*full duplex*) dan selain itu merupakan *pin* khusus, seperti dapat dilihat pada tabel dibawah ini.

Tabel 2.2 Fungsi Khusus *Port C*

Pin	Fungsi Khusus
PC0	SCL ( <i>Two-wire Serial Bus Clock Line</i> )
PC1	SDA ( <i>Two-wire Serial BusData Input/Output Line</i> )
PC2	TCK ( <i>Joint Test Action Group Test Clock</i> )
PC3	TMS ( <i>JTAG Test Mode Select</i> )
PC4	TDO ( <i>JTAG Data Out</i> )
PC5	TDI ( <i>JTAG Test Data In</i> )
PC6	TOSC1 ( <i>Timer Oscillator pin 1</i> )
PC7	TOSC2 ( <i>Timer Oscillator pin 2</i> )

6. *Port D* (PD0 – PD7) merupakan *pin input/output* dua arah (*full duplex*) dan selain itu merupakan *pin* khusus, seperti dapat dilihat pada tabel dibawah ini.

Tabel 2.3 Fungsi Khusus *Port D*

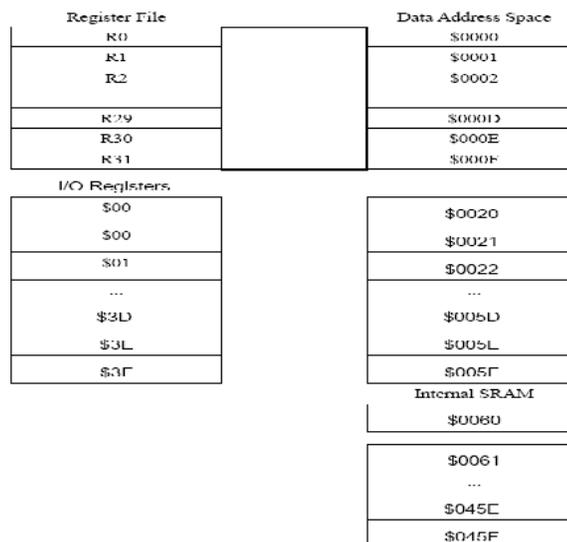
Pin	Fungsi Khusus
PD0	RXD (USART <i>Input Pin</i> )
PD1	TXD (USART <i>Output Pin</i> )
PD2	INT0 ( <i>External Interupt 0 Input</i> )
PD3	INT1 ( <i>External Interupt 1 Input</i> )
PD4	OC1B ( <i>Timer/Counter1 Output Compare B Macth Output</i> )
PD5	OC1A ( <i>Timer/Counter1 Output Compare A Macth Output</i> )
PD6	ICP ( <i>Timer/Counter1 Input Capture Pin</i> )
PD7	OC2 ( <i>Timer/Counter2 Output Compare Macth Output</i> )

7. RESET merupakan *pin* yang digunakan untuk me-*reset* mikrokontroler.
8. XTAL1 dan XTAL2, merupakan *pin masukan external clock*
9. AVCC merupakan *pin masukan tegangan* untuk ADC
10. AREF merupakan *pin masukan tegangan referensi* untuk ADC.

Terlihat pada gambar diatas, terdapat 4 buah *port* yaitu *Port A (PA)*, *Port B (PB)*, *Port C (PC)*, *Port D (PD)*, ke-empat *port* tersebut merupakan jalur *birectional* yang semuanya dapat diprogram sebagai *input* ataupun *output* dengan pilihan *internal pull-up*. Tiga *port* mempunyai tiga buah *register bit*, yaitu *DDxn*, *PORTxn*, dan *PINxn*. Huruf x mewakili nama *port* sedangkan n mewakili nama *bit*. Bit *PORTx* dan bit *PINxn* terdapat pada *I/O address PINx*. (Andrianto, 2008).

### 2.5.2 Memori Data (SRAM)

Memori data AVR *ATMega16* terbagi menjadi 3 bagian, yaitu 32 *register* umum, 64 buah *register I/O* dan 1 *Kbyte SRAM internal*. *General purpose register* menempati alamat data terbawah, yaitu \$00 sampai \$1F. Sedangkan *memori I/O* menempati 64 alamat berikutnya mulai dari \$20 hingga \$5F.



Gambar 2.4 Peta Memori Data *ATMega16*

Memori *I/O* merupakan *register* yang khusus digunakan untuk mengatur fungsi terhadap berbagai *fitur* mikrokontroler seperti kontrol *register*, *timer/counter*, fungsi-fungsi *I/O*, dan sebagainya. 1024 alamat berikutnya mulai dari \$60 hingga \$45F digunakan untuk SRAM *internal*.

### 2.5.3 Memori Data *EEPROM*

*ATMega16* terdiri dari 512 *byte* memori data *EEPROM* 8 *bit*, data dapat ditulis/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 dari \$000 sampai \$1FF.

## 2.6 Sensor

Menurut D.Sharon, dkk. (2008), mengatakan sensor adalah suatu peralatan yang berfungsi untuk mendeteksi gejala-gejala atau sinyal-sinyal yang berasal dari perubahan suatu *energy* seperti *energy* listrik, *energy* fisika, *energy* kimia, *energy* biologi, *energy* mekanik, dan sebagainya.

### 2.6.1 Sensor *Photodiode*

*Photodiode* adalah suatu jenis dioda yang resistansinya berubah-ubah jika cahaya yang jatuh pada dioda berubah-ubah intensitasnya. Dalam gelap nilai tahanannya sangat besar hingga praktis tidak ada arus yang mengalir. Semakin kuat cahaya yang jatuh pada dioda maka semakin kecil nilai tahanannya. *Photodiode* digunakan sebagai teknologi *elektronik* yang dapat dibentuk menjadi sensor *inframerah*. Cahaya diserap pada daerah penyambungan atau daerah *instrinsik* menimbulkan pasangan *electron-hole*, kebanyakan pasangan tersebut menghasilkan arus yang berasal dari cahaya.

*Photodiode* dapat dioperasikan dalam 2 macam mode yang berbeda yaitu :

1. *Mode photovoltaic* : seperti *solar sell*, penyerapan pada *photodiode* menghasilkan tegangan yang dapat diukur. Bagaimanapun, tegangan yang

2. dihasilkan dari tenaga cahaya ini sedikit tidak linear, dan *range* perubahannya sangat kecil.
3. *Mode fotokonduktivitas* : disini *photodiode* di aplikasikan sebagai tegangan *revers* (tegangan balik) dari sebuah dioda (yaitu tegangan pada arah tersebut pada dioda tidak akan menghantarkan tanpa terkena cahaya) dan pengukuran menghasilkan arus photo (hal ini juga bagus untuk mengaplikasikan tegangan mendekati nol)



Gambar 2.5. Photodiode dan Simbolnya

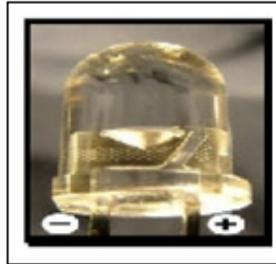
*Photodiode* digunakan sebagai penangkap gelombang cahaya yang dipancarkan oleh Infrared. Besarnya tegangan atau arus listrik yang dihasilkan oleh *photodiode* tergantung besar kecilnya radiasi yang dipancarkan oleh *infrared*. Sebuah *photodiode* mempunyai karakteristik yang lebih baik dari pada *phototransistor* dalam responya terhadap cahaya *inframerah*. Biasanya *photodiode* mempunyai respon 100 kali lebih cepat dari pada *phototransistor*.

### 2.6.2 Sensor Inframerah

Sistem sensor *infra merah* menggunakan *infra merah* sebagai media untuk komunikasi data antara *receiver* dan *transmitter*. Sistem akan bekerja jika sinar *infra merah* yang dipancarkan terpantul oleh suatu benda yang mengakibatkan sinar *inframerah* tersebut dapat terdeteksi oleh penerima. Keuntungan atau manfaat dari sistem ini dalam penerapannya antara lain sebagai pengendali jarak jauh, alarm keamanan, dan *otomatisasi* pada sistem. Pemancar pada sistem ini terdiri atas sebuah LED *infra merah* yang dilengkapi dengan rangkaian yang mampu membangkitkan data untuk dikirimkan melalui sinar *inframerah*, sedangkan pada bagian penerima biasanya terdapat *fototransistor*, *photodiode*, atau

*inframerah module* yang berfungsi untuk menerima sinar *inframerah* yang dikirimkan oleh pemancar.

Prinsip kerja dari alat ini adalah mengubah energi dari foton menjadi elektron. Idealnya satu foton dapat membangkitkan satu elektron. Salah satu bentuk energi foton adalah cahaya *inframerah*.



Gambar 2.6 Sensor *Inframerah*

#### Karakteristik Inframerah

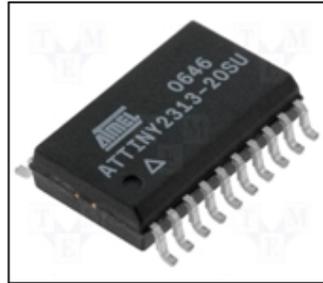
- ✓ Tidak dapat dilihat oleh manusia
- ✓ Tidak dapat menembus materi yang tidak tembus pandang
- ✓ Dapat ditimbulkan oleh komponen yang menghasilkan panas
- ✓ Panjang gelombang pada inframerah memiliki hubungan yang berlawanan atau berbanding terbalik dengan suhu. Ketika suhu mengalami kenaikan, maka panjang gelombang mengalami penurunan.

#### 2.7 ATtiny 2313

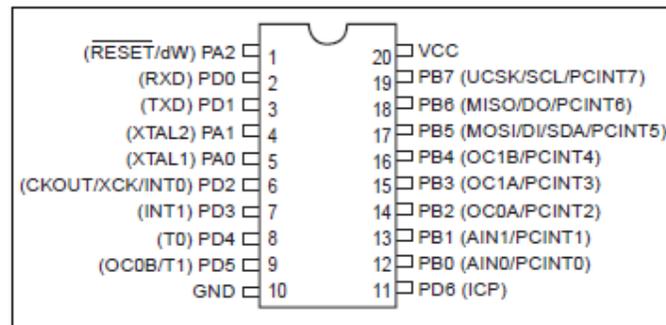
Perkembangan teknologi dalam dunia elektronika dewasa ini sangat pesat khususnya perkembangan dalam dunia *mikroelektronika*. *Mikrokontroler* adalah suatu komponen *semikonduktor* yang didalamnya sudah terdapat suatu sistem *mikroprosesor* seperti : ALU, ROM, RAM, dan Port I/O. Berdasarkan arsitekturnya *mikrokontroler* dapat dibedakan menjadi dua jenis yaitu *mikrokontroler* dengan arsitektur CISC dan *mikrokontroler* dengan arsitektur RISC. Perbedaan mendasar dari kedua jenis mikrokontroler ini adalah kecepatan dalam eksekusi program.

Mikrokontroler dengan arsitektur CISC untuk mengeksekusi 1 *cycle* perintah membutuhkan 12 siklus *clock* sedangkan mikrokontroler RISC semua

instruksi dikemas dalam kode 16-bit (*word*) sehingga sebagian besar instruksi dieksekusi dalam 1 siklus *clock*. Contoh mikrokontroler CISC adalah mikrokontroler keluarga ASM51 (AT89Cxx, AT89Sxx, dll) sedangkan mikrokontroler RISC contohnya adalah mikrokontroler keluarga AVR (AT90Sxx, ATtiny, Atmega)



Gambar 2.7 Mikrokontroler ATtiny 2313



Gambar 2.8 Pin ATtiny 2313

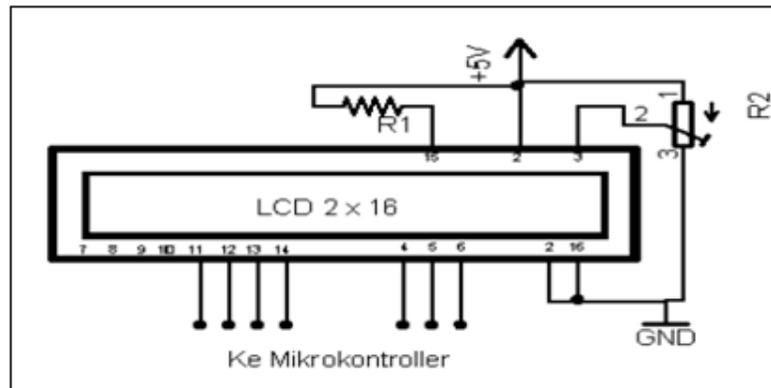
*Mikrokontroler* ATtiny2313 memiliki fitur-fitur utama antara lain sebagai berikut :

1. Sistem *mikroprosesor* RISC- 8bit dengan kecepatan maksimum 20Mhz.
2. 120 macam *instruksi*.
3. 32 x 8 bit *General Purpose Register*,
4. 2 *Kbyte memori* program,
5. 128 *byte memori* SRAM dan EEPROM,
6. *Timer/Counter* 8 bit dan 16 bit,

7. 18 *pin* jalur I/O,
8. 4 *channel* PWM (*Pulse Width Modulation*),
9. Komparator *analog*,
10. Komunikasi *serial* USART (*Universal Synchronous Asynchronous Receiver and Trasnmitter*),
11. Port SPI (*Serial Peripheral Interface*),
12. *Internal Osilator* dan *Internal Power-on-reset*.

## 2.8 LCD (*Liquid Crystal Display*)

LCD merupakan media penampil data yang sangat *efektif* dalam suatu sistem elektronika. Untuk menampilkan sebuah pesan atau gambar pada layar LCD, maka diperlukan sebuah rangkaian pengatur *scanning* dan pembangkit tegangan *sinus*. Pada rangkaian ini, penulis menggunakan modul LCD M16×32 sebagai penampilnya. LCD M16×32 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 *cursor*).



Gambar 2.9 Rangkaian LCD Karakter 16 x 2

Tabel 2.4 Konfigurasi *Pin* LCD 2 X 16 Karakter (a)

No.	Nama Pin	Deskripsi
1.	VCC	+5V
2.	GND	0V
3.	VEE	Tegangan kontras LCD
4.	RS	Register select, 0=register perintah, 1=register data
5.	R/W	1=read, 0=write

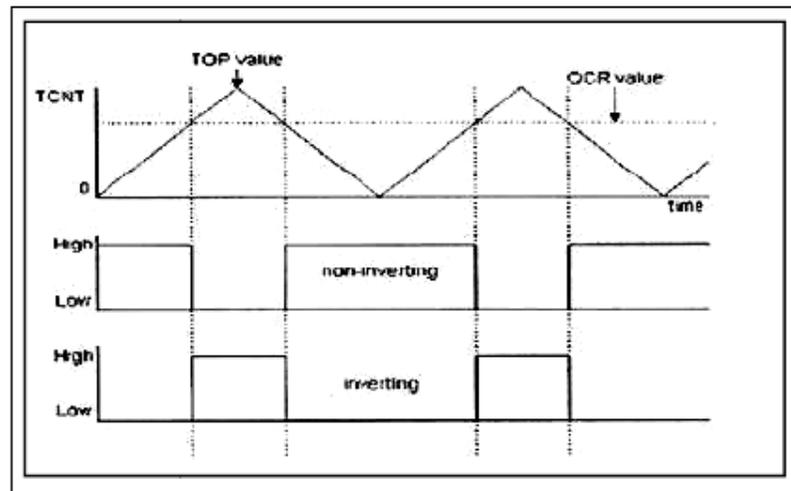
Tabel 2.4 Konfigurasi *Pin LCD 2 X 16 Karakter* (b)

6.	EN	<i>Enable Clock LCD</i> , logika 1 setiap kali pengiriman atau pembacaan data
7.	D0	Data Bus 0
8.	D1	Data Bus 1
9.	D2	Data Bus 2
10.	D3	Data Bus 3
11.	D4	Data Bus 4
12.	D5	Data Bus 5
13.	D6	Data Bus 6
14.	D7	Data Bus 7
15.	Anoda (Kabel Coklat Untuk LCD Hitachi)	Tegangan <i>positif backlight</i>
16.	Katoda (kabel merah untuk LCD Hitachi)	Tegangan <i>negative backlight</i>

### 2.9 *Pulse Width Modulation (PWM)*

PWM (*Pulse Width Modulation*) atau odulasi lebar pulsa adalah salah satu keunggulan *timer/counter* yang terdapat pada ATmega16. Ketiga jenis *Timer/Counter* pada ATmega16 dapat menghasilkan pulsa PWM. Pulsa PWM adalah sederetan pulsa yang lebar pulsanya dapat diatur. Pulsa PWM berfungsi mengatur kecepatan motor DC. PWM adalah *Timer mode Output Compare* yang canggih. Mode PWM Timer juga dapat mencacaturun yang berlawanan dengan mode Timer lainnya yang hanya mencacah naik. Pada mode PWM tersebut, Timer mencacah naik hingga mencapai nilai TOP, yaitu 0xFF (255) untuk PWM 8 bit dan 0x3FF (1023) untuk PWM 10 bit. *Timer/Counter 0* hanya memiliki PWM 8 bit, sedangkan pada *Timer/Counter 1* memiliki 9 bit PWM 10 bit, dan *Timer/Counter 2* memiliki PWM 8 bit.

Pemilihan mode PWM diseting melalui bit COM01 dan bit COM00 pada register TCCR. Saat COM00 *clear* dan COM01 set, pin OC0 *clear set timer* mencacah diatas *Compare Match* dan pin OC0 set saat *timer* mencavah dibawah *Compare Match* atau *non-inverting PWM*. Kebalikannya, saat COM00 set dan COM01 juga set, maka pin OC0 set saat timer mencacah dibawah *Compare Match* atau disebut juga *inverting PWM*. Agar lebih jelasnya perhatikan gambar ini.



Gambar 2.10 Pulsa PWM inverting dan non-inverting

## 2.10 Motor DC

Motor listrik merupakan perangkat *elektromagnetis* yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk, misalnya memutar *impeller* pompa, *fan* atau *blower*, menggerakkan *kompresor*, mengangkat bahan, dll. Motor listrik digunakan juga di rumah (*mixer*, bor listrik, *fan* angin) dan di industri. Motor listrik kadangkala disebut “kuda kerja” nya industri sebab diperkirakan bahwa motor menggunakan sekitar 70% beban listrik total di industri. (Sumanto, 1994)

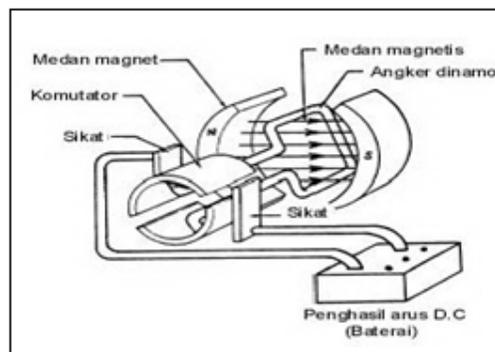
Motor DC umum yang menggunakan sikat (*brush*), yang menggunakan lilitan pada rotor dan menggunakan *magnet* tetap pada sisi *stator*, pada dasarnya dapat dianggap sebagai suatu beban yang dapat dihubungkan langsung ke rangkaian *switching* arus DC. Oleh karena itu, pemilihan ruang tepat cukup diperoleh dengan memperhatikan besar kebutuhan arus untuk memutar motor DC dapat diidentikkan dengan lilitan pada kumparan *relay* sehingga rangkaian driver-nya sama.

Tujuan motor DC adalah untuk menghasilkan gaya yang menggerakkan (*torsi*). Pada beberapa kasus sering diperlukan arah putaran motor DC yang berubah-ubah. Prinsip dasar untuk mengubah arah perputarannya adalah dengan membalik polaritas pada catu daya tegangannya.



Gambar 2.11 Motor DC

Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor dc disebut *stator* (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan *magnet*, maka akan timbul tegangan (GGL) yang berubah-ubah arah pada setiap setengah putaran, sehingga merupakan tegangan bolak-balik. Prinsip kerja dari arus searah adalah membalik *phasa* tegangan dari gelombang yang mempunyai nilai *positif* dengan menggunakan *komutator*, dengan demikian arus yang berbalik arah dengan kumparan jangkar yang berputar dalam medan magnet. Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen. (Sumanto, 1994).



Gambar 2.12 Bagian-Bagian Motor DC

Mekanisme kerja untuk seluruh jenis motor secara umum :

- Arus listrik dalam medan magnet akan memberikan gaya.

- Jika kawat yang membawa arus dibengkokkan menjadi sebuah lingkaran / *loop*, maka kedua sisi *loop*, yaitu pada sudut kanan medan magnet, akan mendapatkan gaya pada arah yang berlawanan.
- Pasangan gaya menghasilkan tenaga putar / *torque* untuk memutar kumparan.
- Motor-motor memiliki beberapa *loop* pada dinamonya untuk memberikan tenaga putaran yang lebih seragam dan medan magnetnya dihasilkan oleh susunan *elektromagnetik* yang disebut kumparan medan.

Maka dalam medan magnet akan dihasilkan kumparan medan dengan kerapatan fluks sebesar B dengan arus adalah I serta panjang konduktor sama dengan L maka diperoleh gaya sebesar F, dengan persamaan sebagai berikut :

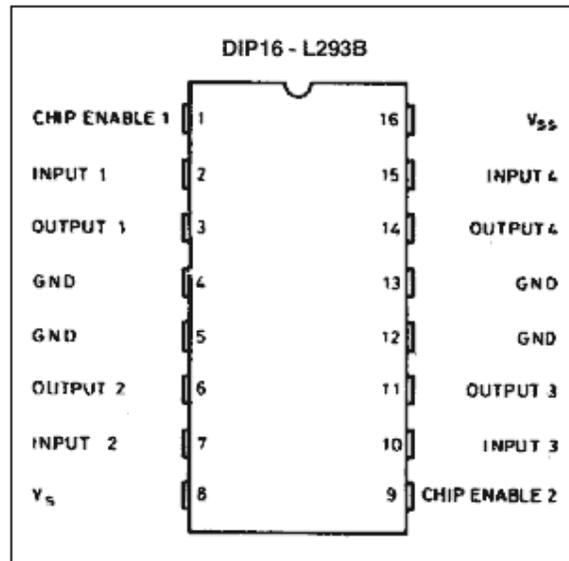
$$F = B I L$$

Dengan      F= Gaya magnet pada sebuah arus (Newton)  
                   B= Medan magnet (Tesla)  
                   I= Arus yang mengalir (Ampere)  
                   L= Panjang Konduktor (Meter)

### 2.10.1 Driver Motor DC

Driver Motor DC digunakan untuk menggerakkan motor DC menggunakan mikrokontroler. Arus yang mampu diterima atau dikeluarkan oleh mikrokontroler sangat kecil ( dalam satuan *miliampere*) sehingga agar mikrokontroler dapat menggerakkan motor Dcdiperlukan suatu rangkaian *driver* motor yang mampu mengalirkan rus sampai dengan beberapa *ampere*. Rangkaian *driver* motor DC dapat berupa rangkaian transistor, relay, atau IC (*Integrated Circuit*). Rangkaian *driver* yang umum digunakan adalah dengan IC L293D. IC L293D berisi 4 *channel driver* dengan kemampuan mengalirkan arus sebesar 600Ma per *Channel*.

Tegangan kerja ICL293D dari 6 Volt sampai dengan 36 Volt dan arus *impuls* tak berulang maksimum sebesar 1,2 *ampere*. Konfigurasi pin IC L293D ditunjukkan pada gambar dibawah ini.



Gambar 2.13 Konfigurasi Pin IC L293B

### 2.11 Power Supply

Catudaya atau *power supply* merupakan suatu rangkaian *elektronik* yang mengubah arus listrik bolak-balik menjadi arus listrik searah. Hampir semua peralatan *elektronik* membutuhkan catu daya agar dapat berfungsi.

### 2.12 Buzzer

*Buzzer* adalah sebuah *transduser* yang berfungsi untuk merubah *energy* listrik menjadi energi suara. Pada rangkaian alat pemantau kedatangan kereta *buzzer* dipergunakan sebagai *alarm*. Fungsi *buzzer* adalah sama seperti *speaker*, yaitu menghasilkan suara. Namun *buzzer* hanya mampu menghasilkan suara berfrekuensi tinggi, sedangkan *speaker* mampu untuk menghasilkan *frekuensi* tinggi dan rendah.

Jadi *buzzer* juga terdiri dari kumparan yang terpasang pada *diafragma* dan kemudian kumparan tersebut dialiri arus sehingga menjadi *elektromagnet*, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan *polaritas* magnetnya, karena kumparan dipasang pada *diafragma* maka setiap gerakan kumparan akan menggerakkan *diafragma* secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. *Buzzer* biasa digunakan

sebagai *indikator* bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (*alarm*).



Gambar 2.14 *Buzzer*

## 2.13 Komputer Interfacing

Saluran komunikasi antara peralatan diluar komputer dengan komputer umumnya menggunakan *port seial*, *port parallel* dan *port USB (Universal Serial Bus)*. Masing-masing *port* tersebut memiliki karakteristik sendiri-sendiri, tetapi dalam hal ini penulis hanya akan membahas karakteristik *port serial* saja. Hal tersebut dikarenakan penulis menggunakannya sebagai saluran komunikasi alat yang akan penulis lakukan.

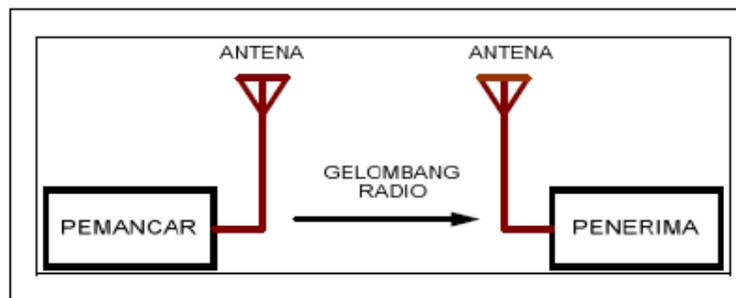
### 2.13.1 Komunikasi Serial

Transmisi data seri dibedakan menjadi dua macam yaitu komunikasi data *seri sinkron* dan komunikasi data *asinkron*. Perbedaan ini tergantung pada *clock* pendorong data. Dalam komunikasi data seri *sinkron*, *clock* untuk *shift register* ikut dikirimkan bersama dengan data *seri*. Sebaliknya dalam komunikasi data *seri asinkron* *clock* dalam *shift register* tidak ikut dikirim, rangkaian penerima data harus dilengkapi dengan rangkaian yang mampu membangkitkan *clock* yang diperlukan. Bagian yang terpenting dari komunikasi *seri asinkron* adalah upaya agar penerima data bisa membangkitkan *clock* yang bisa dipakai untuk mendorong *shift register* penerima. (Andrianto,2008).

## 2.14 Media Pemancar (Tranceiver)

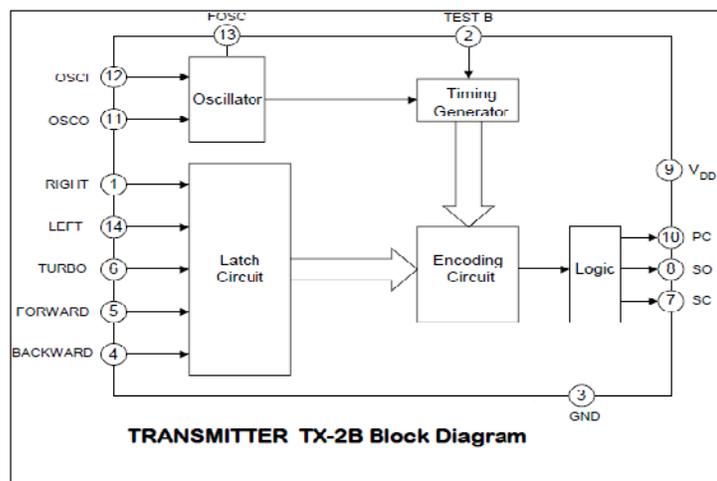
Pemancar adalah perangkat radio yang memproduksi sinyal modulasi frekuensi tinggi dan memancarkannya ke udara bebas melalui antena sebagai gelombang radio atau gelombang elektromagnetik. Pada sistem komunikasi melalui fasilitasnya sangat diperlukan adanya peralatan pemancar dan penerima serta media penghatarnya yaitu udara. Gelombang radio yang dihasilkan oleh pemancar bertugas membawa sinyal informasi ke penerima.

Pada pemancar dihasilkan suatu sinyal yang disebut dengan sinyal pembawa (*carrier*). Dikatakan sinyal pembawa karena sinyal ini akan membawa sinyal informasi dari pemancar ke penerima. Secara umum proses transmisi dari pemancar ke penerima dapat dilihat pada gambar berikut ini.

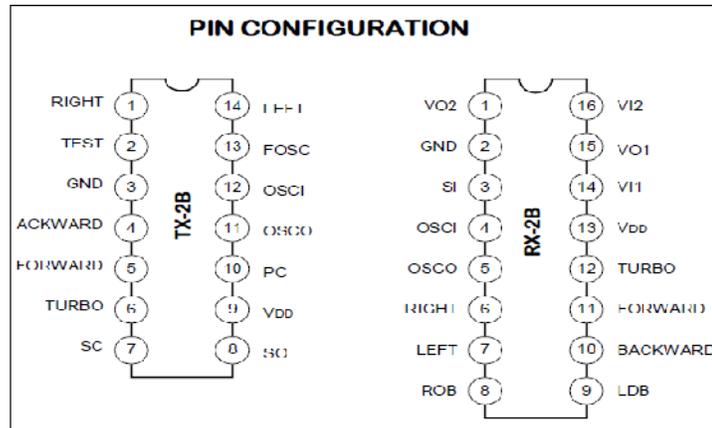


Gambar 2.15 Jalur Sistem Transmitter dan Receiver

Jarak jangkauan yang dapat dicapai atau ditempuh oleh gelombang radio dipengaruhi oleh daya keluaran dari sistem pemancar, penguatan antenna.



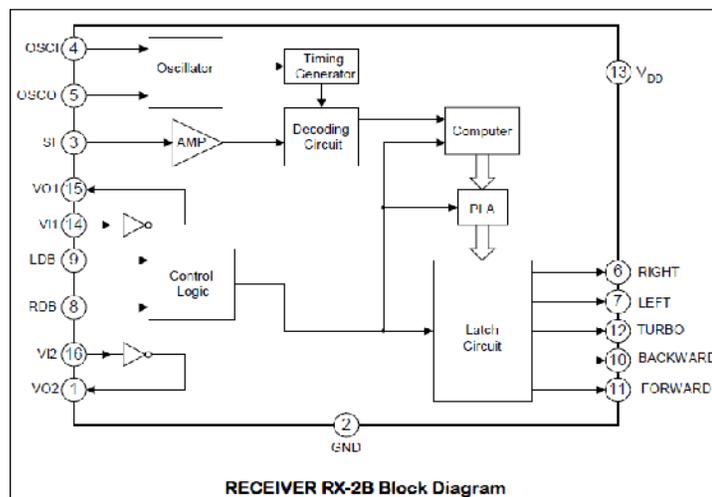
Gambar 2.16 Blok Diagram Transmitter TX2B



Gambar 2.17 Konfigurasi Pin IC TX2B dan RX2B

TX-2B / RX-2B merupakan IC CMOS yang berguna sebagai perangkat pemancar dan menerima yang yang biasa digunakan dalam sistem radio kontrol. TX-2B / RX-2B memiliki 5 fungsi yang biasa digunakan dalam radio kontrol.

## 2.15 Media Penerima (Receiver )



Gambar 2.18 Blok Diagram Receiver RX 2B

Data yang diterima sebisa mungkin bebas dari faktor error yang mungkin terjadi pada saat data dikirim dari pemancar maupun pada saat diterima oleh sistem pemancar (*Radio Frequency*). Sama dengan sistem pemancar, ada dua komponen utama yang ada pada sistem pemancar , yaitu antena dan alat penerima itu sendiri.

Disini antena berguna sebagai penerima gelombang radio sedangkan pada alat penerima memiliki berbagai elemen fungsi. Diantaranya adalah penyetoran frequency yang dikombinasikan langsung dengan antena, penyetoran data, penerjemah data, dan penghitung data sehingga mudah diolah menjadi sebuah produk informasi pada saat masuk ke sistem komputer.

## **2.16 Antena**

Antena merupakan salah satu perangkat radio yang berkerja mengubah sinyal listrik menjadi sinyal gelombang *elektromagnetik* dan dipancarkan ke udara bebas atau sebaiknya menangkap sinyal *gelombang elektromagnetik* dari udara bebas dan mengubahnya menjadi sinyal listrik. Antena mempunyai beberapa fungsi yaitu sebagai *converter*, sebagai radiator, dan sebagai penyesuaian *impedansi*. Ketiga fungsi tersebut berlaku pada saat yang bersamaan. Antena dikatakan sebagai *converter* karena antena tersebut mengubah bentuk sinyal yaitu sinyal listrik menjadi gelombang *elektromagnetik* atau sebaliknya.

Antena dikatakan sebagai radiator karena antena tersebut meradiasikan energi gelombang dimana energi tersebut akan diradiasikan ke segala arah disekitarnya atau sebaliknya apabila antena menerima atau menangkap energi radiasi gelombang *elektromagnetik*, maka dikatakan sebagai *radiator*.

Antena dikatakan sebagai *impedance matching* karena pada saat berkerja, akan menyesuaikan *impedance* pesawat radio dan *impedance* udara bebas. Adapun *impedance* radio biasanya dinyatakan oleh *impedance* karakteristik saluran yang menghubungkan pesawat radio dengan antena.

## **2.17 Bahasa Pemograman Mikrokontroler**

Secara umum bahasa pemograman *mikrokontroler* adalah bahasa tingkat rendah *assembly*, dimana setiap *mikrokontroler* memiliki bahasa-bahasa pemograman yang berbeda-beda. Banyaknya hambatan dalam penggunaan bahasa *assembly*, maka dikembangkan *compiler* atau penerjemah untuk bahasa tingkat tinggi. (Wahyudin,2006).

### 2.17.1 *Basic Compiler (BASCOM-AVR)*

Bahasa *BASIC* adalah salah satu bahasa pemrograman yang banyak digunakan untuk aplikasi *mikrokontroler* karena kemudahan dan *kompatibel* terhadap mikrokontroler jenis AVR dan didukung oleh *compiler software* berupa *BASCOM-AVR* yang dibuat untuk melakukan pemograman *chi-chip* mikrokontroler tertentu, salah satunya adalah *ATMega 16*.

**Menu Bar** : Secara umum pilihan menu pada *menu bar* hampir sama dengan *software* lain yang berbasis *Windows* sebagai *Operating System*.

**Toolbar** : *Toolbar* berisikan ikon-ikon yang mewakili perintah-perintah dasar yang ada pada *menu bar* yang sering digunakan.

#### 1. *Menu Bar BASCOM AVR*



Gambar 2.19 *Menu Bar BASCOM AVR*

Secara umum pilihan *menu* pada *menu bar* hampir sama dengan *software* lain yang berbasis *Windows* sebagai *Operating System*. Pada *menu bar* pilihan *menu BASCOM AVR* hampir sama dengan *software* lain yang berbasis *Windows*. Sebagai contoh pada pilihan *menu file* berisikan antara lain *submenu New, Open, Save dan Close*.

#### 2. *Toolbar BASCOM AVR*

*Icon-icon command* atau perintah pada *toolbar* adalah *icon* yang mewakili perintah-perintah *submenu* yang terdapat di *menu bar*. Pada dasarnya perintah *submenu* yang ditampilkan di *toolbar* dalam bentuk *ikon* berguna untuk mempercepat akses melalui *mause*. Pada *toolbar* anda bisa mengurangi atau menambahkan *ikon-ikon command* mana saja yang perlu ditampilkan untuk mempercepat penggunaan.



Gambar 2.20 *Toolbar BASCOM AVR*

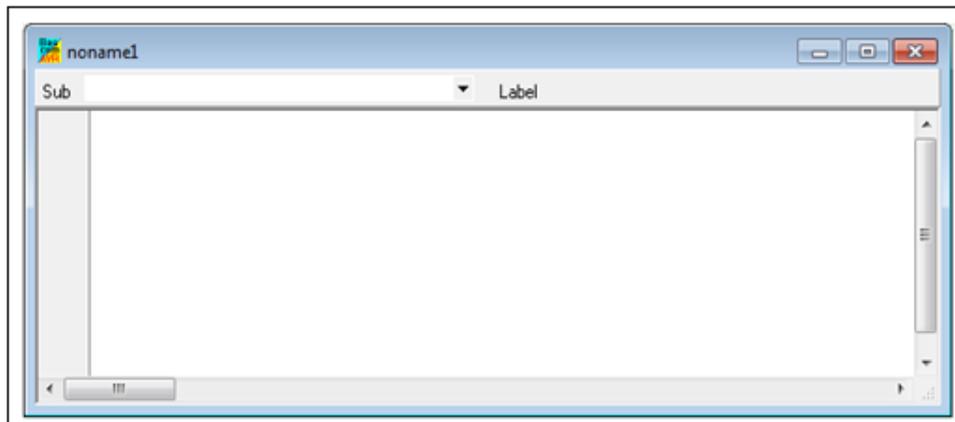
Keterangan lengkap ikon-ikon dari program *BASCOM AVR* dapat dilihat pada tabel berikut ini :

Tabel 2.5 Keterangan ikon-ikon pada *BASCOM AVR*

Icon	Nama	Fungsi	Shortcut
	<i>File New</i>	Membuat file baru	Ctrl+N
	<i>Open File</i>	Untuk membuka File	Ctrl+O
	<i>File Close</i>	Untuk menutup program yang dibuka	-
	<i>File Save</i>	Untuk menyimpan file	Ctrl+S
	<i>Save as</i>	Menyimpan dengan nama yang lain	-
	<i>Print Preview</i>	Untuk melihat tampilan sebelum dicetak	-
	<i>Print</i>	Untuk mencetak dokumen	Ctrl+P
	<i>Exit</i>	Untuk keluar dari program	-
	<i>Program Compile</i>	Untuk mengkompile program yang dibuat, <i>output</i> -nya bisa berupa *.hex, *.bin dll.	F7
	<i>Syntax Check</i>	Untuk memeriksa kesalahan bahasa	Ctrl+F7
	<i>Show Result</i>	Untuk menampilkan hasil kompilasi program.	Ctrl+W

### 3. Text Editor

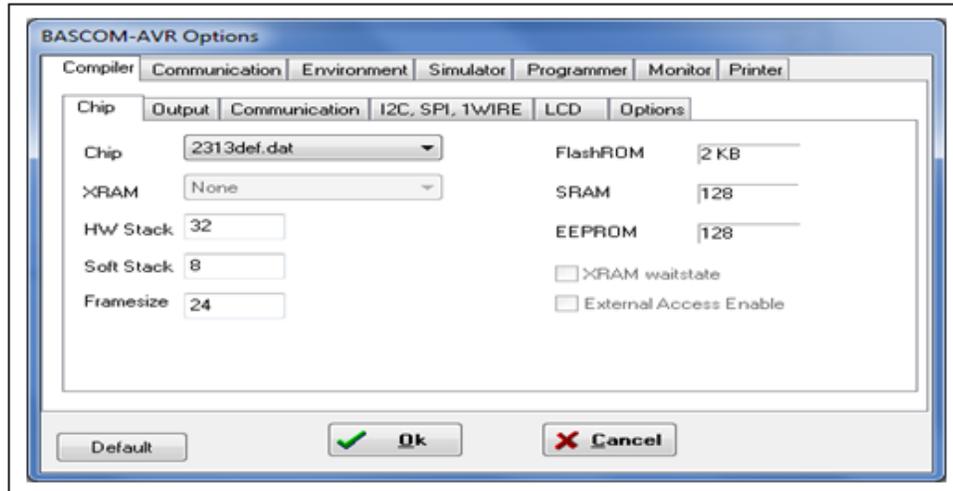
*Text editor* adalah tempat menulis atau membuat program. Setelah *text editor* tampil maka isikan pemrograman tersebut pada *text editor*.



Gambar 2.21 *Text Editor*

### 2.17.2 Compiler

*BASCOM AVR* menyediakan pilihan untuk memodifikasikan pilihan pada kompilasi. Dengan memilih *menu compiler* maka jendela akan ditampilkan seperti pada gambar ini :



Gambar 2.22 Jendela option BASCOM AVR

Keterangan dari gambar diatas dapat dilihat pada tabel berikut ini :

Tabel 2.6 Keterangan Menu Pilihan (a.)

Tab Menu	Option	Keterangan
<i>Chip</i>	<i>Chip</i>	<i>Mikrokontroler</i> yang digunakan sebagai contoh m16.dat untuk <i>ATMega 16</i> .
	<i>XRAM</i>	Jika menggunakan <i>eksternal</i> RAM nilai ini bisa ditampilkan.
	<i>Soft Stack</i>	Stack <i>software</i> nilai default nya 8.
	<i>FlashROM</i>	Nilai <i>flashROM</i> chip yang dipilih.
	<i>SRAM</i>	Nilai RAM <i>internal</i> chip yang dipilih.
	<i>EEPROM</i>	Nilai <i>EEPROM</i> chip yang dipilih.
<i>Output</i>		<i>File output</i> yang dihasilkan dalam proses kompilasi.
<i>Communication</i>	<i>Baudrate 0</i>	Nilai <i>baudrate</i> yang digunakan dalam komunikasi <i>serial</i> .
	<i>Frekuensi</i>	Nilai <i>osilator</i> yang digunakan.

Tabel 2.6 Keterangan Menu Pilihan (b.)

<i>I2C,SPI, 1 Wire</i>	<i>SDA</i>	Pin yang berfungsi untuk data <i>serial</i> dalam komunikasi <i>I2C</i> .
	<i>SCL</i>	Pin yang berfungsi untuk data <i>clock</i> dalam komunikasi <i>I2C</i> .
	<i>1 Wire</i>	Pin yang berfungsi untuk komunikasi <i>1 wire</i>
	<i>SPI</i>	Pin untuk komunikasi <i>serial sinkron</i> .

### 2.17.3 Operasi –operasi pada BASCOM AVR

Pada bagian ini akan dibahas tentang bagaimana cara menggabungkan, memodifikasi, membandingkan atau mendapatkan informasi tentang sebuah pernyataan dengan menggunakan *operator-operator* yang tersedia di *BASCOM* seperti berikut ini :

#### 1. Operator Aritmatika

Digunakan dalam perhitungan, yang termasuk *operator aritmatika* adalah :

Tabel 2.7 Operator Aritmatika

Operator	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian
Operator /	Pembagian

#### 2. Relasi

Digunakan untuk membandingkan nilai sebuah angka, hasilnya dapat digunakan untuk membuat keputusan sesuai dengan program yang kita buat. Yang termasuk relasi adalah :

Tabel 2.8 Operator Relasi

Operator	Relasi	Pernyataan
=	Sama Dengan	$X = Y$
<>	Tidak Sama Dengan	$X <> Y$
<	Lebih Kecil Dari	$X < Y$
>	Lebih Besar Dari	$X > Y$
<=	Lebih Kecil atau sama dengan	$X <= Y$

### 3. Operator Logika

Digunakan untuk menguji sebuah kondisi atau untuk memanipulasi *bit* dan operasi *Boolean*. Dalam BASCOM ada 4 buah operator yaitu *AND*, *OR*, *NOT*, dan *XOR*. Operator logika ini juga bisa digunakan untuk menguji sebuah *byte* dengan pola bit tertentu, sebagai contoh :

```
DIM A As Byte
A = 63 And 19
Print A
A = 10 Or 19
Print A
Output A
16
```

### 4. Operator Fungsi

Digunakan untuk melengkapi operator yang sederhana.

#### 2.17.4 Tipe Data BASCOM-AVR

Setiap *variabel* dalam *BASCOM* memiliki tipe data yang menunjukkan daya tampung *variabel* tersebut, hal ini berhubungan dengan penggunaan memori dari mikrokontroler. Berikut ini adalah tipe data pada *BASCOM* berikut keterangannya. (Fazri,2010).

Tabel 2.9 Tipe Data *BASCOM-AVR*

Tipe Data	Ukuran (Byte)	Range
<i>Bit</i>	1/8	-
<i>Byte</i>	1	0 – 255
<i>Integer</i>	2	-32,768 - + 32,767
<i>Word</i>	2	0 – 65535
<i>Long</i>	4	-2147483648 - + 2147483647
<i>Single</i>	4	-
<i>String</i>	s/d 254 byte	-

### 2.17.5 Variabel

*Variabel* dalam sebuah program berfungsi sebagai tempat penyimpanan data atau penampung data sementara, misalnya menampung hasil perhitungan, menampung data hasil pembacaan *register* dan lain-lain. *Variabel* merupakan *pointer* yang menunjuk pada alamat memori fisik di mikrokontroler.

Dalam *BASCOM* ada beberapa aturan dalam penamaan sebuah variabel :

1. Nama *variabel* maksimum terdiri atas 32 karakter.
2. *Karakter* bisa berupa angka atau huruf.
3. Nama *variabel* harus dimulai dengan huruf.
4. *Variabel* tidak boleh menggunakan kata-kata yang digunakan oleh *BASCOM* sebagai perintah, pernyataan, internal *register* dan nama *operator* (*AND, OR, DIM* dan lain-lain).

Contoh pendeklarasian menggunakan *DIM* sebagai berikut :

```
Dim nama as byte
Dim tombol1 as integer
Dim tombol2 as word
Dim tombol3 as word
Dim tombol4 as word
Dim Kas as string*10
```

- ✓ Untuk mempercepat pendeklarasian sebuah *variable* yang banyak adalah :

```
Dim nama as byte, tombol1 as integer
Dim tombol2 as bit, tombol4 as word
Dim kas as string *10
```

- ✓ Cara lain untuk mendeklarasikan sebuah *variable* dengan menggunakan *DEFINT, DEFBIT, DEFBYTE* dan *DEFWORD*, sebagai contoh :

```
DEFBYTE nama
DEFINT tombol1
DEFWORD tombol2 ; tombol3 ; tombol4
```

### 2.17.6 Kontrol Program pada *BASCOM-AVR*

Keunggulan sebuah pemrograman terletak pada kontrol program ini. Dengan kontrol program, dapat mengendalikan alur dari sebuah program dan menentukan apa yang harus dilakukan oleh sebuah program ketika menemukan

suatu kondisi tertentu. Berikut beberapa kontrol program yang sering digunakan dalam pemrograman dengan *BASCOM-AVR*. (fazri,2010)

#### 1. **IF ... THEN**

Dengan pernyataan ini, dapat menguji sebuah kondisi tertentu dan kemudian menentukan tindakan yang sesuai dengan kondisi yang diinginkan. *Sintak* penulisannya adalah sebagai berikut :

```
IF <syarat kondisi> THEN <pernyataan>
```

*Sintak* diatas digunakan jika hanya ada satu kondisi yang diuji dan hanya melakukan satu tindakan. Jika melakukan lebih dari satu tindakan maka sintaks yang akan dibuat adalah sebagai berikut :

```
IF <syarat kondisi> THEN  
    <pernyataan ke-1>  
    <pernyataan ke-2>  
    <pernyataan ke n>
```

```
END IF
```

Jika ada lebih dari satu kondisi atau lebih yang akan diuji maka sintaknya menjadi :

```
IF <syarat kondisi 1> THEN  
    <blok pernyataan ke-1>  
ELSEIF <syarat kondisi 2> THEN  
    <blok pernyataan ke-2>  
ELSEIF <syarat kondisi n> THEN  
    <blok pernyataan ke-n>  
ELSE  
    <blok pernyataan>  
END IF
```

#### 2. **SELECT ... CASE**

Perintah ini akan mengeksekusi beberapa blok pernyataan tergantung dari nilai *variable*-nya. Perintah ini mirip dengan **IF ... THEN**, namun perintah ini memiliki kelebihan yaitu kemudahan pada penulisannya. *Sintaknya* sebagai berikut :

```

SELECT CASE variabel
    CASE test1 : statiments
    CASE test2 : statiments
    CASE ELSE : statiments
END SELECT

```

### 3. DO ... LOOP

Digunakan untuk mengetahui sebuah blok pernyataan terus menerus. Untuk membatasi perulangannya dapat ditambahkan sebuah syarat kondisi agar perulangan berhenti dan perintahnya menjadi **DO ... LOOP UNTIL**

*Sintak* penulisannya sebagai berikut:

```

DO
    <blok pernyataan>
LOOP

```

### 4. FOR ... NEXT

Digunakan untuk mengeksekusi sebuah blok pernyataan secara berulang. Perintah ini hampir sama dengan perintah **DO ... LOOP**, namun pada perintah ini awal dan akhir perulangan serta tingkat kenaikan atau turunya bisa ditentukan.

```

FOR var = start TO/DOWNTO end [STEP value]
    <blok pernyataan>
NEXT

```

Untuk menentukan nilai perulangan maka gunakan **TO** dan untuk menurunkannya gunakan **Downto**. Tingkat kenaikan merupakan pilihan, sehingga bisa digunakan atau tidak.

### 5. GOSUB

Dengan perintah **GOSUB**, program akan melompat ke sebuah label dan akan menjalankan program yang ada dalam sub rutin tersebut sampai menemui perintah **RETURN**. Perintah **RETURN** akan mengembalikan program ke titik setelah perintah **GOSUB**. Sintaknya sebagai berikut :

```

Print "coba rutin"
GOSUB cabang
Print "Hello"
END

```

Cabang :

```

X=x + 2
PRINT
X
RETURN

```

#### 6. WHILE ... WEND

Perintah ini akan mengeksekusi sebuah pernyataan secara perulangan ketika masih menemukan kondisi yang sama. Perintah ini akan berhenti jika ada perubahan kondisi dan melakukan perintah selanjutnya.

*Sintaksnya* sebagai berikut :

```

WHILE <syarat kondisi>
    <pernyataan>
WEND

```

#### 7. GOTO

Perintah GOTO digunakan untuk melakukan percabangan. Perintah GOTO hampir sama dengan perintah GOSUB, perbedaannya adalah pada perintah GOTO tidak memerlukan perintah RETURN, sehingga programnya tidak akan kembali lagi ke titik dimana perintah GOTO itu berada. Berikut ini sintak perintah GOTO :

```
GOTO label
```

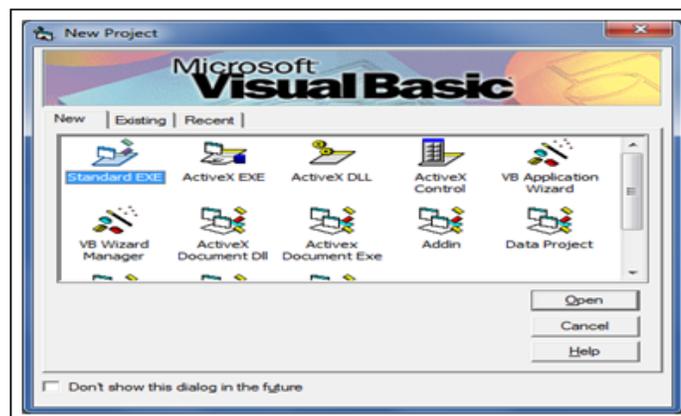
### 2.18 Microsoft Visual Basic 6.0

Bahasa *Basic* pada dasarnya adalah bahasa yang mudah dimengerti sehingga pemograman di dalam bahasa *basic* dapat dengan mudah dilakukan meskipun oleh orang yang baru belajar membuat program. *Visual Basic 6.0* merupakan bahasa pemograman yang sangat mudah dipelajari, dengan teknik pemograman *visual* yang memungkinkan penggunaannya untuk berkreasi dalam *Visual Basic 6.0* adalah *Form*, dimana pengguna dapat mengatur tampilan *form* kemudian dijalankan dalam *script* yang sangat mudah. Pemakaian *Visual Basic 6.0* ditandai dengan kemampuan *Visual Basic 6.0* untuk dapat berinteraksi dengan aplikasi lain di dalam sistem operasi *Windows* dengan komponen *ActiveX Control*.(Pamungkas,2000).

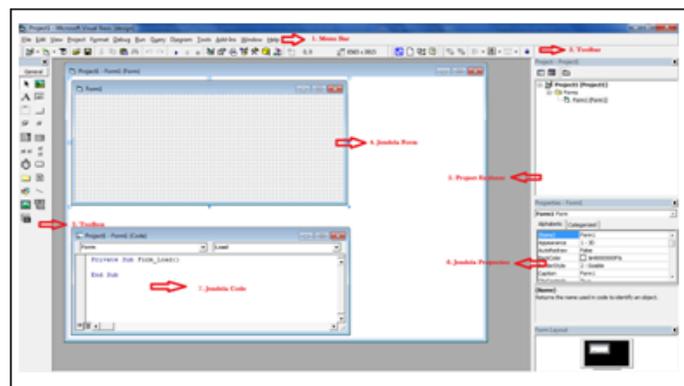
### 2.18.1 Menjalankan *Microsoft Visual Basic 6.0*

Seperti yang telah dibahas pada pembahasan sebelumnya, dikemukakan bahwa *Visual Basic 6.0* juga merupakan aplikasi *windows*. Oleh sebab itu kita harus menjalankannya dari dalam *windows*. Seperti biasa kita melakukannya dari menu *Start*.

- a. Klik tombol *mouse* pada menu *Start*.
- b. Pilih menu program *Microsoft Visual Basic 6.0*.
- c. Sebuah kotak *dialog* seperti pada gambar akan muncul saat kita memulai *Microsoft Visual Basic 6.0*. Disini kita dapat memilih jenis aplikasi yang akan kita buat dengan *Visual Basic 6.0*. Biasanya untuk membuat program aplikasi *standard*, kemudian pilihlah *Standard Exe*,
- d. Lalu klik tombol *open*.



Gambar 2.23 Kotak *Dialog New Project*

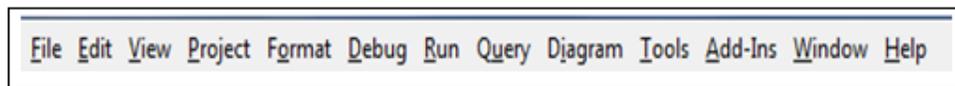


Gambar 2.24 Tampilan Dasar *Visual Basic 6.0*

Selanjutnya pilih *Standard EXE* dan klik tombol *Open*, maka kita akan melihat tampilan area kerja atau *IDE VB 6* sebagai di atas.

### 1. *Menu Bar*

Bagian ini menampilkan perintah-perintah yang dipergunakan selama menjalankan bekerja dilingkungan *Visual Basic 6.0*. Beberapa fungsi yang terdapat pada *menu bar* antara lain membuat dan menyimpan *project*, menyalin *file*, menjalankan program, mencetak, beberapa alat bantu pemrograman dan lainnya.



Gambar 2.25 *Menu Bar*

### 2. *Toolbar*

*Toolbar* adalah sebuah jendela yang berisi ikon-ikon yang fungsinya sama dengan *menu bar*, tetapi dapat digunakan dengan lebih cepat karena sebuah ikon mewakili suatu perintah tertentu. Contohnya adalah ikon *Open*, *Save*, *Copy*, *Paste*, *Undo* dan sejenisnya.

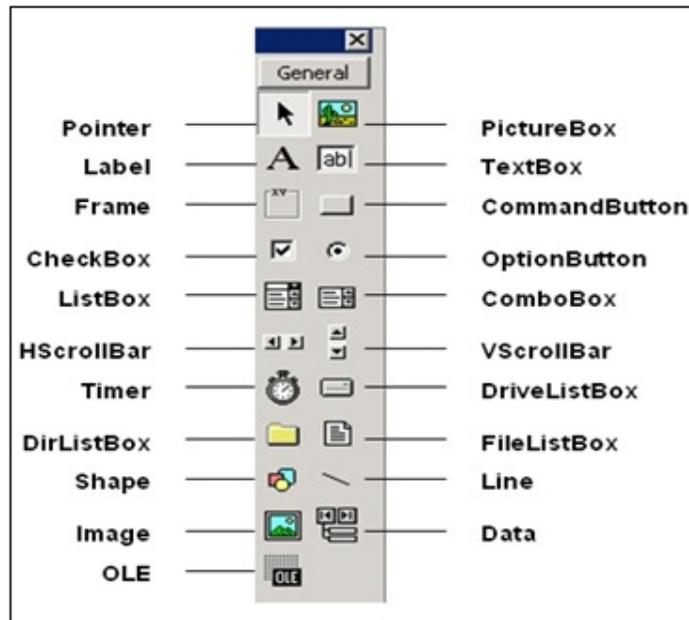


Gambar 2.26 *Toolbar*

### 3. *Toolbox*

*Toolbox* berisi tombol-tombol kontrol yang akan dipergunakan dalam membuat sebuah *form*. Komponen-komponen yang terdapat pada kontrol dapat ditambah dan dikurangi sesuai dengan kebutuhan program, penambahan dan pengurangan komponen yang dilakukan seiring dengan penambahan atau pengurangan pada *project components* dan *project references*.

Setelah kontrol yang terdapat pada *toolbox* diletakkan pada *form*, kontrol tersebut dapat diberikan nilai sesuai dengan fungsi yang akan di jalankan, nilai tersebut diletakkan di *windows properties*.



Gambar 2.27 Komponen dasar pada *ToolBox*

Adapun secara garis besar fungsi dari masing-masing kontrol-kontrol tersebut adalah sebagai berikut :

a. *Pointer*

Bukan merupakan suatu kontrol, menggunakan *icon* ini ketika anda ingin memilih kontrol yang sudah berada pada *form*.

b. *PictureBox*

*Icon* yang digunakan untuk menampilkan *image* dengan *format*: BMP, DIB (*bitmap*), ICO (*icon*), CUR (*cursor*), WMF (*metafile*), EMF (*enhanced metafile*), GIF, dan JPEG.

c. *Label*

*Icon* yang digunakan untuk menampilkan teks yang tidak dapat diperbaiki oleh pemakai.

d. *TextBox*

*Icon* yang mengandung string yang dapat diperbaiki oleh pemakai, dapat berupa satu baris tunggal atau banyak baris.

e. *Frame*

*Icon* yang digunakan sebagai container bagi kontrol lainnya.

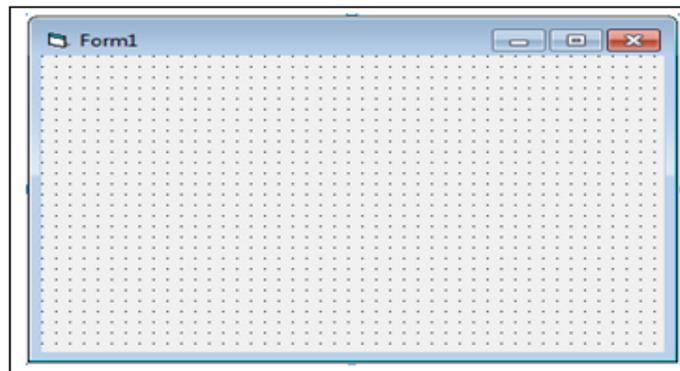
- f. *CommandButton*  
Merupakan kontrol yang ditemukan pada setiap *form*, dan digunakan untuk membangkitkan *event* proses tertentu ketika pemakai melakukan.
- g. *CheckBox*  
Digunakan untuk pemilihan yang isinya bernilai *yes or no, true or false*.
- h. *OptionButton*  
Digunakan lebih dari satu sebagai pilihan terhadap beberapa option yang hanya dapat dipilih satu.
- i. *ListBox*  
Memunyai sejumlah item, dan *user* dan juga dapat memilih lebih dari satu (bergantung pada *property Multi Select*).
- j. *ComboBox*  
Merupakan kombinasi dari *TextBox* dari suatu *ListBox* dimana pemasukkan data dapat dilakukan dengan mengetikkan maupun pemilihan.
- k. *HScrollBar*  
Befungsi untuk menambahkan kontrol batang penggulung *horizontal*.
- l. *VScrollBar*  
Befungsi untuk menambahkan kontrol batang penggulung *vertical*.
- m. *Timer*  
Merupakan kontrol *non-visual* yang digunakan untuk proses *background* yang diaktifkan berdasarkan *interval* waktu tertentu.
- n. *DriveListBox, DirListBox, dan FileListBox*  
Digunakan untuk membentuk *dialog box* yang berkaitan dengan *file*.
- o. *Shape dan Line*  
Befungsi untuk menampilkan bentuk seperti garis, persegi, bulatan, *oval*.
- p. *Image*  
Befungsi untuk menyerupai *image box*, tetapi tidak dapat digunakan sebagai kontainer bagi kontrol lainnya. Sesuatu yang perlu diketahui bahwa *image* menggunakan *resource* yang lebih kecil dibandingkan dengan *PictureBox*.
- q. *Data*  
Befungsi untuk menambahkan kontrol yang berupa *database*.

r. *OLE*

Digunakan sebagai tempat bagi program *eksternal* seperti *Microsoft Excel, Word*.

#### 4. *Jendela Form*

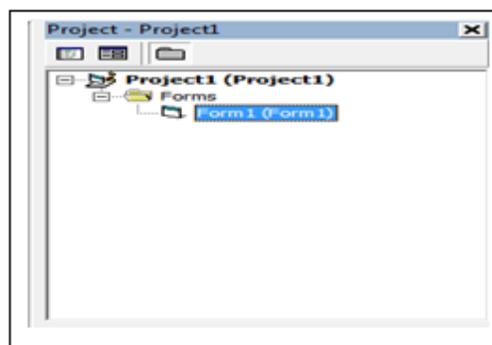
Jendela *form* dipergunakan untuk meletakkan berbagai kontrol yang dipergunakan sebagai antar muka dari program yang dibuat. Jendela *form* ini pada awalnya berukuran kecil, tetapi ukurannya dapat diubah-ubah sesuai dengan kebutuhan tampilan yang diperlukan.



Gambar 2.28 *Jendela Form*

#### 5. *Project Explorer*

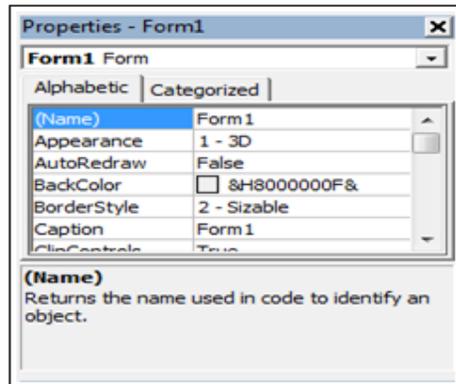
Jendela *Project Explorer* ini mengandung semua *file* didalam aplikasi visual basic. Setiap aplikasi dalam *visual basic 6.0* disebut dengan istilah *project* dan setiap *project* dapat mengandung lebih dari satu *file*.



Gambar 2.29 *Project Explorer*

## 6. Jendela *Properties*

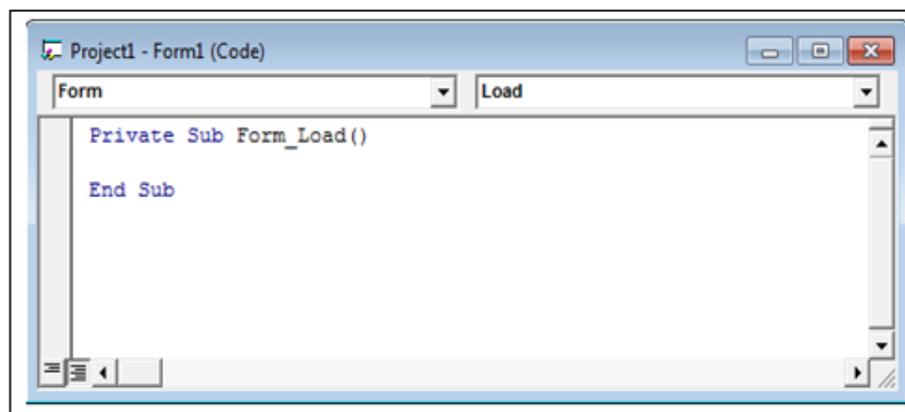
Jendela *properties* adalah jendela yang mengandung semua informasi mengenai objek yang terdapat pada aplikasi *visual basic 6.0*. *Property* adalah sebuah objek, misalnya sifat tampilan, warna, ukuran huruf dan sebagainya. Setiap objek sebagian besar memiliki jenis *property* yang sama, tetapi tidak menutup kemungkinan untuk berbeda.



Gambar 2.30 Jendela *Properties*

## 7. Jendela *Code*

Jendela *code* digunakan untuk menulis program, setiap kontrol dalam *form* dapat memiliki fungsi tertentu. Fungsi tersebut diwujudkan dalam deret perintah, perintah-perintah ini yang di tuliskan kedalam *window code editor*. *Visual Basic 6.0* menyediakan *auto list* untuk memberikan nilai yang terdapat pada suatu kontrol, sehingga mengurangi kesalahan penulisan program. Cara menggunakannya adalah dengan mengklik ganda kontrol yang ada.



Gambar 2. 31 Jendela *Code*

### 2.18.2 Operator *Visual Basic 6.0*

Operator adalah suatu komputer atau tanda untuk menyatakan suatu operasi atau proses. Pada dasarnya kompone dengan *ALU*-nya (*Aritmethic Logical Unit*), mempunyai dua macam operator yaitu operator *Aritmatika* dan operator logika (perbandingan). Operator-operator didalam *Visual Basic 6.0* antara lain :

Tabel 2.10 Operator-operator Aritmatika didalam *Visual Basic 6.0*

Jenis	Operator	Kegunaan
<i>Aritmatika</i>	+	Penjumlahan
	-	Pengurangan
	*	Perkalian
	/	Pembagian
	^	Pangkat
	<i>mod</i>	Sisa Pembagian
	\	Hasil Bulat Pembagian
	&	Penggabungan String
<i>Relasi</i>	=	Sama dengan
	<	Lebih Kecil
	<=	Lebih Kecil atau sama dengan
	>	Lebih Besar
	>=	Lebih Besar atau sama dengan
	<>	Tidak Sama
<i>Logika</i>	<i>AND</i>	Dua kondisi harus dipenuhi
	<i>OR</i>	Dari dua kondisi, akan benar bila ada salah satu atau lebih kondisi yang dipenuhi
	<i>NOT</i>	Invers dari kondisi yang diberikan

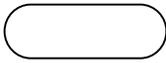
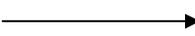
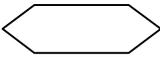
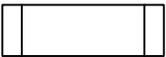
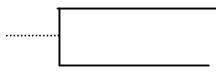
Penjelasan lebih lanjut mengenai operator ada dibagian ekspresi, karena operator ini merupakan bagian dari *ekspresi*. Pemakaian operator ini sangat berhubungan dengan penulisan rumus didalam pemograman.

### 2.19 *Flowchart*

*Flowchart* adalah cara penyajian *visual* aliran data melalui sistem informasi, *Flowchart* dapat membantu menjelaskan pekerjaan yang saat ini dilakukan dan bagaimana cara meningkatkan atau mengembangkan perkerjaan tersebut. Dengan

menggunakan *flowchart* dapat juga membantu untuk menemukan sebuah proses, selama garis digambarkan secara proses berakhir dan proses selanjutnya dimulai. Adapun simbol-simbol dari *flowchart* adalah sebagai berikut :

Table 2.11 *Flowchart*

Bagan	Nama	Fungsi
	<i>Terminator</i>	Awal atau akhir program
	<i>Flow</i>	Arah aliran program
	<i>Preparation</i>	Inisialisasi/Pemberian Nilai Awal
	<i>Process</i>	Proses/pengolahan data
	<i>Input/Output Data</i>	<i>Input/Output Data</i>
	<i>Sub Program</i>	Sub Program
	<i>Decision</i>	Seleksi atau kondisi
	<i>On Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> pada halaman yang sama
	<i>Off Page Connector</i>	Penghubung bagian-bagian <i>flowchart</i> pada halaman yang berbeda
	<i>Comment</i>	Tempat komentar tentang suatu proses.

*Flowchart* merupakan logika atau urutan instruksi program dalam suatu diagram". Diagram alur dapat menunjukkan secara jelas arus pengendalian *Algoritma*, yaitu bagaimana rangkaian pelaksanaan kegiatan. Adapun tujuan dari pembuatan *flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi dan jelas dengan menggunakan simbol-simbol standar. (Fazri,2010).