

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Pengertian *Wireless Sensor Network* (WSN)**

*Wireless Sensor Network* (WSN) adalah sebuah sistem berbasis jaringan *wireless* yang bentuk data-data digital pada dunia komputer. Secara umum *Wireless Sensor Network* (WSN) atau sensor jaringan nirkabel merupakan suatu peralatan sistem *embedded* (tertanam) yang di dalamnya terdapat satu atau lebih sensor dan dilengkapi dengan peralatan sistem komunikasi atau bisa disebut sensor yang bekerja tanpa menggunakan kabel. Sensor dalam peralatan ini digunakan untuk menangkap atau mengumpulkan suatu informasi yang sesuai dengan karakteristik dari sensor tersebut. Informasi yang telah dikumpulkan berupa sinyal analog di ubah dalam bentuk sinyal digital dan kemudian transmisikan ke suatu *node* (titik) melalui media tanpa kabel atau *wireless* seperti *wifi*, *bluetooth*, *infrared*, dan lainnya.

Masing-masing titik dalam WSN dilengkapi dengan *radio transceiver* sebagai *node* penerima atau pengirim atau juga perangkat pendukung lainnya. Sehingga *Wireless Sensor Network* (WSN) juga dikenal sebagai sebuah sistem yang terdiri dari sejumlah besar *low-cost* sensor yang berukuran kecil dan tersebar pada sebuah area yang sangat luas dengan satu *node* penampung untuk mengumpulkan hasil proses pembacaan sensor *node* lainnya. Sensor di WSN sebuah memiliki berbagai tujuan, fungsi, dan kemampuan. (Sohrabi, 2007:2)

##### **2.1.1 Arsitektur *Wireless Sensor Network* (WSN)**

Dalam WSN terdapat komponen jaringan :

1. Motes Sensor ( Perangkat *Field* ) merupakan perangkat dasar yang dipasang proses dan dapat merouting paket ke perangkat lain. Sebuah *router* adalah tipe khusus dari perangkat dasar itu tidak memiliki sensor proses atau peralatan kontrol dan dengan demikian tidak *inteface* dengan proses itu sendiri.

2. *Gateway* atau akses point. Sebuah *gateway* akan memungkinkan komunikasi antara host aplikasi dan bidang perangkat.
3. *Manager Jaringan*. Sebuah manajer jaringan bertanggung jawab untuk konfigurasi jaringan, penjadwalan komunikasi antar perangkat, manajemen tabel routing dan pemantauan dan pelaporan kesehatan jaringan. (*Kalita, 2009 Vol :1*)

*Node Sensor* pada *Wireles Sensor Network* (WSN) memiliki kemampuan untuk merutekan data yang dikumpulkan ke *node* lain yang berdekatan. Masing-masing *node* dalam jaringan sensor nirkabel biasanya dilengkapi dengan radio *tranceiver* atau alat komunikasi *wireless* lainnya, mikrokontroler dan sumber tegangan. Data dikirimkan melalui transmisi radio akan diteruskan menuju *Base Station* (BS) atau *sink node* yang merupakan penghubung antara *node* sensor dan *user*. Informasi tersebut dapat diakses melalui berbagai *platform* seperti koneksi internet atau satelit sehingga memungkinkan user untuk dapat mengakses secara *realtime* melalui *remote server*. (*Jason, 2003*)

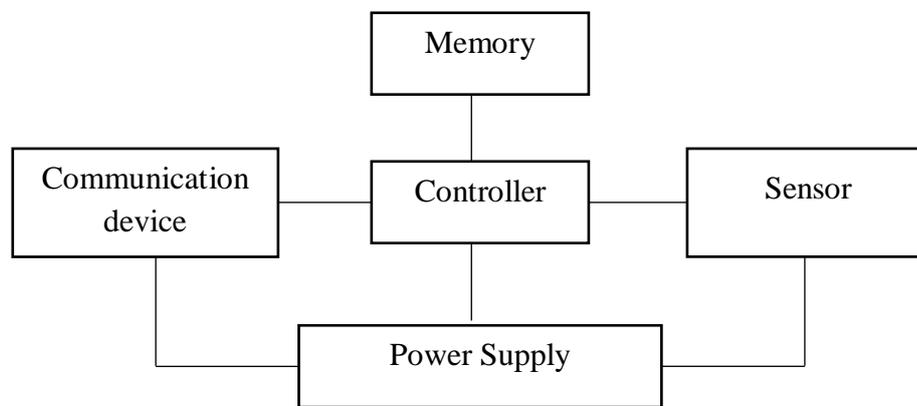
BTS adalah salah satu atau lebih komponen dari WSN dengan lebih banyak sumber daya komputasi, energi dan komunikasi. Mereka bertindak sebagai *gateway* antara *node* sensor dan pengguna akhir karena mereka data yang biasanya maju dari WSN pada ke server. Komponen khusus lainnya di *routing* yang jaringan berbasis *router*, yang dirancang untuk menghitung, menghitung dan mendistribusikan tabel *routing*. (*Sugiarto, 2009 Vol:3*).

### **2.1.2 Komponen-Komponen Penyusun *Wireless Sensor Metwork***

Sebuah WSN umumnya terdiri atas bagian-bagian berikut :

1. *Transceiver* ataupun *Communication device* Berfungsi untuk menerima / mengirim data dengan menggunakan protokol IEEE 802.15.4 kepada *device* lain seperti *concentrator*, modem *Wifi*, dan modem RF.
2. *Mikrokontroler*, Berfungsi untuk melakukan fungsi perhitungan, mengontrol dan memproses *device-device* yang terhubung dengan mikrokontroler.
3. *Power Supply*, Berfungsi sebagai sumber energi bagi sistem WSN secara keseluruhan.

4. *External Memory*, Berfungsi sebagai tambahan memory bagi sistem WSN, pada dasarnya sebuah unit mikrokontroler memiliki unit memory sendiri.
5. *Sensor*, Berfungsi untuk men-sensing besaran-besaran fisis yang hendak diukur. Sensor adalah suatu alat yang mampu untuk mengubah suatu bentuk energi ke bentuk energi lain, dalam hal ini adalah mengubah dari energi besaran yang diukur menjadi energi listrik yang kemudian diubah oleh ADC menjadi deretan pulsa terkuantisasi yang kemudian bisa dibaca oleh mikrokontroler. (Sugiarto, 2009)



**Gambar 2.1** Komponen- Komponen Penyusun Node dalam WSN

(Sumber : Sugiarto, 2009)

## 2.2 Mikrokontroler

### 2.2.1 Gambaran Umum Mikrokontroler

Mikrokontroler merupakan suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data. Mikrokontroler sebuah sistem komputer kecil yang lengkap (*Special Purpose computers*) dalam satu IC berisi CPU, memori, *timer*, saluran komunikasi serial dan paralel, Port input dan Port output, ADC, Mikrokontroler digunakan untuk suatu tugas dan menjalankan suatu program. Mikrokontroler lebih dari sekedar mikroprosesor karena sudah terdapat atau berisikan ROM (*Read Only Memory*), RAM (*Read-Write Memory*), beberapa bandar masukan maupun keluaran, dan beberapa *peripheral* seperti pencacah/pewaktu, ADC (*Analog Dto Digital Converter*), DAC (*Digital to Analog Converter*) dan serial komunikasi.

Secara teknis hanya ada dua macam dari mikrokontroler yaitu RISC dan CISC dan masing-masing mempunyai keturunan atau keluarga sendiri.

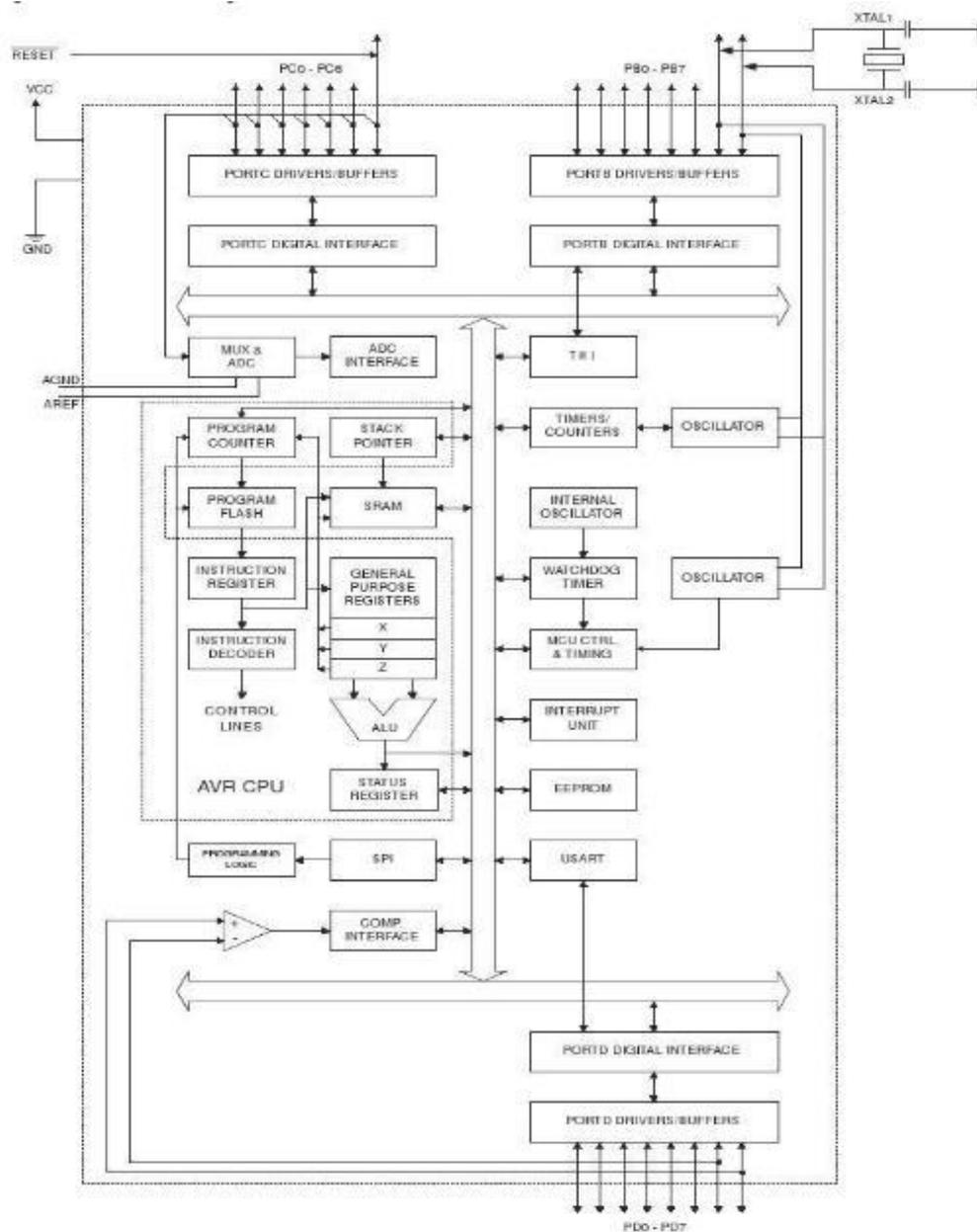
1. RISC (*Reduce Instruction Set Computer*), Instruksi ini bebas tapi memiliki fasilitas yang lebih banyak.
2. CISC (*Complex Instruction Set Computer*), instruksi ini bisa dikatakan lebih lengkap tapi dengan fasilitas secukupnya.

Sedangkan dalam hal aplikasi, sistem mikrokontroler memiliki karakteristik sebagai berikut :

1. Memiliki program khusus yang disimpan dalam memori untuk aplikasi tertentu, tidak seperti PC yang multifungsi karena mudahnya memasukan program. Program mikrokontroler relatif lebih kecil dari pada program-program pada PC.
2. Konsumsi daya kecil
3. Rangkaian sederhana dan sama serta Unit I/O yang sederhana.

### **2.2.2 Mikrokontroler ATMEGA 8**

ATMEGA 8 adalah mikrokontroler CMOS 8-bit daya rendah berbasis arsitektur RISC yang ditingkatkan. Kebanyakan instruksi dikerjakan pada satu siklus *clock*, ATMEGA 8 mempunyai *throughput* mendekati 1 MPS per MHz membuat disain dari sistem untuk mengoptimasi konsumsi daya versus kecepatan proses. (*Atmel, 2013:3*)



**Gambar 2.2** Blok Diagram Microcontroler ATmega 8

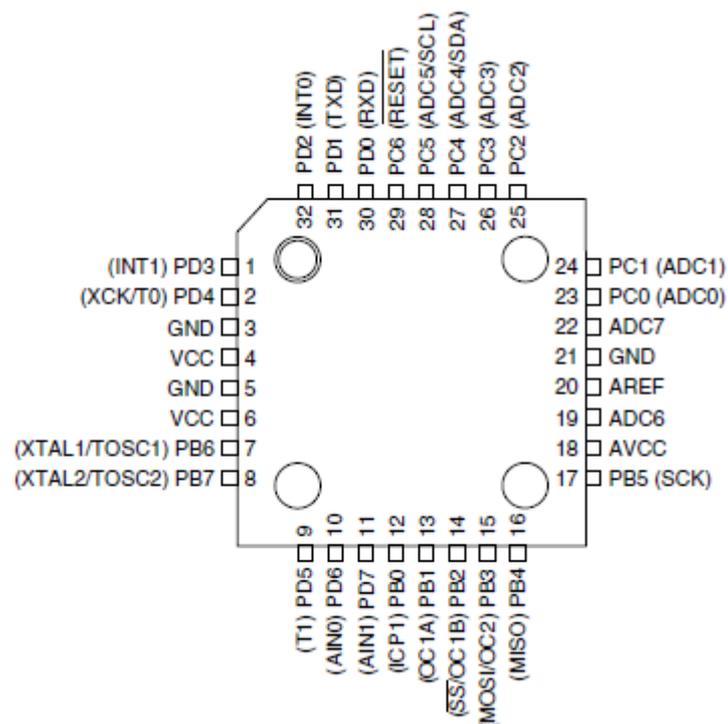
( Sumber: *Atmega, 2013:3*)

Penggunaan rangkaian mikrokontroler ATmega8 ada dua pilihan, dengan menggunakan *board ATmega8 development board* yang sudah ada dipasaran atau dengan membuat sendiri rangkaian mikrokontroler tersebut. Jika menggunakan rangkaian mikrokontroler yang sudah tersedia dipasaran maka akan

memepersingkat waktu pembuatan sistem, karena hanya tinggal membeli rangkaian berupa kit dan hanya tinggal menggunakannya.

mikrokontroler ATmega8 menggabungkan banyak instruksi set dengan 32 tujuan umum dari kerja register. Semua 32 register secara langsung terhubung ke *Unit Arithmetic Logic (ALU)* memungkinkan dua register independen untuk diakses dalam satu instruksi tunggal dieksekusi dalam satu siklus *clock*. Arsitektur dihasilkan kode yang lebih efisien sementara mencapai throughput hingga sepuluh kali lebih cepat dari mikrokontroler CISC Konvensional. Tujuan utama dari arsitektur CISC adalah melaksanakan suatu perintah cukup dengan beberapa baris bahasa mesin sesedikit mungkin. (Putra, 2010:23)

Susunan pin – pin dari IC mikrokontroler ATMEGA 8 diperlihatkan pada gambar dibawah ini. IC ini tersusun atas pin yang memiliki beberapa fungsi tertentu.



**Gambar 2.3** Susunan Pin Microcontroller ATMEGA8

(Sumber : Atmel, 2013:2)

ATMega8 memiliki pin yang masing – masing pin tersebut memiliki fungsi yang berbeda – beda baik sebagai port ataupun sebagai fungsi yang lain. Berikut akan dijelaskan tentang kegunaan dari masing – masing kaki pada ATMega8.

a. VCC

VCC Merupakan supply tegangan untuk digital.

b. GND

GND Merupakan *ground* untuk semua komponen yang membutuhkan *grounding*.

c. Port B

Port B Adalah 8 buah pin mulai dari pin B.0 sampai dengan pin B.7. Tiap pin dapat digunakan sebagai input dan juga output. Port B merupakan sebuah 8-bit bit-directional I/O port dengan *inernal pull-up* resistor. Sebagai input, pin – pin yang terdapat pada port B yang secara eksternal diturunkan, maka akan mengeluarkan arus jika *pull-up* resistor diaktifkan. Jika ingin menggunakan tambahan kristal, maka cukup untuk menghubungkan kaki dari kristal ke kaki pada pin port B. Namun jika tidak digunakan, maka cukup untuk dibiarkan saja. Pengguna kegunaan dari masing – masing kaki ditentukan dari *clock fuse setting*-nya.

d. Port C

Port C Port C merupakan sebuah *7-bit bi-directional* I/O yang di dalam masing – masing pin terdapat *pull-up* resistor. Jumlah pin-nya hanya 7 buah mulai dari C.0 sampai dengan pin C.6. Sebagai keluaran / output, port C memiliki karakteristik yang sama dalam hal kemampuan menyerap arus ( *sink* ) ataupun mengeluarkan arus ( *source* ).

e. Reset/PC6

Jika RSTDISBL *Fuse* diprogram, maka PC6 akan berfungsi sebagai pin I/O. Untuk diperhatikan juga bahwa pin ini memiliki karakteristik yang berbeda dengan pin – pin yang terdapat pada port C. Namun jika RSTDISBL *Fuse* tidak diprogram, maka pin ini akan berfungsi sebagai *input reset*. Dan jika level tegangan yang masuk ke pin ini rendah dan pulsa yang ada lebih pendek dari

pulsa minimum, maka akan menghasilkan suatu kondisi reset meskipun *clock*-nya tidak berkerja.

f. Port D

Port D merupakan 8-bit bi-directional I/O dengan *internal pull-up* resistor. Fungsi dari port ini sama dengan port – port yang lain. Hanya saja pada port ini tidak terdapat kegunaan-kegunaan yang lain. Pada port ini hanya berfungsi sebagai masukan dan keluaran saja atau biasa disebut dengan I/O.

g. AVCC

Pada pin ini memiliki fungsi sebagai *power supply* tegangan untuk ADC. Untuk pin ini harus dihubungkan secara terpisah dengan VCC karena pin ini digunakan untuk analog saja. Bahkan jika ACD pada AVR tidak digunakan, tetap saja disarankan untuk menghubungkan secara terpisah dengan VCC. Cara menghubungkan AVCC adalah melewati *low-pass filter* setelah itu dihubungkan dengan VCC.

h. AREF

Merupakan pin referensi analog jika menggunakan ADC. Pada AVR status Register mengandung beberapa informasi mengenai hasil dari kebanyakan hasil eksekusi intruksi aritmatik. Informasi ini dapat digunakan untuk *altering* arus program sebagai kegunaan untuk meningkatkan performa pengoperasian. Perlu diketahui bahwa register ini di-*update* setelah semua operasi ALU (*Arithmetic Logic Unit*). Hal tersebut seperti yang telah tertulis dalam datasheet khususnya pada bagian *Intruction Set Reference*. Dalam hal ini untuk beberapa kasus dapat membuang kebutuhan penggunaan instruksi perbandingan yang telah didedikasikan serta dapat menghasilkan peningkatan dalam hal kecepatan dan kode yang lebih sederhana dan singkat. Register ini tidak secara otomatis tersimpan ketika memasuki sebuah rutin interupsi dan juga ketika menjalankan sebuah perintah setelah kembali dari interupsi. Namun hal ini harus dilakukan melalui *software*.

i. Bit7 (1)

Merupakan bit *Global Interrupt Enable*. Bit ini harus di-*set* supaya semua perintah interupsi dapat dijalankan. Untuk fungsi interupsi individual akan

dijelaskan pada bagian yang lain. Jika bit ini di-reset, maka semua perintah interupsi baik yang secara individual maupun yang secara umum akan diabaikan. Bit ini akan dibersihkan atau *cleared* oleh hardware setelah sebuah interupsi dijalankan dan akan di-*set* kembali oleh perintah RETI. Bit ini juga dapat di-*set* dan di-*reset* melalui aplikasi dengan instruksi SEI dan CLI.

- j. Bit 6 (T)
 

Merupakan bit *Copy Storage*. Instruksi *bit Copy Instruction* BLD (*Bit Load*) dan BST (*Bit Store*) menggunakan bit ini sebagai asal atau tujuan untuk bit yang telah dioperasikan. Sebuah bit dari sebuah register dan Register File dapat disalin ke dalam bit ini dengan menggunakan intruksi BST, dan sebuah bit di dalam bit ini dapat disalin ke dalam sebuah bit di register pada Register File dengan menggunakan perintah BLD.
- k. Bit 5 (H)
 

Merupakan bit *Half Carry Flag*. Bit ini menandakan sebuah *Half Carry* dalam beberapa operasi aritmatika. Bit ini berfungsi dalam aritmatik BCD
- l. Bit 4 (S)
 

Merupakan Sign bit. Bit ini selalu merupakan sebuah eksklusif di antara *Negative Flag* (N) dan *Two's Complement Overflow Flag* (V).
- m. Bit 3 (V)
 

Merupakan bit *Two's Complement Overflow Flag*. Bit ini menyediakan fungsi aritmatika dua komplemen
- n. Bit 2 (N)
 

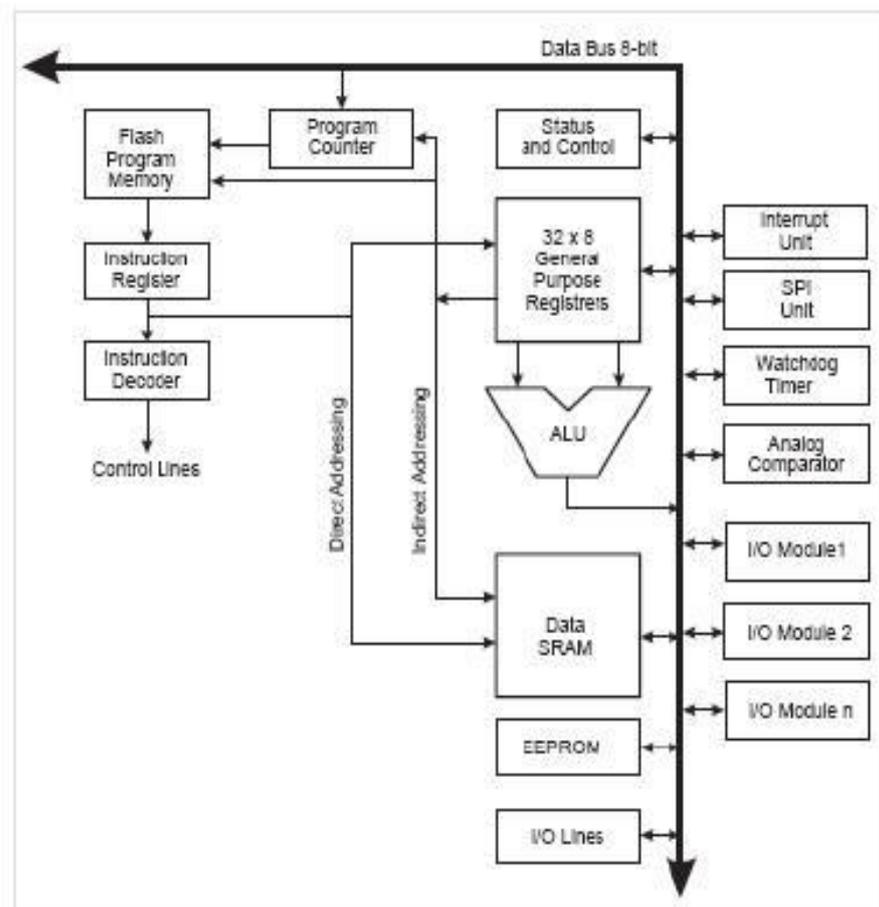
Merupakan bit *Negative Flag*. Bit ini menyediakan sebuah hasil negative di dalam sebuah fungsi logika atau aritmatika.
- o. Bit 1 (Z)
 

Merupakan bit *Zero Flag*. Bit ini mengindikasikan sebuah hasil nol "0" dalam sebuah fungsi aritmatika atau logika.
- p. Bit 0 (C)
 

Merupakan bit *Carry Flag*. Bit ini mengindikasikan sebuah Carry atau sisa dalam sebuah fungsi aritmatika atau logika.

(Atmel, 2013:5)

### 2.2.3 Arsitektur Mikrokontroler ATmega 8



**Gambar 2.4** Arsitektur *Microcontroller* ATmega8

(Sumber: *Atmel*, 2013:9)

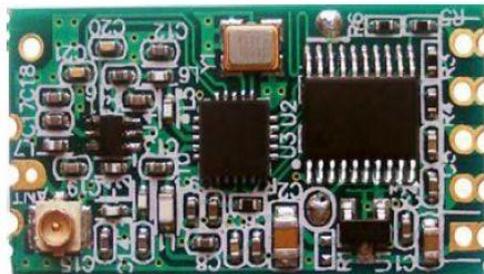
### 2.2.4 FITUR Mikrokontroler ATmega 8

- Saluran I/O sebanyak 23 buah terbagi menjadi 3 port.
- ADC sebanyak 6 saluran dengan 4 saluran 10 bit dan 2 saluran 8 bit
- Tiga buah *timer counter*, dua diantaranya memiliki fasilitas pembandingan.
- CPU dengan 32 buah register
- Watchdog timer dan *oscillator* internal.
- SRAM sebesar 1K byte.
- Memori flash sebesar 8K Bytes *system Self-programable Flash*
- Unit interupsi internal dan eksternal.
- Port antarmuka SPI

- j) EEPROM sebesar 512 byte.
- k) Port USART ( *Universal Synchronous and Asynchronous Serial Receiver and Transmitter* ) untuk komunikasi serial.  
(Atmel, 2013 : 1)

### 2.3 ` Wireless Serial Port HC12

Modul komunikasi port serial nirkabel HC-12 adalah modul transmisi data nirkabel multibannel generasi baru. Band frekuensi kerja nirkabelnya adalah 433,4-473.0MHz, beberapa saluran dapat diatur, dengan loncatan 400 KHz, dan totalnya ada 100 saluran. Daya maksimum transmisi modul adalah 100mW (20dBm), sensitivitas penerimaan adalah -117dBm pada baud rate 5.000bps di udara, dan jarak komunikasi 1.000 m di ruang terbuka.



**Gambar 2.5** *Wireless Serial Port HC12*

(*DatasheetCatalog.com*)

Fitur Produk :

1. Transmisi nirkabel jarak jauh (1.000 m di ruang terbuka / baud rate 5.000 bps di udara)
2. Rentang frekuensi kerja (433,4-473.0MHz, hingga 100 saluran komunikasi)
3. Daya pemancar maksimum 100mW (20dBm) (8 roda gigi dapat diatur)
4. Tiga mode kerja, beradaptasi dengan berbagai situasi aplikasi
5. Built-in MCU, melakukan komunikasi dengan perangkat eksternal melalui port serial

6. Jumlah byte yang dikirim tidak terbatas satu kali
7. Perbarui versi perangkat lunak melalui port serial

## **2.4 Pemograman BASCOM AVR**

### **2.4.1 Bahasa Basic Pada Bascom AVR**

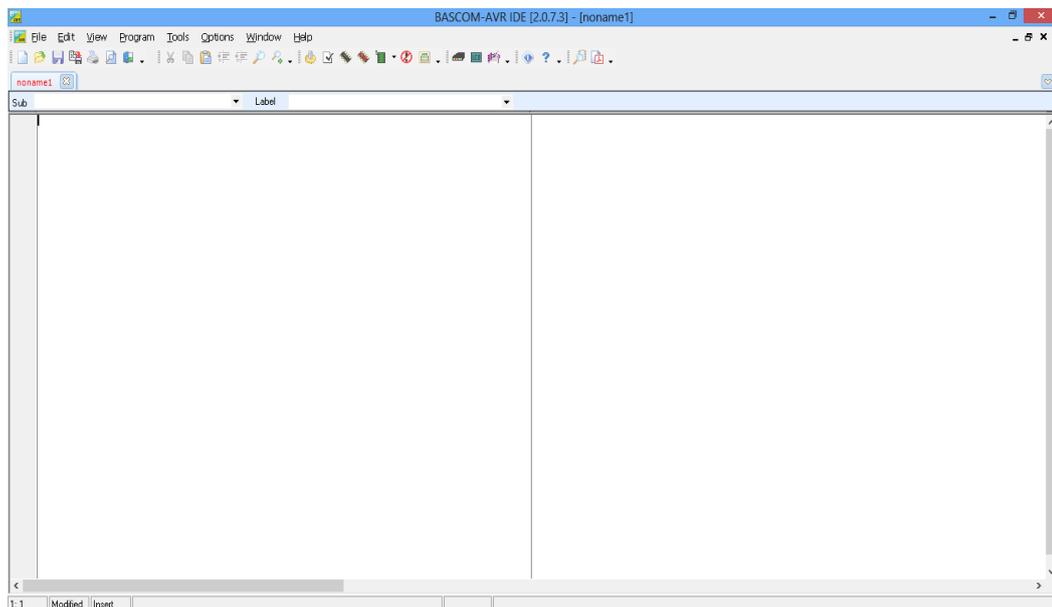
Bahasa Basic adalah salah satu bahasa yang banyak digunakan untuk aplikasi mikrokontroller karena kemudahan dan kompatible terhadap mikrokontroller jenis AVR dan didukung oleh *Compiler software* berupa BASCOM AVR. Program penerjemah dari bahasa *Assembly* kedalam bahasa mesin disebut *Assambler*. Sedangkan kompiler menerjemahkan bahasa tingkat tinggi kedalam bahasa assambley. Intrepter mempunyai pengertian yang mirip dengan kompiler. Keuntungan interpter adalah user dapat cepat memperoleh tanggapan. Dengan menulis satu baris perintah, lalu menulis run pemakai bisa langsung mengetahui hasilnya. Pada saat kompilasi, kompiler tidak menerjemahkan semua perintah program sumber menjadi objek code, tetapi kompiler akan menyediakan subroutine khusus yang hanya akan digunakan pada saat program hasil kompilasi dijalankan. Kumpulan subroutine tersebut dinamakan *run time library*. (Sedyono, 2007: 3-4)

### **2.4.2 Sofware Basic Compiler AVR**

Bascom-AVR adalah program *basi compiler* berbasis windows untuk mikrokontroler keluarga AVR merupakan pemograman bahasa tingkat tinggi “BASIC” yang dikembangkan dan dikeluarkan oleh MCS elektronika sehingga dapat dengan mudah dimengerti atau diterjemahkan.

Dalam program BASCOM AVR terdapat beberapa kemudahan, untuk membuat program software ATMEGA 8, seperti program simulasi yang sangat berguna untuk melihat, simulas program yang telah kita buat sebelum program tersebut kita download ke IC atau ke Mikrokontroler. (Sedyono, 2007:5)

Ketika Program BASCOM AVR dijalankan dengan mengklik icon BASCOM AVR, maka akan tampil:



**Gambar 2.6** Tampilan jendela program BASCOM AVR

**Tabel 2.1** Fungsi-Fungsi Submenu pada Menu File

Icon	Nama	Fungsi	Shortcut
	<i>File New</i>	Membuat file baru	Ctrl+N
	<i>Open File</i>	Untuk membuka file	Ctrl+N
	<i>File Close</i>	Untuk menutup program yang telah dibuka	Ctrl+O
	<i>File Save</i>	Untuk menyimpan file	Ctrl+S
	<i>Save As</i>	Menyimpan dengan nama 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>Syntax Check</i>	Untuk memeriksa kesalahan bahasa	Ctrl+F7
	<i>Show result</i>	Untuk menampilkan hasil kompilasi program	Ctrl+W

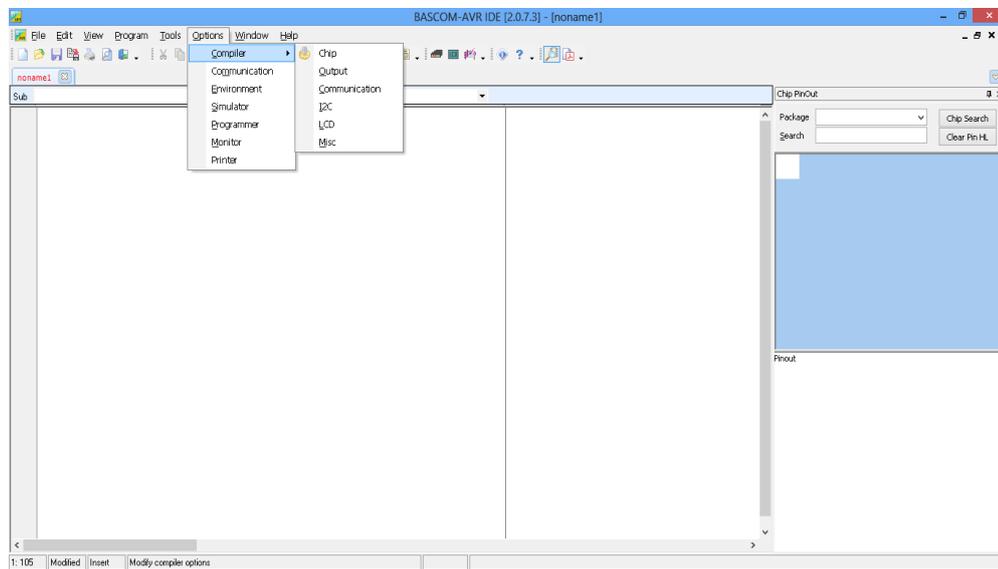
**Tabel 2.2** Informasi yang akan ditampilkan ditampilkan Menu show Result

Info	Keterangan
<i>Compiler</i>	Versi dari kompiler yang digunakan
<i>Processor</i>	Menampilkan target prosesor yang dipilih

<i>Date and Time</i>	Tanggal dan waktu kompilasi
<i>Baud rate dan xtal</i>	Baudrate yang dipilih dan kristal yang digunakan up
<i>Error</i>	Nilai baud yang diset dengan nilai baud sebenarnya
<i>Flash Used</i>	Persentase flash ROM yang terisi program.
<i>Stack Start</i>	Lokasi awal stack pointer memori
RAM Star	Lokasi awal eksternal RAM
LCD Mode	Mode LCD yang digunakan, 4 bit atau 8 bit

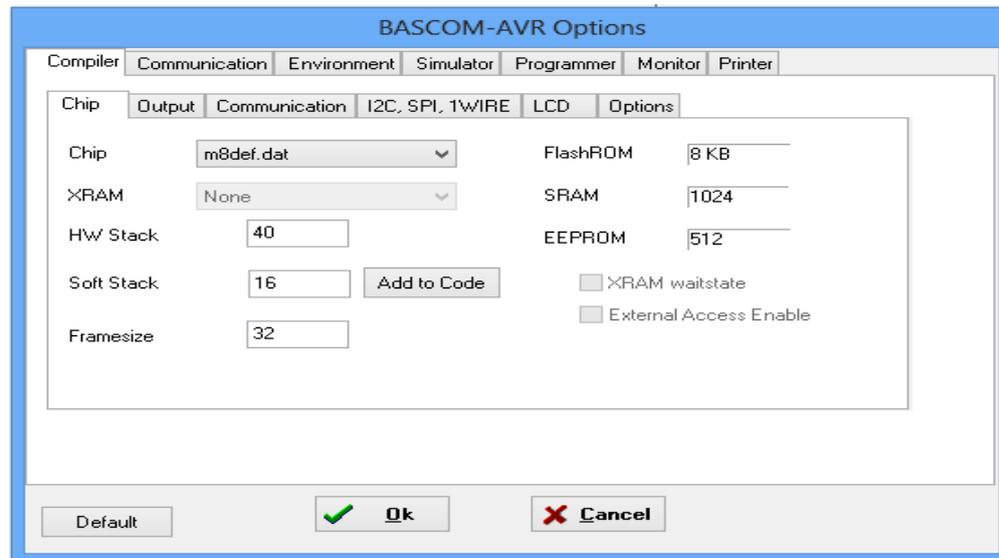
Pada bagian menu option didalam program BASCOM AVR terdapat

- Compiler, digunakan untuk mensetting chip, output, comunication 12C dan LCD
- Communication, digunakan untuk mensetting komunikasi mikrokontroler.
- Simulator, digunakan untuk mensetting simulasi pada BASCOM AVR
- Programmer, digunakan untuk mensetting downloder programmer t=yang akan digunakan.
- Monitor, untuk mensetting tampilan
- Printer, digunakan untuk mensetting printer yang digunakan



**Gambar 2.7** Bar Pada Menu Options

Fungsi Luas Menu Compiler Bascom AVR menyediakan pilihan untuk memodifikasi pilihan-pilihan pada kompilasi. Dengan memilih menu Compiler maka jendela berikut akan ditampilkan:

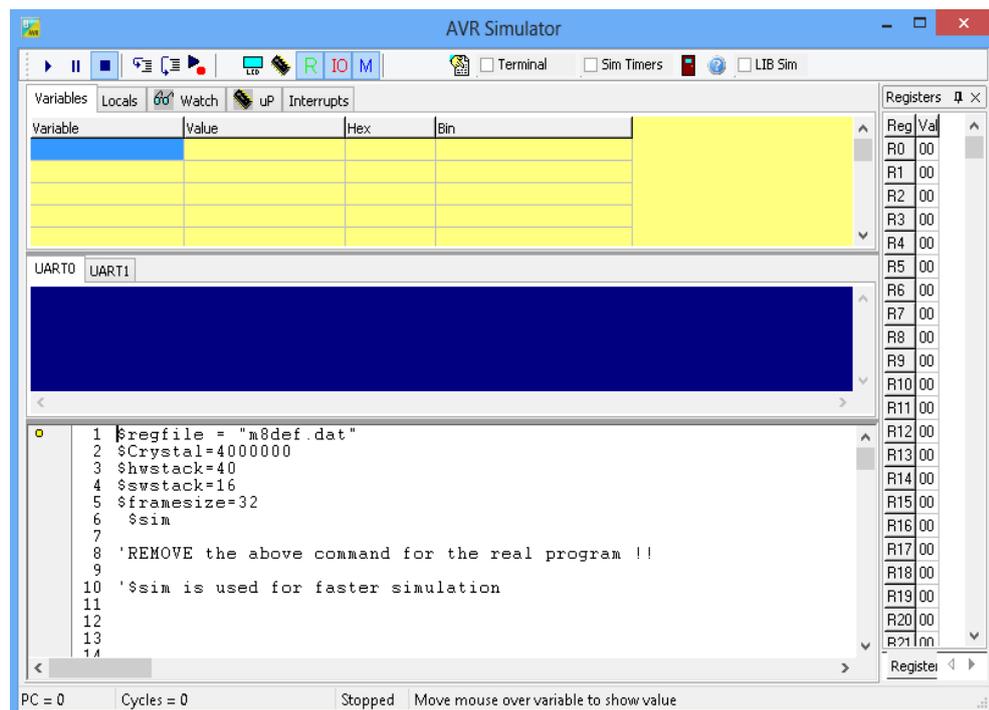


**Gambar 2.8** Jendela Option

**Tabel 2.3** Keterangan dari Jendela Option

TAB Menu	OPTION	Keterangan
<i>Chip</i>	<i>Chip</i>	Mikrokontroler yang digunakan, sebagai contoh m16.dat untuk ATMEGA16
	<i>XRAM</i>	Jika menggunakan eksternal RAM nilai bisa ditampilkan
	<i>HW Stack</i>	Stack memory Hardware, sriap Gosub membutuhkan 2 byte. Jika menggunakan interupsi, naikan nilainya
	<i>Soft stack</i>	Stack Software, nilai defaultnya 8
	FlashROM	Nilai FlashROM Chip yang dipilih
	SRAM	Nilai RAM Internal Chip yang dipilih
	EEPROM	Nilai EEPROM chip yang terpilih
	<i>Communication Baudrate 0</i>	Nilai Baudrate yang digunakan dalam komunikasi serial
	Frekuensi	Nilai Osilator yang digunakan
	<i>Error</i>	Error antara baudrate yang dipilih dengan nilai sebenarnya, hal ini tergantung pada osilator yang dipilih
	12C,SPI, Wire SDA	1 Pin yang berfungsi untuk data serial dalam komunikasi 12C
	SCL	Pin yang berfungsi untuk data clock dalam komunikasi 12C
	<i>1 Wire</i>	Pin yang digunakan untuk komunikasi 1 wire
	SPI	Pin yang digunakan untuk komunikasi serial sinkron

BASCOM AVR menyediakan pilihan yang dapat mensimulasikan program. Program simulasi ini bertujuan untuk menguji suatu aplikasi yang dibuat dengan penggerak LED yang ada pada layar simulasi dan dapat juga langsung dilihat pada LCD, jika membuat aplikasi yang berhubungan dengan LCD. Agar dapat menjalankan simulator ini, file DBG dan OBJ harus dipilih pada menu Options Copmpiler Output. Tampilan simulasi adalah sebagai berikut: ( Afrie:2010)



**Gambar 2.9** Tampilan Simulasi BASCOM AVR

Tekan tombol  untuk memulai simulasi dan untuk memberhentikan simulasi atau menahan proses simulasi gunakan tombol sebelahnya .  Layar biru ditengah merupakan simulasi layar monitor ketika menggunakan komunikasi serial. Untuk dapat mengamati perubahan-perubahan nilai register atau variabel selama program berjalan, simulator ini menyediakan beberapa jendela, antara lain :

1. Variabel

Jendela ini berisi tabel yang berfungsi untuk mengamati nilai variabel-variabel yang digunakan dalam program yang sedang disimulasikan. Untuk menambahkan variabel klik ganda pada kolom variabel maka daftar variabel akan ditampilkan, klik variabel yang ingin diamati.



## 2. **IF – THEN – ELSE**

Perintah *IF – THEN – ELSE* digunakan untuk menguji dua keadaan (benar ataupun salah) dan menentukan tindakan yang sesuai dengan keinginan.

Perintahnya :

```

If <Keadaan> Then
    <Perintah_1>
Else
    <Perintah_2>
End If

```

## 3. **IF – THEN – ELSEIF**

Perintah *IF – THEN – ELSEIF* digunakan untuk menguji lebih dari satu keadaan dan menentukan tindakan yang sesuai dengan keinginan.

Perintahnya :

```

If <Keadaan_1> Then
    <Perintah_1>
Elseif <Keadaan_2> Then
    <Perintah_2>
Elseif <Keadaan_3> Then
    <Perintah_3>
.....
.....
End If

```

## 4. **SELECT – CASE**

Perintah *SELECT – CASE* digunakan untuk pengujian keadaan yang banyak sehingga penulisan program menjadi lebih sederhana.

Perintahnya :

```

Select case <Nama_variabel>
    Case 1 : <Perintah_1>

```

*Case 2* : <Perintah\_2>

*Case\_3* : <Perintah\_3>

.....

.....

*End Select*

## 5. **DO – LOOP**

Perintah *DO – LOOP* merupakan perintah untuk perulangan yang digunakan untuk melakukan perulangan program selama suatu kondisi telah terpenuhi.

Perintahnya :

*Do*

<Pernyataan>

*Loop*

## 6. **FOR – NEXT**

Perintah *FOR – NEXT* merupakan perintah untuk perulangan yang digunakan untuk melakukan perulangan program sesuai dengan jumlah dan tingkat perulangannya.

Perintahnya :

*For* <Variabel=Nilai\_awal> To <Nilai\_Akhir> < Selisih\_pertambahan>

<Pernyataan>

*Next*

## 7. **WHILE – WEND**

Perintah *WHILE – WEND* merupakan perintah untuk perulangan yang akan melakukan perulangan apabila keadaan yang diminta telah terpenuhi.

Perintahnya :

*While* <Keadaan>

<Perintah>

*Wend*

## 8. **GOSUB**

Perintah GOSUB merupakan perintah untuk lompatan yang akan melakukan lompatan ke label yang ditunjuk dan kembali ke tempat semula setelah melakukan perintah pembacaan program dengan menambahkan “Return”

Perintahnya :

```
Gosub <Nama_label>
      <Pernyataan>
<Nama_label>:
      <Pernyataan>

Return
```

## 9. **GOTO**

Perintah GOTO merupakan perintah untuk lompatan yang akan melakukan lompatan ke label yang ditunjuk tanpa kembali lagi ke tempat semula setelah melakukan perintah pembacaan program sehingga tidak menggunakan “Return”.

Perintahnya :

```
Goto<Nama_label>
      <Pernyataan>
<Nama_label>:
      <Pernyataan>
```

## 10. **EXIT**

Perintah *EXIT* merupakan perintah untuk mengakhiri perulangan *DO – LOOP, FOR – NEXT, WHILE – WEND*. 12

Perintahnya :

```
<pernyataan>
```

```
EXIT
```

(Afrie, 2010: 57-63)

## 2.5 Visual Basic 6.0

Bahasa Pemograman Visual Basic adalah suatu event-driven sistem pemograman visual dari perusahaan *Microsoft Visual Basic* (sering disingkat VB) merupakan sebuah bahasa pemograman yang menawarkan *Intergrated Development Environment (IDE)* visual untuk membuat program perangkat lunak berbasis sistem operasi *Microsoft Windows* dengan menggunakan model pemograman (COM). *Microsoft* untuk *Microsoft Windows* yang populer. Bila didefinisikan kata perkata Visual Basic antara lain kata “Visual” dalam nama bahasa pemograman ini mewakili pada metode untuk membuat GUI (*Graphical User Interface*). Dengan cara ini anda tidak perlu lagi menuliskan instruksi pemograman dalam kode-kode baris, tetapi secara mudah anda dapat melakukan drag dan drop objek-objek yang akan anda gunakan.

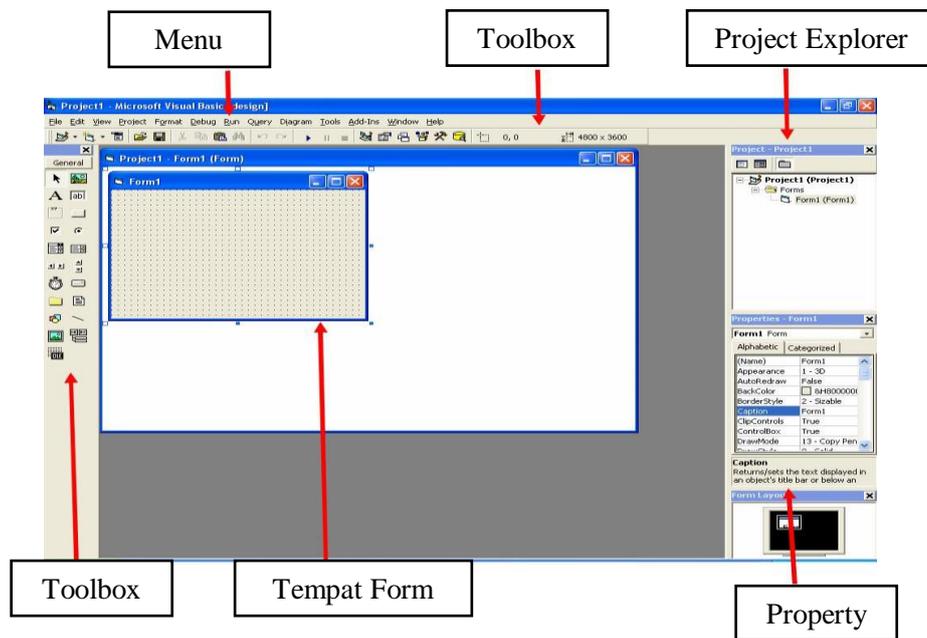
Visual Basic merupakan turunan bahasa pemograman BASIC dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat. Sedangkan kata “Basic” merupakan bagian bahasa BASIC (*Beginners All-Purpose Symbolic Instruction Code*), yaitu sebuah bahasa pemograman yang dalam sejarahnya sudah banyak digunakan oleh para programmer untuk menyusun aplikasi. Visual Basic dikembangkan dari bahasa Pemograman BASIC dan sekarang berisi banyak statement, fungsi, dan keyword yang beberapa diantaranya terhubung ke *Windows GUI*. (Basuki, 2006)

### 2.5.1 Form Layout Windows Visual Basic 6.0

*Form Layout Windows* adalah jendela yang menggambarkan posisi dari form yang ditampilkan pada layar monitor. Posisi form pada *form layout windows* inilah yang merupakan petunjuk dimana aplikasi akan ditampilkan pada layar monitor saat dijalankan.

- a. *Title bar*, digunakan untuk menampilkan judul proyek visual basic.
- b. *Toolbar*, digunakan untuk shortcut yang digunakan untuk membuat perintah suatu proyek.
- c. *Menubar*, menu perintah Visual Basic 6.0.
- d. *Toolbox*, objek yang digunakan untuk membuat suatu form.

- e. *Form*, bidang yang digunakan untuk membuat suatu form.
- f. *Project Explorer*, untuk menampilkan semua file ddalam aplikasi visual basic.
- g. *Property*, daftar setting properti.
- h. *Form Layout*, penampilan form pada layar.

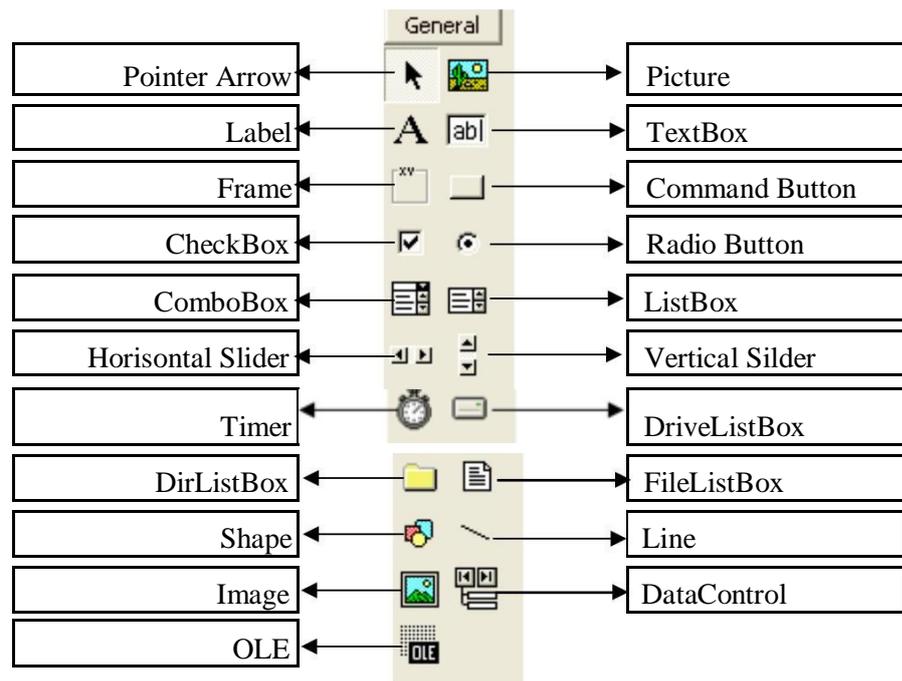


**Gambar 2.12** Tampilan Visual Basic 6.0

(Sumber: *Basuki, 2006:6*)

Pembuatan program aplikasi menggunakan Visual Basic dilakukan dengan membuat tampilan aplikasi pada form, kemudian diberi script program di dalam komponen-komponen yang diperlukan. *Form* disusun oleh komponen-komponen yang berada di *Toolbox*, dan setiap komponen yang dipakai harus diatur propertinya lewat jendela *Property*.

*Toolbox* berisi komponen-komponen yang bisa digunakan oleh suatu *project* aktif, artinya isi komponen dalam *toolbox* sangat tergantung pada jenis *project* yang dibangun. Komponen standar dalam *toolbox* dapat dilihat pada gambar 2.2 berikut in



**Gambar 2.13** Komponen standar dalam *Toolbox*

(Sumber: *Basuki, 2006:6*)

### 2.5.2 Konsep dasar Pemrograman Dalam Visual Basic 6.0

Konsep dasar Visual Basic 6.0 adalah pembuatan form dengan mengikuti aturan pemrograman Property, Metode dan Event. Hal ini berarti :

1. **Property**, setiap komponen didalam pemrograman Visual Basic dapat diatur propertinya sesuai dengan kebutuhan aplikasi. Property yang tidak boleh dilupakan pada setiap komponen adalah "**Name**", yang berarti nama variabel (komponen) yang akan digunakan dalam scripting. Properti "Name" ini hanya bisa diatur melalui jendela Property, sedangkan nilai properti yang lain bisa diatur melalui script seperti  

```
Command1.Caption="Play"
```

```
Text1.Text="Visual Basic"
```

```
Label1.Visible=False
```

```
Timer1.Enable=True
```
2. **Metode**, bahwa jalannya program dapat diatur sesuai aplikasi dengan menggunakan metode pemrograman yang diatur sebagai aksi dari setiap komponen. Metode inilah tempat untuk mengekspresikan logika pemrograman dari pembuatan suatu program aplikasi.

3. **Event**, setiap komponen dapat beraksi melalui event, seperti event click pada Command Button yang tertulis dalam layar script Command1\_Click, atau event Mouse Down pada picture yang tertulis dengan Picture1\_MouseDown. Pengaturan event dalam setiap komponen yang akan menjalankan semua metode yang dibuat. (Basuki, 2006: 2-4)

## 2.6 Downloader

*Downloader* adalah suatu rangkaian elektronik (*hardware*) yang berfungsi untuk mengirim, menulis, mendownload program dari komputer ke mikrokontroler. Biasanya ada 5 buah pin yang digunakan untuk memasukkan program kedalam mikrokontroler yaitu Mosi, Miso, Sck, Reset dan Ground. Terdapat tiga jenis downloder yaitu :

1. Pararel

*Downloader* jenis ini mungkin sudah mulai jarang digunakan karena *downloader* ini membutuhkan port paralel (LPT) sebagai jalur komunikasinya.

2. Serial

*Downloader* jenis ini sudah mulai jarang digunakan dan hanay digunakan oleh beberapa orang saja. Downloder ini membutuhkan port serial (com) sebagai jalur komunikasinya.

3. USB

Downloder jenis terbaru yang banyak dipakai oleh para penggemar Uc. Downloder ii dibuat menggunakan Uc yang diiisi dengan sebuah firmware. Kelebihan dar downloder ini yaitu proes menulisnya yang lebih cepat dan dapat dipakai di laptop.