

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Teori**

##### **2.1.1 Pengendali Lampu Jarak Jauh Menggunakan Internet**

Sistem pengendalian piranti dapat diartikan dengan sistem yang mampu melakukan proses kendali terhadap piranti yaitu mampu mengaktifkan dan menonaktifkan piranti tersebut serta dapat mengetahui kondisi sebenarnya dari piranti yaitu apakah setelah aktifkan, piranti benar-benar aktif atau bila dinonaktifkan, kondisi piranti benar-benar nonaktif. Jadi ada laporan ke pengendali tentang kondisi piranti setelah diadakan penge-set-an. Pengendalian terhadap piranti dapat dilakukan dari jarak dekat yaitu dengan menonaktifkan dan mengaktifkan piranti secara manual dengan tangan dan juga dapat dilakukan dari jarak jauh dengan menggunakan piranti kendali misalnya *remote* kontrol maupun dengan menggunakan komputer dengan perangkat lunaknya yang mampu melakukan pengendalian terhadap alat-alat elektronik dengan berbagai macam gerak dan sistem kerja yang kompleks.

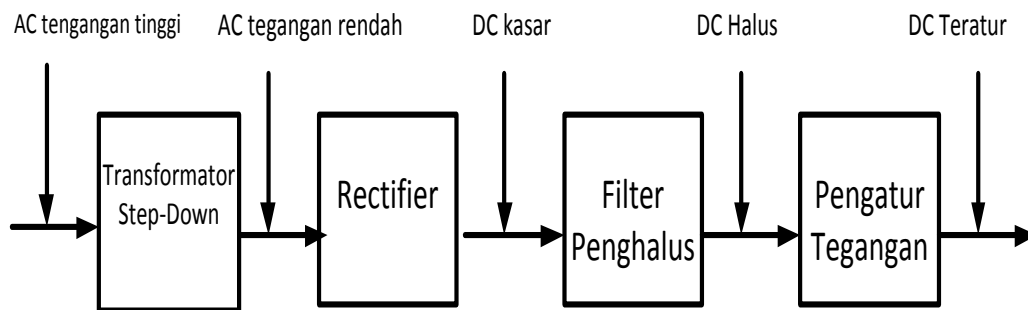
Kebutuhan akan sistem kendali jarak jauh semakin meningkat sejalan dengan era globalisasi. Perpindahan dan pergerakan manusia yang semakin luas dan cepat memerlukan sistem kendali yang mampu mengendalikan piranti dari mana saja tanpa terhambat oleh jarak. Internet merupakan jaringan yang luas yang dapat diakses oleh siapa saja dan dimana saja serta dapat dipakai untuk komunikasi data.

#### **2.2 Hardware yang digunakan**

##### **2.2.1 Power Supply**

Diagram blok dari sebuah power supply DC di perlihatkan pada gambar di bawah ini. Karena input sumbernya memiliki tegangan yang relatif tinggi, di gunakan sebuah transformator *step-down* dengan rasio lilitan yang sesuai untuk mengkonversi tegangan ini ke tegangan rendah. *Output* AC dari sisi sekunder transformator kemudian di searahkan dengan menggunakan dioda-dioda *rectifier*

silikon konvensional untuk menghasilkan *output* yang masih kasar. *Output* ini kemudian dihaluskan dan kemudian difilter sebelum disalurkan ke sebuah rangkaian yang akan mengatur atau menstabilkan tegangan *output* ini tetap berada dalam keadaan yang relatif konstan walaupun terdapat fluktuasi baik arus beban maupun pada tegangan *input* sumber. (Sumber : Mike Tooley, 2002 Rangkaian Elektronik Prinsip dan Aplikasi,107)



**Gambar 2.1** Blok Rangkaian *Power supply*

(Sumber : Mike Tooley, 2002 Rangkaian Elektronik Prinsip dan Aplikasi,107)

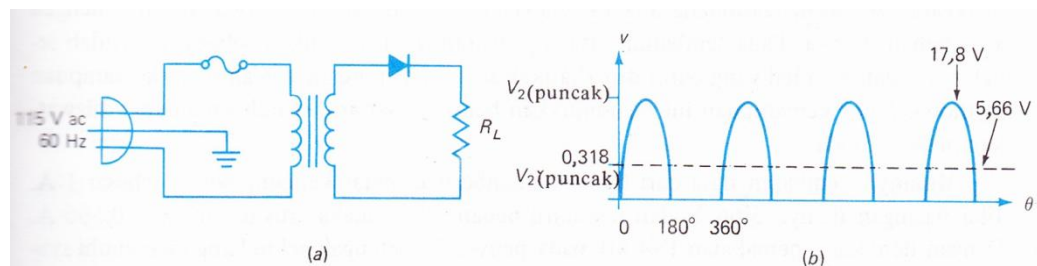
a. Transformator

Trafo atau transformator merupakan komponen utama dalam membuat rangkaian *power supply* yang berfungsi untuk mengubah tegangan listrik. trafo dapat menaikkan/menurunkan tegangan. Pada rangkaian *power supply* yang dibuat yaitu menggunakan trafo *step down* (penurun tegangan) apabila tegangan lilitan sekunder lebih rendah dari tegangan primer. (Sumber : Malvino Barmawi Prinsip-Prinsip Elektronika 47)

b. Dioda *Rectifier* (penyearah)

Rectifier dalam rangkaian *power supply* berfungsi untuk mengubah tegangan listrik AC yang berasal dari trafo *step down* menjadi tegangan listrik arus DC.

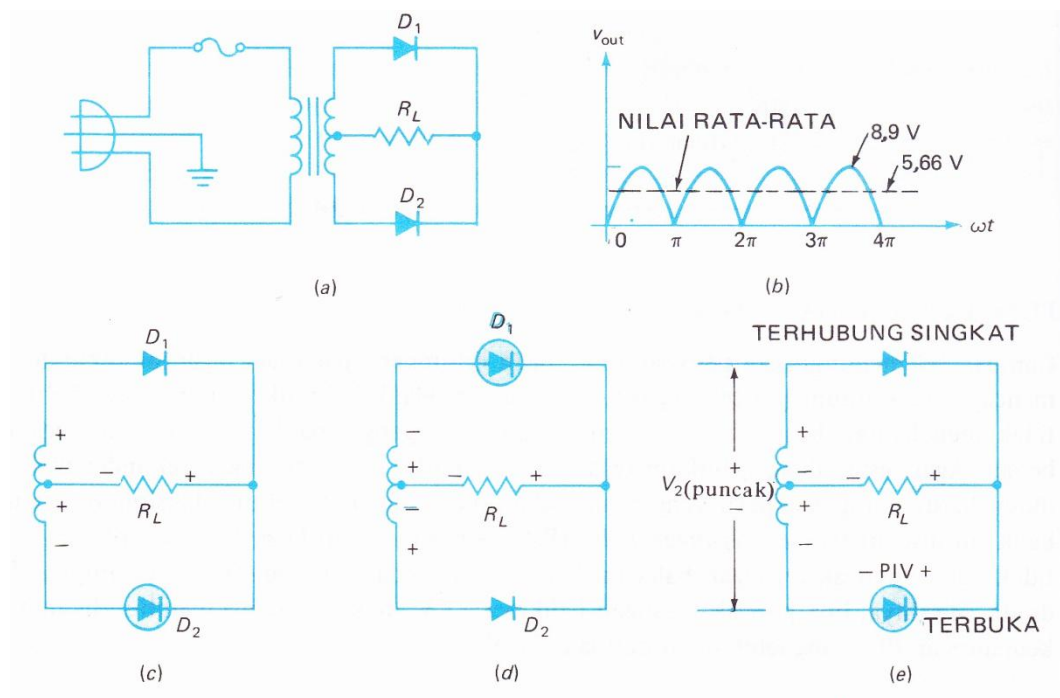
Penyearah setengah gelombang. Gambar 2.2 (a) memperlihatkan rangkaian penyearah setengah gelombang . pada setengah siklus tegangan sekunder yang positif, dioda mengalami prategangan maju untuk setiap tegangan-tegangan sesaat yang lebih besar daripada tegangan *offset* ( sekitar 0,7 V untuk dioda silicon dan 0,3 V untuk dioda germanium). Ini menghasilkan tegangan lintas tahanan beban yang mendekati bentuk setengah gelombang sinus. Dengan menggunakan pendekatan dioda ideal karena puncak tegangan sumber jauh lebih besar daripada tegangan *offset* dioda.puncak tegangan yang di searahkan sama dengan puncak tegangan sekunder, seperti ditunjukkan pada gambar 2.2 (b) . pada setengah siklus negatif, dioda mengalami prategangan balik. Dengan mengabaikan arus bocor ( yang sama dengan arus balik) arus beban menjadi nol. Inilah sebabnya mengapa tegangan beban jatuh menjadi nol diantara  $180^\circ$  dan  $360^\circ$ . (Sumber : Malvino Barmawi Prinsip-Prinsip Elektronika 49)



**Gambar 2.2** (a). Penyearah Setengah Gelombang Gambar (b). Gambar keluaran yang disearahkan  
(Sumber : Malvino Barmawi Prinsip-Prinsip Elektronika 49)

Penyearah gelombang penuh (Gambar 2.3 a) memperlihatkan penyearah gelombang penuh. Selama setengah siklus tegangan sekunder yang positif, dioda yang diatas mengalami prategangan maju dan dioda yang dibawah mengalami prategangan balik sehingga, arus mengalir melalui dioda yang diatas, tahanan beban, dan setengah lilitan yang diatas gambar (Gambar 2.3 b) . Selama Setengah siklus yang negatif, arus mengalir melalui dioda yang di bawah, tahanan beban dan setengah lilitan yang di bawah (gambar 2.3 c) perhatikan bahwa dalam (Gambar 2.3 d) dan di tegangan beban mempunyai polaritas yang sama. Hal ini disebabkan karena arus mengalir melalui tahanan beban dari arah yang sama tanpa

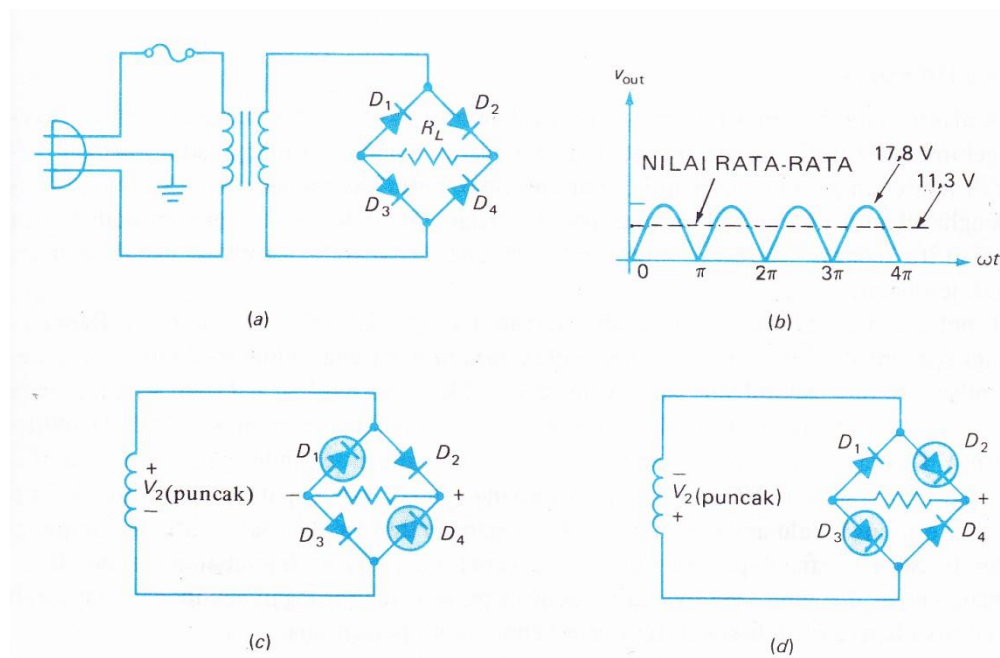
memperhatikan dioda mana yang menghantar. Jadi, tegangan beban berbentuk sinyal gelombang penuh yang disearahkan seperti (gambar 2.3 e) (Sumber : Malvino Barmawi Prinsip-Prinsip Elektronika 52)



**Gambar 2.3** (a). Penyearah Gelombang Penuh (b). Keluaran yang di Searahkan (c). Setengah Siklus Positif (d). Setengah Siklus Negatif (e). Pucak tegangan balik (Sumber : Malvino Barmawi Prinsip-Prinsip Elektronika 52)

Penyearah jembatan yang merupakan cara menyearahkan yang paling terkenal karena menonjolkan puncak tegangan yang sama dengan penyearah setengah gelombang dan mempunyai nilai rata-rata yang lebih tinggi daripada penyearah gelombang penuh (Gambar 2.4 a) menunjukkan penyearah jembatan. Selama setengah siklus tegangan sekunder yang positif ( Gambar 2.4 b ) dioda  $D_2$  dan  $D_3$  mengalami prategangan maju sehingga tegangan beban mempunyai polaritas seperti yang di tunjukkan tanda kurang pada sebelah kiri dan tanda kurang pada sebelah kanan. Selama setengah siklus negatif (Gambar 2.4 b ) dioda  $D_1$  dan  $D_4$  mengalami prategangan maju dan sekali lagi tegangan beban mempunyai polaritas kurang tambah seperti yang di tunjukkan. Jadi, untuk kedua

setengah siklus, tegangan beban mempunyai polaritas yang sama karena arus beban selalu mengalir dalam arah yang sama tanpa memperhatikan dioda mana yang menghantar. Itulah sebabnya mengapa tegangan beban adalah sinyal gelombang penuh yang disearahkan. (Sumber : Malvino Barmawi Prinsip-Prinsip Elektronika 55)



**Gambar 2.4** (a). Penyearah Gelombang Penuh (b). Keluaran yang disearahkan (c). Setengah Siklus Positif (d). Setengah Siklus Negatif  
 (Sumber : Malvino Barmawi Prinsip-Prinsip Elektronika 55)

Tegangan rata-rata dengan mengabaikan penurunan tegangan pada dioda dalam gambar puncak tegangan beban adalah

$$V_{out}(\text{puncak}) = V_2(\text{puncak})$$

Perhatikan bahwa semua tegangan sekunder yang muncul melintasi tahanan beban ini salah satu hal yang membuat penyearah jembatan lebih baik daripada penyearah gelombang penuh, karena hanya setengah tegangan sekunder saja yang mencapai keluaran . untuk membuat transformator yang disadap di tengah menghasilkan tegangan yang sama di setiap setengah lilitan sekundernya yang cukup susah dan mahal. Karena keluaran jembatan adalah sinyal gelombang penuh nilai rata-rata atau nilai DC-nya adalah

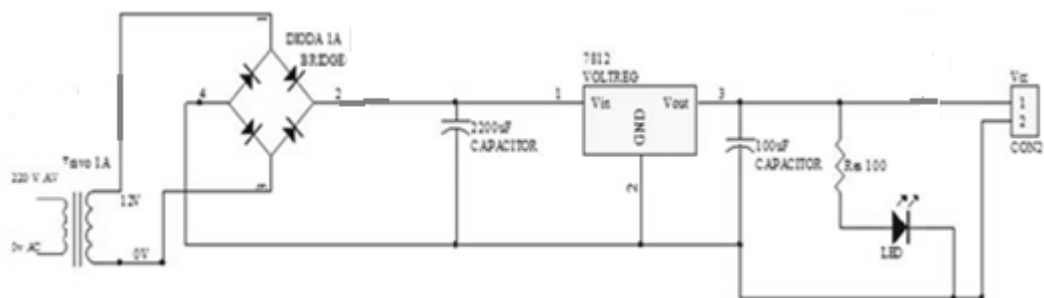
$$V_{dc} = 0,636V_{out} \text{ (puncak).}$$

c. *Filter* (penyaring)

Penyaring atau *filter* merupakan bagian yang terdiri dari kapasitor yang berfungsi sebagai penyaring atau meratakan tegangan listrik yang berasal dari *rectifier*.

d. *Regulator*

Pada rangkaian ini IC yang digunakan yaitu IC LM7812 yang merupakan IC regulator yang memiliki nilai keluaran 12 volt.



**Gambar 2.5** Rangkaian *Power Supply*

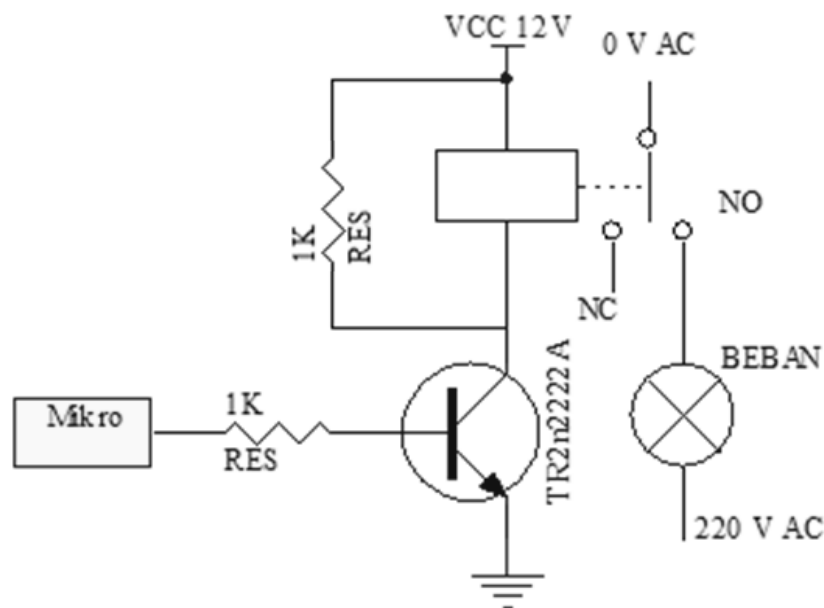
( Sumber : Malvino, Albert Paul, Prinsip-Prinsip Elektronika, Salemba Teknika 2004 Hal : 447 )

### 2.2.2 *Driver Relay*

*Driver relay* adalah rangkaian elektronika yang biasanya digunakan untuk mengendalikan serta pengoperasian sesuatu dari jarak jauh atau semacam *remote*. Tentunya rangkaian ini bisa mempermudah dan juga memperlancar pekerjaan yang memang kadang membutuhkan rangkaian dari *relay* ini. Dengan menggunakan *rangkaian relay* tersebut, anda bisa melakukan kontrol dan juga mengoperasikan perangkat elektronik yang anda miliki dari jarak jauh dan tentu saja anda tidak perlu bergeser serta berpindah tempat duduk.

Komponen inti dari *driver relay* adalah transistor. Cara yang termudah untuk menggunakan sebuah transistor adalah sebagai sebuah *switch* artinya bahwa kita mengoperasikan transistor pada salah satu dari saturasi atau titik sumbat,

tetapi tidak di tempat-tempat sepanjang garis beban. Jika sebuah transistor berada dalam keadaan saturasi, transistor tersebut seperti sebuah *switch* yang tertutup dari kolektor emiter. Jika transistor tersumbat (*cutoff*), transistor seperti sebuah *switch* yang terbuka.



**Gambar 2.6** Rangkaian *Driver Relay*  
( Sumber : PCcontrol.wordpress.com)

Gambar 2.6 merupakan gambar rangkaian driver relay dimana rangkaian tersebut merupakan rangkaian *switching* transistor yang digerakkan oleh tegangan *step* yang berasal dari *power supply*. Jika tegangan input (dari mikrokontroler) nol, transistor tersumbat (*cutoff*). Dalam hal ini, transistor kelihatannya seperti sebuah *switch* yang terbuka. Dengan tidak adanya arus yang melalui tahanan kolektor, maka tegangan output sama dengan +12 V.

Jika tegangan input sebesar +5V, maka arus basis adalah :

$$I_B = \frac{V_{BB} - V_{BE}}{R_B}$$

Bayangkan transistor terhubung – singkat antara kolektor dan emiter. Maka idealnya tegangan output jatuh menjadi nol dan arus saturasi adalah

$$I_{C(sat)} = \frac{V_{cc}}{R_c}$$

### 2.2.3 Sistem Minimum ATmega8535

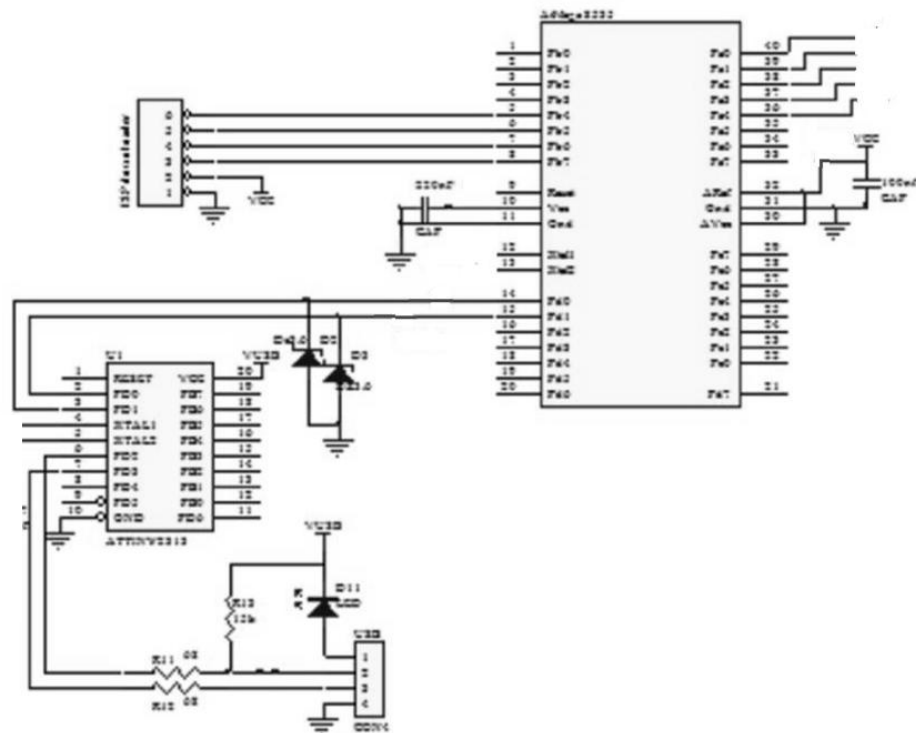
Sistem minimum (sismin) *mikrokontroler* adalah rangkaian elektronik minimum yang diperlukan untuk beroperasinya IC *mikrokontroler*. Sismin ini kemudian bisa dihubungkan dengan rangkaian lain untuk menjalankan fungsi tertentu. Di keluarga *mikrokontroler* AVR, seri 8535 adalah salah satu seri yang sangat banyak digunakan.

*Mikrokontroler* Atmega8535 telah dilengkapi dengan *osilator internal*, sehingga tidak diperlukan kristal atau resonator eksternal untuk sumber *clock* CPU. Namun osilator ini maksimal 8Mhz jadi saya sarankan untuk tetap memakai kristal eksternal. Osilator internal oleh pabriknya telah disetting 1 Mhz, dan untuk merubahnya kita perlu merubah setting pada Fuse bit. Namun untuk pengaturan fuse bit perlu berhati-hati, sebab pengaturan ini begitu rawan karena bila salah menyetingannya bisa menyebabkan *mikrokontroler* rusak.

Sistem minimum AVR sangat sederhana dimana hanya menghubungkan VCC dan AVCC ke +5V dan GND dan AGND ke ground serta pin *reset* tidak dihubungkan apa-apa (diambangkan) . *Chip* akan reset jika tegangan nol atau pin RESET dipaksa nol. Dan ini merupakan sistem minimum tanpa memakai kristal. Untuk yang memakai kristal rangkaian diatas ditambah kristal pada pin XTAL1 dan XTAL2. (<http://thathit.files.wordpress.com> diakses tanggal 13/04/2014)

Untuk membuat rangkaian sismin ATmega8535 diperlukan beberapa komponen yaitu:

1. *Mikrokontoler* ATmega8535
2. *Con Header* putih 8 pin, 4 pcs
3. *Con header* hitam 40 pin, 1 pcs
4. Kapasitor 22pF (2 pcs), 10nF (1 pcs), 100nF (1 pcs).
5. Kristal 11.059200 MHz, 1 pcs
6. *Push Button* 2 pin, 1 pcs
7. LED 2 pcs
8. Resistor 4k7 (1 pcs), 220Ohm (1 pcs).
9. *Socket* mikro 40 pin.



**Gambar 2.7** Rangkaian Sistem Minimum ATmega8535  
(Sumber : <http://thathit.files.wordpress.com>)

## 2.2.4 Mikrokontroler ATMEGA8535

*Mikrokontroler* merupakan sebuah sistem komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu *chip* IC, sehingga sering disebut *single chip micro computer*. Lebih lanjut, *mikrokontroler* merupakan sistem komputer yang mempunyai satu atau beberapa tugas yang sangat spesifik, berbeda dengan PC (*Personal Computer*) yang memiliki beragam fungsi. Perbedaan lainnya adalah perbandingan RAM dan ROM yang sangat berbeda antara komputer dengan *mikrokontroler*.

Mikrokontroler adalah sebuah *sistem microprocessor* dimana didalamnya sudah terdapat CPU, ROM, RAM, I/O, *Clock* dan peralatan *internal* lainnya yang sudah saling terhubung dan terorganisasi (teralamati) 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 menurut Winoto (2008:3).



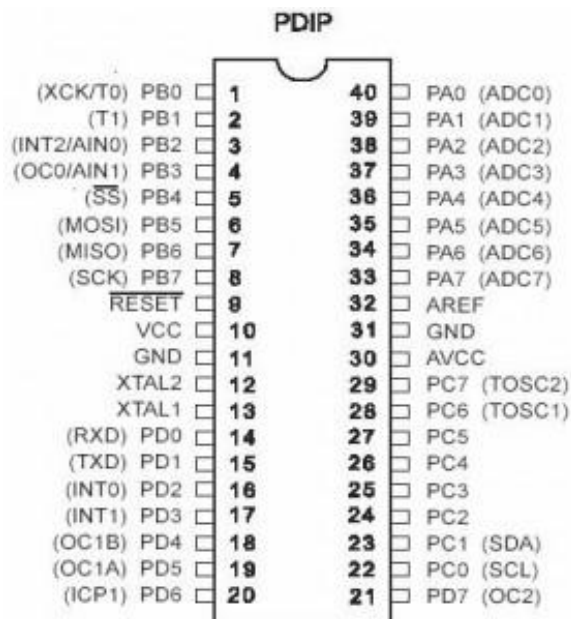
**Gambar 2.8** Mikrokontroler ATmega8535

( Sumber : <http://www.duniaelektronika.net/mikrokontroler-atmega8535-sistem-minimum> )

Teknologi yang digunakan pada *mikrokontroler* AVR berbeda dengan *mikrokontroler* seri MCS-51. AVR berteknologi RISC (*Reduced Instruction Set Computer*), sedangkan seri MCS-51 berteknologi CISC (*Complex Instruction Set Computer*). *Mikrokontroler* AVR dapat dikelompokkan menjadi empat kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega, dan keluarga AT89RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, kelengkapan peripheral dan fungsi-fungsi tambahan yang dimiliki. (<http://www.duniaelektronika.net/mikrokontroler-atmega8535-sistem-minimum> diakses tanggal 13/04/2014)

#### **2.2.4.1 Konfigurasi Pin Atmega8535**

*Mikrokontroler* AVR ATmega memiliki 40 *pin* dengan 32 *pin* diantaranya digunakan sebagai *port paralel*. Satu *port paralel* terdiri dari 8 *pin*, sehingga jumlah *port* pada *mikrokontroler* adalah 4 *port*, yaitu *port* A, *port* B, *port* C dan *port* D. Sebagai contoh adalah *port* A memiliki *pin* antara *port* A.0 sampai dengan *port* A.7, demikian selanjutnya untuk *port* B, *port* C, *port* D. Diagram *pin* mikrokontroler dapat dilihat pada gambar berikut:



**Gambar 2.9** Konfigurasi pin pada ATmega 8535

( Sumber : <http://digilib.polsri.ac.id/files/disk1/72/ssptpolsri-gdl-rizkylazua-3596-3-babii.pdf> )

Dibawah ini merupakan table tentang penjelasan pin pada *mikrokontroler* ATmega8535 :

**Tabel 2.1** Penjelasan pin pada mikrokontroler ATmega8535

|        |                                                                                                                                                                                                                                                                                                           |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Vcc    | Tegangan suplai (5 volt)                                                                                                                                                                                                                                                                                  |
| GND    | Ground                                                                                                                                                                                                                                                                                                    |
| RESET  | Input reset level rendah, pada <i>pin</i> ini selama lebih dari panjang pulsa <i>minimum</i> akan menghasilkan reset walaupun clock sedang berjalan. RST pada <i>pin</i> 9 merupakan reset dari AVR. Jika pada <i>pin</i> ini diberi masukan low selama minimal 2 machine cycle maka sistem akan di-reset |
| XTAL 1 | Input penguat osilator inverting dan input pada rangkaian operasi clock internal                                                                                                                                                                                                                          |
| XTAL 2 | Output dari penguat osilator inverting                                                                                                                                                                                                                                                                    |
| Avcc   | Pin tegangan suplai untuk port A dan ADC. Pin ini harus dihubungkan ke Vcc walaupun ADC tidak digunakan, maka <i>pin</i> ini                                                                                                                                                                              |

|      |                                                                                                                                          |
|------|------------------------------------------------------------------------------------------------------------------------------------------|
|      | harus dihubungkan ke Vcc melalui <i>low pass filter</i>                                                                                  |
| Aref | <i>pin</i> referensi tegangan <i>analog</i> untuk ADC                                                                                    |
| AGND | <i>pin</i> untuk <i>analog ground</i> . Hubungkan kaki ini ke GND, kecuali jika <i>board</i> memiliki <i>analog ground</i> yang terpisah |

( Sumber : <http://digilib.polsri.ac.id/files/disk1/72/ssptpolsri-gdl-rizkylazua-3596-3-babii.pdf> )

Pin33 sampai dengan *pin* 40 merupakan *pin* dari *port A*. Merupakan 8 *bit directional port I/O*. Setiap *pin*-nya dapat menyediakan *internal pull-up resistor* (dapat diatur per *bit*). *Output buffer port A* dapat memberi arus 20 mA dan dapat mengendalikan *display LED* secara langsung. *Data Direction Register port A* (DDRA) harus di-setting terlebih dahulu sebelum *port A* digunakan. *Bit-bit* DDRA diisi 0 jika ingin memfungsikan *pin-pin port A* yang disesuaikan sebagai *input*, atau diisi 1 jika sebagai *output*. Selain itu, *pin-pin* pada *port A* juga memiliki fungsi-fungsi alternatif khusus seperti yang dapat dilihat dalam tabel:

**Tabel 2.2** Penjelasan pin pada port A

| <b>Pin</b> | <b>Keterangan</b>          |
|------------|----------------------------|
| PA.7       | ADC7 (ADC Input Channel 7) |
| PA.6       | ADC6 (ADC Input Channel 6) |
| PA.5       | ADC7 (ADC Input Channel 5) |
| PA.5       | ADC4 (ADC Input Channel 4) |
| PA.3       | ADC3 (ADC Input Channel 3) |
| PA.2       | ADC2 (ADC Input Channel 2) |
| PA.1       | ADC1 (ADC Input Channel 1) |
| PA.0       | ADC0 (ADC Input Channel 0) |

( Sumber : <http://digilib.polsri.ac.id/files/disk1/72/ssptpolsri-gdl-rizkylazua-3596-3-babii.pdf> )

*Pin 1 sampai dengan pin 8 merupakan pin dari port B. Merupakan 8 bit directional port I/O. Setiap pin-nya dapat menyediakan internal pull-up resistor (dapat diatur per bit). Output buffer port B dapat memberi arus 20 mA dan dapat mengendalikan display LED secara langsung. Data Direction Register port B (DDRB) harus di-setting terlebih dahulu sebelum port B digunakan. Bit-bit DDRB diisi 0 jika ingin memfungsikan pin-pin port B yang disesuaikan sebagai input, atau diisi 1 jika sebagai output. Selain itu, pin-pin port B juga memiliki fungsi-fungsi alternatif khusus seperti yang dapat dilihat dalam tabel:*

**Tabel 2.3** Penjelasan pin pada port B

| <b>Pin</b> | <b>Keterangan</b>                                                                       |
|------------|-----------------------------------------------------------------------------------------|
| PB.7       | SCK (SPI Bus Serial Clock)                                                              |
| PB.6       | VISO (SPI Bus Master Input/Slave Output)                                                |
| PB.5       | VOSI (SPI Bus Master Output/Slave Input)                                                |
| PB.4       | SS (SPI Slave Select Input)                                                             |
| PB.3       | AIN1 (Analog Comparator Negative Input)OCC (Timer/Counter0 Output Compare Match Output) |
| PB.2       | AIN0 (Analog Comparator Positive Input)INT2 (External Interrupt2 Input)                 |
| PB.1       | T1 (Timer/Counter1 External Counter Input)                                              |
| PB.0       | T0 (Timer/Counter0 External Counter Input)XCK (JSART External Clock Input/Output)       |

( Sumber : <http://digilib.polsri.ac.id/files/disk1/72/ssptpolsri-gdl-rizkylazua-3596-3-babii.pdf>)

*Pin 22 sampai dengan pin 29 merupakan pin dari port C. Port C sendiri merupakan port input atau output. Setiap pin-nya dapat menyediakan internal pull-up resistor (dapat diatur per bit). Output buffer port C dapat memberi arus 20 mA dan dapat mengendalikan display LED secara langsung. Data Direction*

Register port C (DDRC) harus di-setting terlebih dahulu sebelum port C digunakan. Bit-bit DDRC diisi 0 jika ingin memfungsikan pin-pin port C yang disesuaikan sebagai input, atau diisi 1 jika sebagai output. Selain itu, pin-pin port D juga memiliki fungsi-fungsi alternatif khusus seperti yang dapat dilihat dalam tabel :

**Tabel 2.4** Penjelasan pin pada port C

| <b>Pin</b> | <b>Keterangan</b>                                |
|------------|--------------------------------------------------|
| PC.7       | TOSC2 (Timer Oscillator Pin 2)                   |
| PC.6       | TOSC1 (Timer Oscillator Pin 1)                   |
| PC.1       | SDA (Two-Wire Serial Bus Data Input/Output Line) |
| PC.0       | SCL (Two-Wire Serial Bus Clock Line)             |

( Sumber : <http://digilib.polsri.ac.id/files/disk1/72/ssptpolsri-gdl-rizkylazua-3596-3-babii.pdf> )

Pin 14 sampai dengan pin 20 merupakan pin dari port D. Merupakan 8 bit directional port I/O. Setiap pin-nya dapat menyediakan internal pull-up resistor (dapat diatur per bit). Output buffer port D dapat memberi arus 20 mA dan dapat mengendalikan display LED secara langsung. Data Direction Register port D (DDRD) harus di-setting terlebih dahulu sebelum port D digunakan. Bit-bit DDRD diisi 0 jika ingin memfungsikan pin-pin port D yang disesuaikan sebagai input, atau diisi 1 jika sebagai output. Selain itu, pin-pin port D juga memiliki fungsi-fungsi alternatif khusus seperti yang dapat dilihat dalam tabel:

**Tabel 2.5** Penjelasan pin pada port D

| <b>Pin</b> | <b>Keterangan</b>                 |
|------------|-----------------------------------|
| PD.0       | RDX (UART input line)             |
| PD.1       | TDX (UART output line)            |
| PD.2       | INT0 (external interrupt 0 input) |
| PD.3       | INT1 (external interrupt 1 input) |

|      |                                                             |
|------|-------------------------------------------------------------|
| PD.4 | OC1B ( <i>Timer/Counter1 output compareB match output</i> ) |
| PD.5 | OC1A ( <i>Timer/Counter1 output compareA match output</i> ) |
| PD.6 | ICP ( <i>Timer/Counter1 input capture pin</i> )             |
| PD.7 | OC2 ( <i>Timer/Counter2 output compare match output</i> )   |

( Sumber : <http://digilib.polsri.ac.id/files/disk1/72/ssptpolsri-gdl-rizkylazua-3596-3-babii.pdf> )

#### 2.2.4.2 Fitur Mikrokontroler ATmega8535

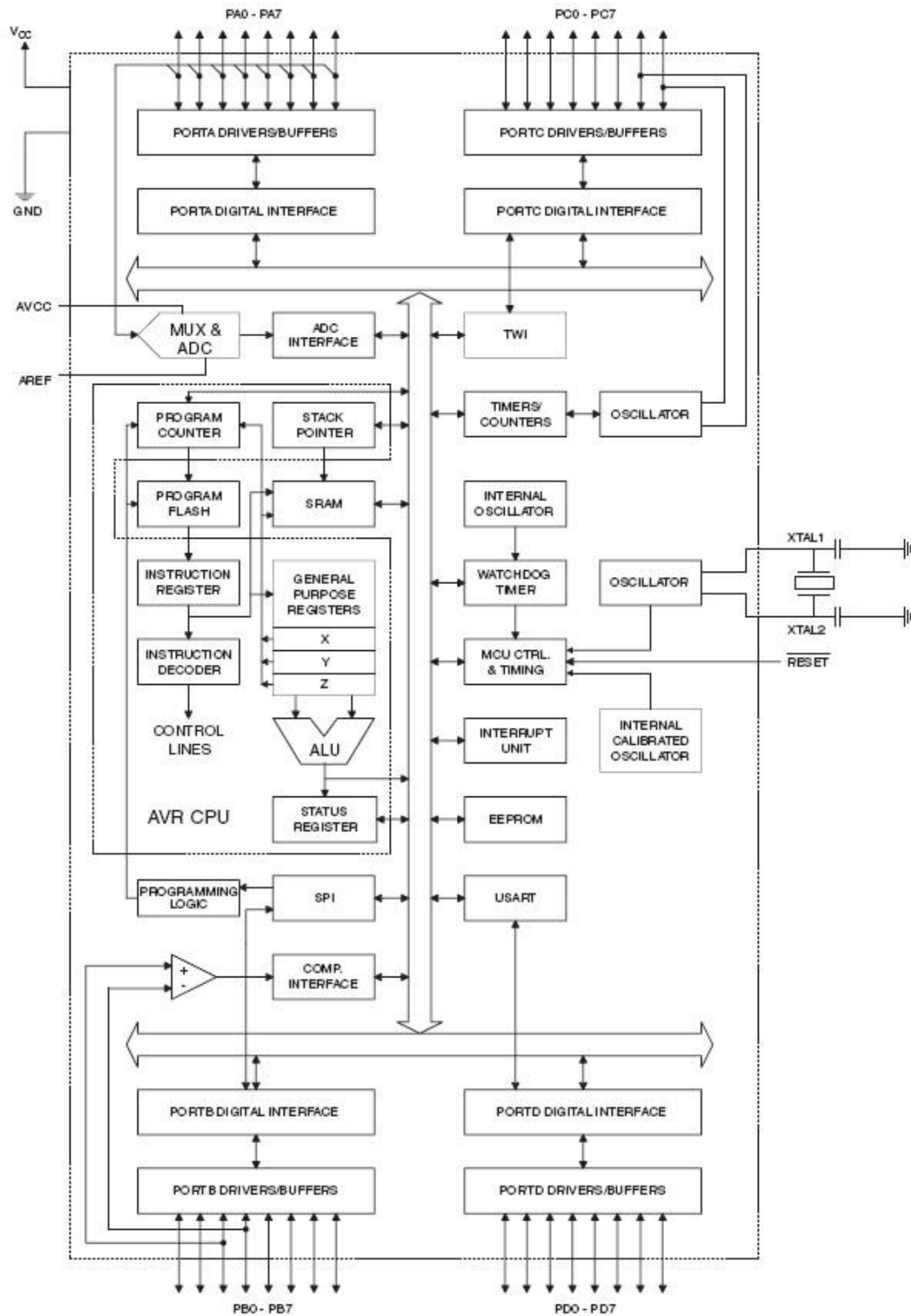
Adapun kapabilitas detail dari ATmega8535 adalah sebagai berikut,

1. Sistem mikroprosesor 8 bit berbasis *RISC* dengan kecepatan maksimal 16 MHz.
2. Kapabilitas memori *flash* 8 KB, *SRAM* sebesar 512 byte, dan *EEPROM* (*Electrically Erasable Programmable Read Only Memori*) sebesar 512 byte.
3. *ADC* internal dengan fidelitas 10 bit sebanyak 8 *channel*.
4. Portal komunikasi serial (*USART*) dengan kecepatan maksimal 2,5 Mbps.
5. Enam pilihan mode *sleep* untuk menghemat penggunaan daya listrik.

#### 2.2.4.3 Arsitektur ATmega8535

Dibawah ini adalah bagian – bagian dari ATmega8535 :

1. Saluran I/O sebanyak 32 buah, yaitu *Port A*, *Port B*, *Port C* dan *Port D*.
2. *ADC* 8 *channel* 10 bit.
3. Tiga buah *Timer/Counter* dengan kemampuan pembanding.
4. CPU yang terdiri atas 32 buah *register*.
5. *Watchdog timer* dengan osilator *internal*.
6. *SRAM* sebesar 512 byte.
7. Memori *Flash* sebesar 8 KB dengan kemampuan *Read While Write*.
8. *Interrupt internal* dan *eksternal*
9. *Port* antarmuka *SPI* (*Serial Peripheral Interface*).
10. *EEPROM* sebesar 512 byte yang dapat diprogram saat operasi.
11. Antarmuka komparator analog.
12. *Port USART* untuk komunikasi serial



**Gambar 2.10** Blok diagram fungsional ATmega8535  
 ( Sumber : <http://digilib.polsri.ac.id/files/disk1/72/ssptpolsri-gdl-rizkylazua-3596-3-babii.pdf> )

### 2.2.5 USB Serial

USB (*Universal Serial Bus*) adalah sebuah spesifikasi untuk komunikasi antara perangkat dan sebuah *host controller* . yang dikembangkan dan diciptakan oleh Ajay Bhatt petugas di Intel. USB pertama kali dikembangkan pada tahun 1994 oleh kelompok tujuh perusahaan: Compaq, DEC, IBM, Intel, Microsoft, NEC dan Nortel Dalam konsepsi aslinya, koneksi standar, dimaksudkan untuk memudahkan koneksi antar perangkat eksternal untuk komputer pribadi PC, menggantikan banyak konektor di bagian belakang komputer menyederhanakan konfigurasi perangkat lunak dari semua perangkat yang terhubung ke USB, serta memungkinkan *bandwidth* yang lebih besar untuk perangkat eksternal. Sebelum standar ini, untuk mengkonfigurasi perangkat terhubung ke komputer, maka perlu untuk mengatur jumper atau IRQS dengan standar ini maka diperkenalkan "standar *Plug and Play* " memungkinkan pengguna yang kurang berpengalaman untuk menghubungkan perangkat apapun tanpa memerlukan konfigurasi tambahan pada tingkat perangkat keras.



**Gambar 2.11** Konstruksi USB  
( Sumber : <http://danangbcp.blogspot.com>)

Panjang maksimal kabel USB standar (USB 2.0) adalah 5,0 meter. Alasan utama untuk batas ini adalah *delay* maksimum yang diperbolehkan untuk 1500 ns. Jika perintah USB host tidak dijawab oleh perangkat USB dalam jangka waktu yang ditentukan, sistem menganggap perintah hilang. Penundaan dapat diterima maksimum untuk kabel adalah 26 Spesifikasi USB 2.0 mengharuskan keterlambatan kabel kurang dari 5,2 ns per meter (192.000Km / s), yang dekat bandwidth maksimum untuk kabel tembaga standar). Hal ini memungkinkan penggunaan kabel meter 5. USB 3.0 standar tidak secara langsung menentukan panjang kabel maksimum, hanya membutuhkan bahwa semua kabel listrik memiliki spesifikasi yang memungkinkan. Untuk kabel kawat tembaga, beberapa

perhitungan menunjukkan panjang maksimal 3m. Dalam kabel serat optik, kemungkinan akan memiliki panjang lebih besar dan pembangunan yang lebih kompleks dan konstruksi. (<http://elektronika-dasar.web.id/artikel-elektronika/usb-universal-serial-bus/> diakses tanggal 12/10/2013 pukul 12:03:00 PM )

**Tabel 2.6** Kabel dan *Chips* (konektor) USB

| USB Kabel 1.x/2.0 |        |             |           |
|-------------------|--------|-------------|-----------|
| PIN               | Nama   | Warna Kabel | Deskripsi |
| 1                 | Vcc    | Merah       | 5 V       |
| 2                 | D-     | Putih       | Data -    |
| 3                 | D+     | Hijau       | Data +    |
| 4                 | Ground | Hitam       | Ground    |

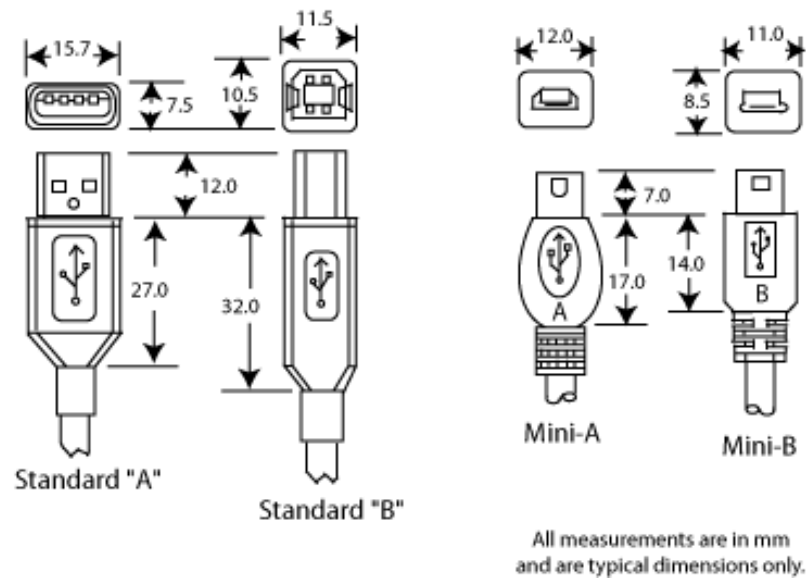
(Sumber : <http://danangbcp.blogspot.com>)

#### 2.2.5.1 Sifat dan kemampuan USB

USB (*Universal Serial Bus*) mempunyai sifat dan kemampuan sebagai berikut :

1. Mudah dihubungkan ke komputer karena tidak perlu membukapenutup komputer.
2. Bentuk konektor USB seragam, hanya ada dua bentuk tipe socket Hot-plugable artinya peralatan USB dapat dihubungkan saatkomputer hidup Plug and Play artinya peralatan USB mudah digunakan, saat dihubungkan ke komputer, komputer akan mengenali mengambilprogram (*driver*) yang sesuai dengan peralatan tersebut.
3. Perlatan yang memerlukan sumber daya kecil, bisa mengambil sumber daya dari *port* USB. Sumber daya peralatan dimatikan saattidak digunakan. Kecepatan transfer data tinggi, dapat mencapai 12 Mbps.
4. Satu *port* USB dapat dihubungkan dengan 127 alat.

Konstruksi Port USB (*Universal Serial Bus*) Konektor USB hanya ada 2 macam, yakni konektor tipe A dan konektor tipe B seperti terlihat dalam gambar dibawah.



**Gambar 2.12** Konstruksi USB Port A dan Port B  
( Sumber : <http://rangkaiaielektronik4.blogspot.com>)

Konektor tipe A dipakai untuk menghubungkan kabel USB ke terminal USB yang ada pada bagian belakang komputer produksi berapa tahun terakhir ini. Konektor tipe B dipakai untuk menghubungkan kabel USB ke terminal USB yang ada pada peralatan, untuk peralatan USB yang sederhana, misalnya mouse, biasanya tidak pakai konektor B, untuk menghemat biaya kabel langsung dihubungkan ke bagian dalam mouse. versi *port* USB dalam acuan baku ditentukan dengan persyaratan yang sangat ketat untuk kabel USB, tidak sembarang kabel bisa dipakai, lebih-lebih untuk USB dengan kecepatan transfer data penuh sampai 1.2 Mbps.

## 2.2.6 Laptop

Laptop adalah komputer bergerak yang berukuran relatif kecil dan ringan, beratnya berkisar dari 1-6 kg, tergantung ukuran, bahan, dan spesifikasi laptop tersebut. (<http://www.elektrabandungsurvey.com/id/laptop/pengertiannotebook.php> diakses tanggal 11/03/2014)



**Gambar 2.13** Laptop  
(sumber : <http://komputer.blogekstra.com>)

Laptop/Notebook memiliki fungsi yang sama dengan komputer desktop pada umumnya. Komponen yang terdapat di dalamnya sama persis dengan komponen pada desktop, hanya saja ukurannya diperkecil, dijadikan lebih ringan, lebih tidak panas, dan lebih hemat daya.

Kebanyakan Laptop/Notebook menggunakan LCD (*Liquid Crystal Display*) dengan ukuran 10 inci sampai 17 inci tergantung dari ukuran laptop itu sendiri. Selain itu, keyboard di laptop juga dilengkapi dengan *touchpad* yang berfungsi sebagai "pengganti" dari mouse. *Keyboard* dan *mouse* tambahan dapat terhubung melalui USB dan PS 2 / jika tersedia.

#### **2.2.6.1 Spesifikasi Laptop**

Pada pembuatan sistem pengendali jarak jauh dapat menggunakan Laptop/Netbook dengan spesifikasi standar yaitu :

1. Processor Intel® Atom™ @1.66 GHz 1.67GHz
2. Installed Memory (RAM) 1,00 GB
3. Sistem tipe 32-bit Operating Sistem
4. Windows 7

#### **2.2.7 Modem**

Modem adalah singkatan dari modulator dan demodulator. Modulator mempunyai fungsi melakukan proses menghantarkan data dalam bentuk sinyal informasi ke sinyal pembawa agar dapat dikirim ke pengguna melalui media

tertentu. Dan proses ini biasa dinamakan dengan proses modulasi. Dalam proses ini data dari komputer yang berbentuk sinyal digital akan diubah menjadi sinyal analog. Demodulator mempunyai fungsi sebagai proses untuk mendapatkan kembali data yang dikirim oleh pengirim. Pada proses ini data akan dipisahkan dari frekuensi tinggi dan data yang berupa sinyal analog akan diubah kembali menjadi sinyal digital agar bisa dibaca oleh komputer.

#### **2.2.7.1 Fungsi modem**

Modem mempunyai beberapa fungsi yaitu, sebagai berikut:

1. Merubah komunikasi dua arah dari sinyal digital menjadi sinyal analog atau sebaliknya. Sebuah sinyal digital yang dikirimkan dari komputer diubah menjadi sinyal analog terlebih dahulu oleh modem dan ditransmisi melalui kabel telepon untuk diantarkan ke komputer lainnya dan sebaliknya.
2. *voice modem* sebagai saluran radio, percakapan telepon maupun audio lainnya.
3. Untuk meminimalisir tingkat kesalahan yang terjadi ketika melakukan transmisi sinyal, dan modem juga akan memeriksa informasi yang diterima apakah rusak atau tidak dengan metode checksum jika terjadi kerusakan maka data tersebut akan dikirim kembali ke tempat asal.

#### **2.2.7.2 Jenis jenis modem**

Banyak sekali jenis jenis modem yang ada pada saat ini, jenis modem dapat dibedakan berdasarkan pemasangannya dan jaringannya yaitu, sebagai berikut:

1. *Modem Internal*

*Modem Internal* merupakan perangkat jaringan yang terdapat pada papan ekspansi yang dipasang pada slot motherboard. keuntungan modem ini adalah cara pemasangannya mudah dan harganya relatif lebih murah. Modem internal ini juga dikenal sebagai On-Board Modem



**Gambar 2.14** Modem *Internal*  
( Sumber : <http://id.wikipedia.org> )

## 2. Modem *eksternal*

Modem *eksternal* adalah modem yang dipasangkan di luar komputer, biasanya ditancapkan pada slot USB dan Kabel LAN. Kelebihan dari modem ini adalah dapat dengan mudah dibawa atau dipindahkan ke tempat lain.



**Gambar 2.15** Modem *Eksternal*  
( Sumber : <http://id.wikipedia.org/wiki/Modem> )

## 3. Modem Kabel

Modem kabel yaitu perangkat keras yang menggunakan media kabel sebagai media perantaranya , contoh : TV kabel dan jaringan telepon. Jaringan TV kabel ini dapat digunakan untuk koneksi internet hingga mencapai kecepatan maksimum 27 Mbps untuk download, dan kecepatan upload hingga 2,5 Mbps.



**Gambar 2.16** Modem Kabel  
( Sumber : <http://id.wikipedia.org/wiki/Modem> )

## 4. Modem tanpa kabel

Modem ini menggunakan media tanpa kabel untuk perantaranya, contoh : modem GSM, Modem CDMA. Kekurangan dari modem jenis ini adalah

kecepatan dapat melemah jika cuaca di sekitarnya tidak mendukung. Kelebihannya adalah mudah dibawa dan mudah dipindahkan sesuai keinginan.



**Gambar 2.17** Modem Tanpa Kabel  
( Sumber : [blog.unikom.ac.id](http://blog.unikom.ac.id) )

## **2.3 Software Yang digunakan**

### **2.3.1 BASCOM AVR**

Bahasa pemrograman basic terkenal didunia sebagai bahasa pemrograman yang handal. Sangat bertolak belakang dari namanya basic, bahasa ini sebenarnya bahasa yang memiliki kemampuan tingkat tinggi. Bahkan banyak para programmer terkenal dunia memakai bahasa pemrograman ini sebagai senjata ampuhnya. Bahasa pemrograman basic banyak digunakan untuk aplikasi *mikrokontroler* karena kompatibel oleh mikrokontroler jenis AVR dan didukung dengan *compiler* pemrograman berupa *software* BASCOM AVR. Bahasa basic memiliki penulisan program yang mudah dimengerti walaupun untuk orang awam sekalipun, karena itu bahasa ini dinamakan bahasa basic. Jenis perintah programnya seperti do, loop, if, then, dan sebagainya.

BASCOM AVR sendiri adalah salah satu tool untuk pengembangan / pembuatan program untuk kemudian ditanamkan dan dijalankan pada mikrokontroler terutama mikrokontroler keluarga AVR . BASCOM AVR juga bisa disebut sebagai IDE (Integrated Development Environment) yaitu lingkungan kerja yang terintegrasi, karena disamping tugas utamanya meng-compile kode program menjadi file hex atau bahasa mesin, BASCOM AVR juga memiliki kemampuan atau fitur lain yang berguna seperti monitoring komunikasi serial dan untuk menanamkan program yang sudah di compile ke mikrokontroler.

BASCOM AVR menyediakan pilihan yang dapat mensimulasikan program. Program simulasi ini bertujuan untuk menguji suatu aplikasi yang dibuat dengan pergerakan LED yang ada pada layar simulasi dan dapat juga langsung dilihat pada LCD, jika kita membuat aplikasi yang

berhubungan dengan LCD. Intruksi yang dapat digunakan pada editor BASCOM AVR relatif cukup banyak dan tergantung dari tipe dan jenis AVR yang digunakan. Berikut ini adalah beberapa perintah intruksi-intruksi dasar yang digunakan pada BASCOM AVR.

(<http://dheni-yulistianto.blogspot.com/2013/07/pengertian-bascom-avr.html>  
diakses tanggal 12/10/2013 pukul 12:03:00 PM)

**Tabel 2.7** intruksi dasar BASCOM AVR

| No | Intruksi   | Keterangan              |
|----|------------|-------------------------|
| 1  | DO...LOOP  | Perulangan              |
| 2  | GOSUB      | Memanggil prosedur      |
| 3  | IF...THEN  | Percabangan             |
| 4  | FOR...NEXT | Perulangan              |
| 5  | WAIT       | Waktu tanda detik       |
| 6  | WAITMS     | Waktu tanda milik detik |
| 7  | WAITUS     | Waktu tanda micro detik |
| 8  | GOTO       | Loncat ke alamat memori |

( Sumber : <http://dheni-yulistianto.blogspot.com/2013/07/pengertian-bascom-avr.html>)

### 2.3.1.1 Kontruksi bahasa BASIC pada BASCOM AVR

Setiap bahasa pemrograman mempunyai standar penulisan program. Konstruksi dari program bahasa BASIC harus mengikuti aturan sebagai berikut:

\$regfile = "header"

'inisialisasi

'deklarasi variabel

'deklarasi konstanta

Do

'pernyataan-pernyataan

Loop

end

Pengarah preprosesor \$regfile = "m16def.dat" merupakan pengarah preprosesor bahasa BASIC yang memerintahkan untuk meyisipkan file lain, dalam hal ini adalah file m 8535 def.dat yang berisi deklarasi *register* dari *mikrokonroller* ATmega 8535, pengarah pre-prosesor lainnya yang sering digunakan ialah sebagai berikut:

\$crystal = 8000000 'menggunakan crystal clock 8 Hz

\$baud = 9600 'komunikasi serial dengan baudrate 9600

\$eeprom 'menggunakan fasilitas eeprom

Tipe data merupakan bagian program yang paling penting karena sangat berpengaruh pada program. Pemilihan tipe data yang tepat maka operasi data menjadi lebih efisien dan efektif.

**Tabel 2.8** Tipe Data pada BASCOM AVR

| No | Tipe    | Jangkauan                                       |
|----|---------|-------------------------------------------------|
| 12 | Bitbyte | 0 atau 10-255                                   |
| 3  | Integer | -32,768-32,767                                  |
| 4  | Word    | 0-65535                                         |
| 5  | Long    | -2147483648-2147483647                          |
| 6  | Single  | $1.5 \times 10^{-45}$ - $3,4 \times 10^{38}$    |
| 7  | Double  | $5.0 \times 10^{-324}$ to $1.7 \times 10^{308}$ |
| 8  | String  | >254 byte                                       |

( Sumber : <http://dheni-yulistianto.blogspot.com/2013/07/pengertian-bascom-avr.html>)

Konstanta merupakan suatu nilai dengan tipe data tertentu yang tidak dapat diubah-ubah selama proses program berlangsung. Konstanta harus didefinisikan terlebih dahulu diawal program.

Contoh : Kp = 35, Ki=15, Kd=40

Variabel adalah suatu pengenalan yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program yang dapat diubah-ubah sesuai dengan kebutuhan. Nama dari variabel sesuai dengan yang diinginkan namun hal yang

terpenting adalah setiap variabel diharuskan Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf, max 32 karakter. Tidak boleh mengandung spasi atau simbol-simbol khusus seperti : \$, ?, %, #, !, &, \*, (, ), -, +, = dan lain sebagainya kecuali underscore.

Deklarasi sangat diperlukan bila akan menggunakan pengenalan (identifier) dalam suatu program. Bentuk umum pendeklarasian Terdiri dari:

1. Deklarasi variable yaitu Dim nama\_variabel AS tipe\_data  
Contoh : Dim x As Integer ' deklarasi x bertipe integer.
2. Deklarasi Konstanta Dalam Bahasa Basic konstanta di deklarasikan langsung.  
Contohnya : S = "Hello world" ' Assign string
3. Deklarasi buatan Fungsi yang perlu dideklarasikan terlebih dahulu adalah fungsi yang dibuat oleh programmer. Bentuk umum deklarasi sebuah fungsi adalah : Sub Test ( byval variabel As tipe)  
Contohnya : Sub Pwm(byval Kiri As Integer , Byval Kanan As Integer)

### 2.3.1.2 Operator Perintah

Operator Perintah (Assignment operator) dalam Bahasa Basic berupa "=". Yang terdiri dari beberapa bagian yaitu :

1. Operator Aritmatika
  - \* : untuk perkalian
  - / : untuk pembagian
  - + : untuk pertambahan
  - : untuk pengurangan
  - % : untuk sisa pembagian (modulus)
2. Operator Hubungan (Perbandingan) digunakan untuk membandingkan hubungan dua buah operand atau sebuah nilai / variable, misalnya :
  - = 'Equality X = Y
  - < 'Less than X < Y
  - > Greater than X > Y
  - <= 'Less than or equal to X <= Y
  - >= 'Greater than or equal to X >= Y

3. Operator logika digunakan untuk membandingkan logika hasil dari operator-operator hubungan. Operator logika ada empat macam, yaitu :  
NOT 'Logical complement  
AND 'Conjunction  
OR 'Disjunction  
XOR 'Exclusive or
4. Operator bitwise digunakan untuk memanipulasi bit dari data yang ada di memori. Operator bitwise dalam Bahasa Basic :  
Shift A, Left, 2 : Pergeseran bit ke kiri  
Shift A, Right, 2 : Pergeseran bit ke kanan  
Rotate A, Left, 2 : Putar bit ke kiri  
Rotate A, right, 2 : Putar bit ke kanan

#### **2.3.1.3 Pernyataan Kondisional (IF-THEN – END IF)**

Pernyataan ini digunakan untuk melakukan pengambilan keputusan terhadap dua buah bahkan lebih. kemungkinan untuk melakukan suatu blok pernyataan atau tidak. Konstruksi penulisan pernyataan IF-THEN-ELSE-END IF pada bahasa BASIC ialah sebagai berikut:

IF pernyataan kondisi 1 THEN

'blok pernyataan 1 yang dikerjakan bila kondisi 1 terpenuhi

IF pernyataan kondisi 2 THEN

'blok pernyataan 2 yang dikerjakan bila kondisi 2 terpenuhi

IF pernyataan kondisi 3 THEN

'blok pernyataan 3 yang dikerjakan bila kondisi 3 terpenuhi

Setiap penggunaan pernyataan IF-THEN harus diakhiri dengan perintah END IF sebagai akhir dari pernyataan kondisional.

#### **2.3.1.4 Pernyataan Kondisional (SELECT-CASE-END SELECT)**

Pernyataan ini digunakan untuk melakukan pengambilan keputusan terhadap banyak kondisi. Konstruksi penulisan pernyataan SELECT-CASE-END SELECT pada bahasa BASIC ialah sebagai berikut:

SELECT CASE var

CASE 'kondisi1' : 'blok perintah1

CASE 'kondisi2' : 'blok perintah2

CASE 'kondisi3' : 'blok perintah3

CASE 'kondisi4' : 'blok perintah4

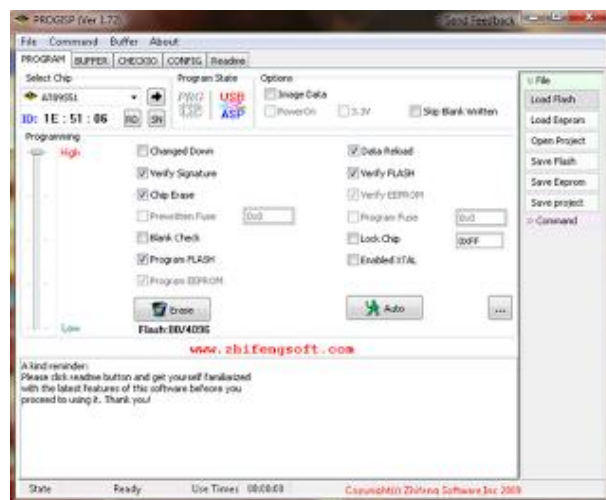
CASE 'kondisi5' : 'blok perintah5

CASE 'kondisi'n' : 'blok perintah'n'

END SELECT 'akhir dari pernyataan SELECT CASE

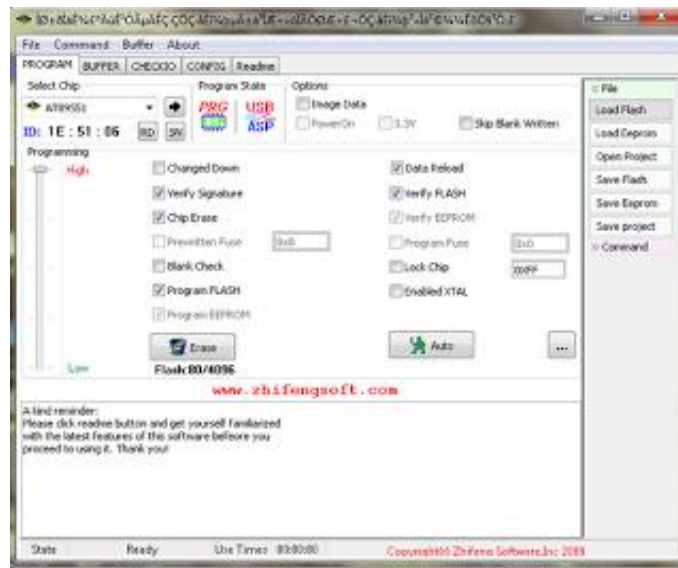
### 2.3.2 ProgISP

Progisp merupakan *Software* yang digunakan dalam pemrograman *mikrokontroler* khususnya saat melakukan Download File \*.HEX ke dalam memori *mikrokontroller* MCS. Untuk bisa menggunakan *software* ini tentunya harus mempunyai soft-copy *software* tersebut, karena *software* ini bersifat portable jadi tidak perlu di instal terlebih dahulu. Bentuk tampilan programnya sebagai berikut:



**Gambar 2.18** Tampilan program ProgISP  
( Sumber : Rytelyne Technologies, 3013)

Selain mempunyai *soft-copy* programnya kita juga harus punya hardware USBASP yang merupakan perangkat komunikasi antara PC dengan *Chip Mikrokontroler*. Setelah *hardware* terhubung tampilan jendela program menjadi seperti di bawah ini, perhatikan perbedaannya:

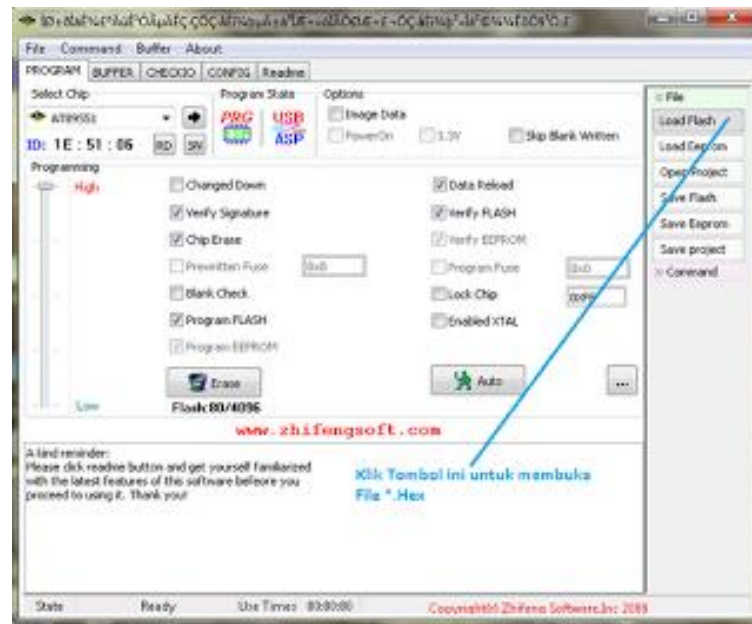


**Gambar 2.19** Perhatikan perubahan yang terjadi  
( Sumber : RyteLyne Technologies, 3013)

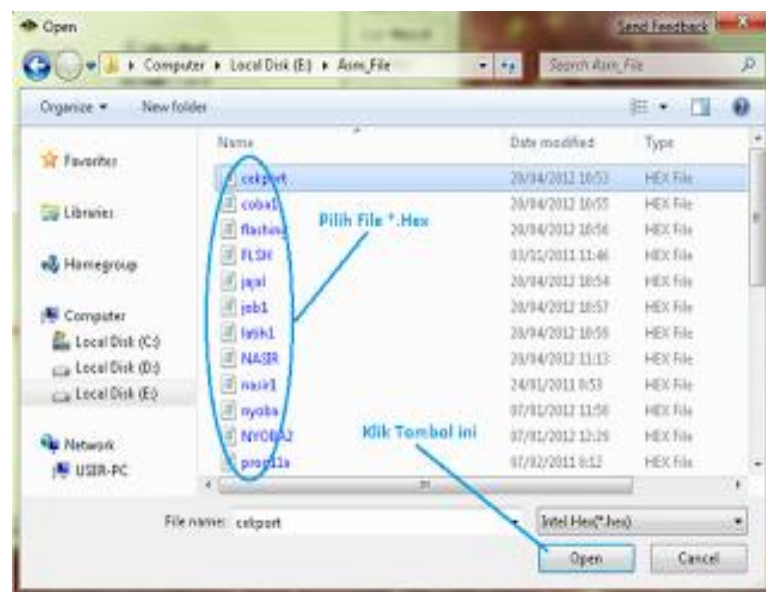


**Gambar 2.20** Hardware terdeteksi  
( Sumber : RyteLyne Technologies, 3013)

Selanjutnya memanggil file \*.HEX yang akan di download ke memori *mikrokontroller*, caranya Klik tombol *Load Flash* kemudian pilih file \*.HEX yang akan di *download* dan Klik tombol *Open*, seperti ditunjukan pada gambar berikut ini :



**Gambar 2.21** Klik Load Flash  
( Sumber : RyteLyne Technologies, 3013)



**Gambar 2.22** Memilih File \*.HEX  
( Sumber : RyteLyne Technologies, 3013)

Kemudian Klik tombol *Auto* untuk *Download*ing, pastikan proses downloadnya hingga selesai.



**Gambar 2.23** Download proses.  
( Sumber : Rytelyne Technologies, 3013)

Terakhir perhatikan apakah program dapat berjalan seperti yang diharapkan (pada rangkaian aplikasi *mikrokontroler*).

### 2.3.3 Visual Basic

Visual basic adalah salah satu *development tools* untuk membangun aplikasi dalam lingkungan *Windows*. Dalam pengembangan aplikasi, Visual Basic menggunakan pendekatan Visual untuk merancang user interface dalam bentuk *form*, sedangkan untuk kodingnya menggunakan dialek bahasa Basic yang cenderung mudah dipelajari. Visual Basic telah menjadi *tools* yang terkenal bagi para pemula maupun para developer dalam pengembangan aplikasi skala kecil sampai ke skala besar. Dalam lingkungan *Window's User-interface* sangat memegang peranan penting, karena dalam pemakaian aplikasi yang kita buat, pemakai senantiasa berinteraksi dengan User interface tanpa menyadari bahwa dibelakangnya berjalan instruksi-instruksi program yang mendukung tampilan dan proses yang dilakukan.

Pada pemrograman Visual, pengembangan aplikasi dimulai dengan pembentukan *user interface*, kemudian *mengatur properti dari objek-objek* yang

digunakan dalam *user interface*, dan baru dilakukan penulisan kode program untuk menangani kejadian-kejadian.

Bahasa Basic pada dasarnya adalah bahasa yang mudah dimengerti sehingga pemrograman di dalam bahasa Basic dapat dengan mudah dilakukan meskipun oleh orang yang baru belajar membuat program. Hal ini lebih mudah lagi setelah hadirnya Microsoft Visual Basic, yang dibangun dari ide untuk membuat bahasa yang sederhana dan mudah dalam pembuatan scriptnya (*simple scripting language*) untuk graphic user interface yang dikembangkan dalam sistem operasi Microsoft Windows.

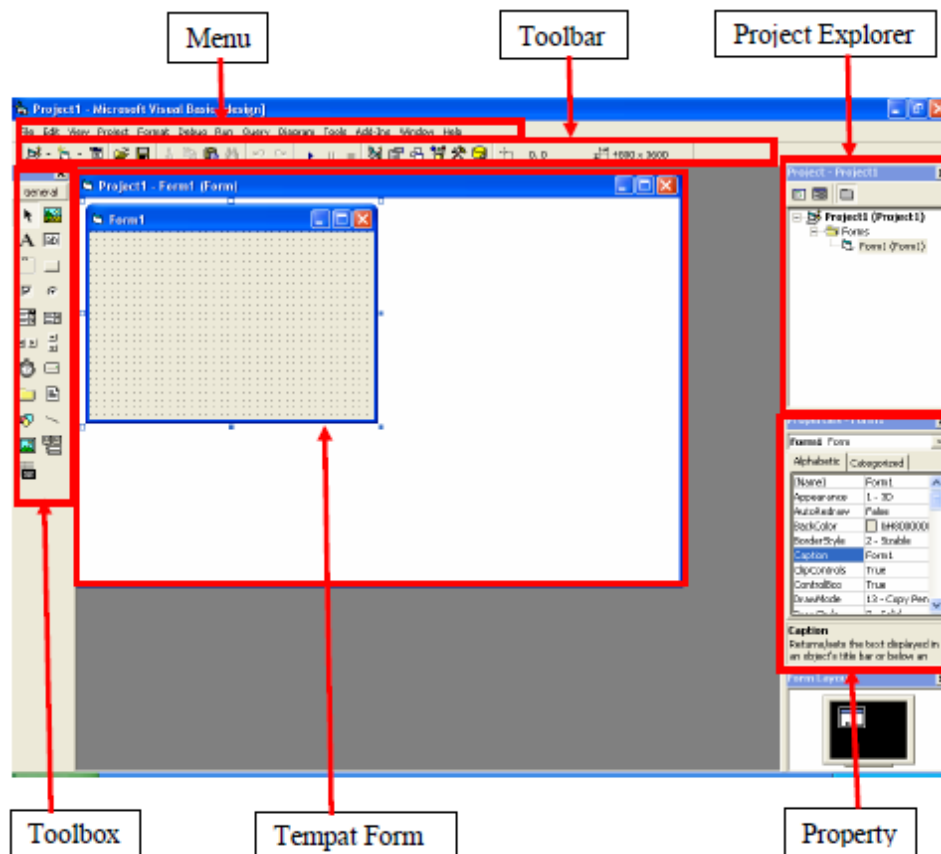
Visual Basic merupakan bahasa pemrograman yang sangat mudah dipelajari, dengan teknik pemrograman visual yang memungkinkan penggunanya untuk berkreasi lebih baik dalam menghasilkan suatu program aplikasi. Ini terlihat dari dasar pembuatan dalam visual basic adalah FORM, dimana pengguna dapat mengatur tampilan form kemudian dijalankan dalam script yang sangat mudah.

Ledakan pemakaian Visual Basic ditandai dengan kemampuan Visual Basic untuk dapat berinteraksi dengan aplikasi lain di dalam sistem operasi Windows dengan komponen *ActiveX Control*. Dengan komponen ini memungkinkan pengguna untuk memanggil dan menggunakan semua model data yang ada di dalam sistem operasi windows. Hal ini juga ditunjang dengan teknik pemrograman di dalam Visual Basic yang mengadopsi dua macam jenis pemrograman yaitu Pemrograman Visual dan *Object Oriented Programming* (OOP).

Visual Basic 6.0 sebetulnya perkembangan dari versi sebelumnya dengan beberapa penambahan komponen yang sedang tren saat ini, seperti kemampuan pemrograman internet dengan DHTML (*Dynamic HyperText Mark Language*), dan beberapa penambahan fitur database dan multimedia yang semakin baik. Sampai saat buku ini ditulis bisa dikatakan bahwa Visual Basic 6.0 masih merupakan pilih pertama di dalam membuat program aplikasi yang ada di pasar perangkat lunak nasional. Hal ini disebabkan oleh kemudahan dalam melakukan proses *development* dari aplikasi yang dibuat. (Sumber : Algoritma Pemrograman 2 Menggunakan Visual Basic 6.0, Achmad basuki, 2006, 1)

### 2.3.3.1 Interface Antar Muka Visual Basic 6.0

Interface antar muka Visual Basic 6.0, berisi menu, toolbar, toolbox, form, project explorer dan property seperti terlihat pada gambar 1.1. berikut:



**Gambar 2.24.** Interface antar muka Visual Basic 6.0

(Sumber : Algoritma Pemrograman 2 Menggunakan Visual Basic 6.0, Achmad basuki, 2006, 2)

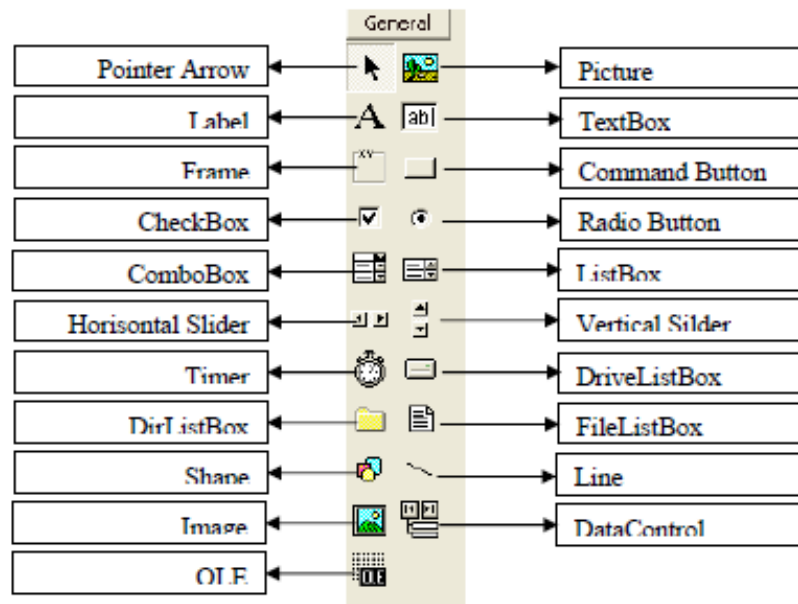
Pembuatan program aplikasi menggunakan Visual Basic dilakukan dengan membuat tampilan aplikasi pada form, kemudian diberi script program di dalam komponen-komponen yang diperlukan. Form disusun oleh komponen-komponen yang berada di [Toolbox], dan setiap komponen yang dipakai harus diatur propertinya lewat jendela [Property]. (Sumber : Algoritma Pemrograman 2 Menggunakan Visual Basic 6.0, Achmad basuki, 2006, 2)

Menu pada dasarnya adalah operasional standar di dalam sistem operasi windows, seperti membuat form baru, membuat *project* baru, membuka *project*

dan menyimpan project. Di samping itu terdapat fasilitas-fasilitas pemakaian visual basic pada menu. Untuk lebih jelasnya Visual Basic menyediakan bantuan yang sangat lengkap dan detail dalam MSDN.

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 1.2 berikut ini.

(Sumber : Algoritma Pemrograman 2 Menggunakan Visual Basic 6.0, Achmad basuki, 2006, 3)



**Gambar 2.25.** Komponen standar dalam Toolbox

(Sumber : Algoritma Pemrograman 2 Menggunakan Visual Basic 6.0, Achmad basuki, 2006, 3)

### 2.3.3.2 Konsep Dasar Pemrograman Dalam Visual Basic 6.0

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

1. **Property**: Setiap komponen di dalam 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 peroperti 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 prgram 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.

(Sumber : Algoritma Pemrograman 2 Menggunakan Visual Basic 6.0, Achmad basuki, 2006, 3)

### 2.3.3.3 Variable, Operasi, Ekspresi

#### a. Variabel

Variabel adalah suatu tempat untuk menampung suatu nilai pada memory komputer. Untuk lebih mudah diakses, variabel diberi nama. Nama inilah yang akan menjadi pengenalan suatu variabel, misalkan **a=4**, ini berarti bahwa nama variabelnya adalah a dan nilainya adalah 4.

Variabel di dalam visual basic mempunyai beberapa macam berdasarkan nilai yang ada di dalam variabel tersebut, antara lain:

1. Boolean : menampung nilai biner, True atau False
2. Byte : Menampung nilai bulat kecil antara 0-256
3. Integer : Menampung nilai bulat antara -32768 s/d 32768 (15 bit)

4. Long : Menampung nilai bulat dengan bit yang panjang (31 bit)
5. Single : Menampung nilai pecahan dari  $10^{-38}$  sampai dengan  $10^{38}$  pada bagian positif, dan  $-10^{-38}$  sampai dengan  $-10^{38}$  pada bagian negatif.
6. Double : Menampung nilai pecahan dari  $10^{-108}$  sampai dengan  $10^{108}$  pada bagian positif, dan  $-10^{-108}$  sampai dengan  $-10^{108}$  pada bagian negatif.
7. String : Menampung nilai non numerik atau *string*, misalkan untuk menyimpan alamat. Variabel ini tidak bisa dioperasikan secara aritmatika.
8. Date : Menampung nilai tanggal
9. Variant : Merupakan variabel bebas, yang menampung nilai tergantung nilai apa yang ditampung pertama kali. Variabel ini yang merupakan kelebihan dari visual basic (bahkam bisa menjadi kekurangan untuk pemrograman yang lebih tinggi) karena bersifat seperti bunglon.

(Sumber : Algoritma Pemrograman 2 Menggunakan Visual Basic 6.0, Achmad basuki, 2006, 20-23)

Kelebihan dari visual basic adalah bahwa setiap obyek baik berupa *form*, *report* (laporan), data atau komponen dapat menjadi variabel.

Variabel di dalam visual basic dibedakan menjadi 3 macam variabel yaitu variabel lokal, variabel *global* terhadap *form* dan variabel *global* terhadap aplikasi (*project*).

1. Variabel lokal: adalah variabel yang hanya aktif dalam suatu fungsi atau subroutine di dalam visual basic. Caranya dituliskan secara langsu di dalam suatu fungsi satu subroutine tanpa pendefinisian atau dengan definisi variabel di dalam fungsi.
2. Variabel global dalam *form*: adalah variabel yang aktif selama satu *form* berjalan, bila pindah ke *form* yag lain maka variabel ini tidak aktif. Ini dapat dilakukan dengan mendefinisikan variabel di luar *subroutine*, biasanya diletakkan pada baris paling atas dari suatu *form*. Contohnya **Dim a as integer** diletakkan pada baris paling atas, maka variabel a selalu aktif selama form aktif.

3. Variabel *global* dalam aplikasi: adalah variabel yang aktif selama aplikasi masih aktif. Variabel ini masih aktif meskipun *form* yang berjalan sudah berganti. Untuk mendefinisikan variabel ini dilakukan dengan menambahkan modul dan pendefinisian dengan ***global***.

(Sumber : Algoritma Pemrograman 2 Menggunakan Visual Basic 6.0, Achmad basuki, 2006, 20-23)

#### b. Operator

Operatort adalah suatu simbol atau tanda untuk menyatakan suatu operasi atau proses. Pada dasarnya komputer dengan ALUnya (Aritmethic Logical Unit), mempunyai dua macam operator yaitu operator Aritmatika dan operator logika (perbandingan). Operator-operator di dalam Visual Basic antara lain:

**Tabel 2.9** Operasi Visual Basic

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

(Sumber : Algoritma Pemrograman 2 Menggunakan Visual Basic 6.0, Achmad basuki, 2006, 20-23)

Penjelasan lebih lanjut mengenai operator ada di bagian ekspresi, karena operator ini merupakan bagian dari ekspresi. Pemakaian operator ini sangat berhubungan dengan penulisan rumus di dalam pemrograman.

(Sumber : Algoritma Pemrograman 2 Menggunakan Visual Basic 6.0, Achmad basuki, 2006, 1-2)

### c. Ekspresi

Ekspresi adalah suatu cara penulisan untuk memberikan atau memasukkan nilai kedalam variabel. Ekspresi secara umum dalam *computer statement* dituliskan sebagai:

**Variabel  $\leftarrow$  Nilai**

Di dalam Visual Basic ekspresi menggunakan tanda sama dengan (=). Dengan aturan sebelah kiri adalah variabel penampung (hasil) dan sebelah kanan adalah nilai yang dimasukkan ke variabel

**Variabel = Nilai**

Sebagai contoh untuk memasukkan nilai 5 ke dalam variabel a dapat dilakukan dengan **a=5**. Atau memasukkan nama 'widya' ke dalam variabel nama dapat dituliskan dengan **nama="widya"** (Pada tipe data string, penulisan diberi tanda petik dua). Atau untuk memasukkan nilai keputusan benar dapat dituliskan dengan **keputusan=True**, Pada tipe data boolean nilainya hanya *True* dan *False*. Sehingga dapat dikatakan bahwa cara memasukkan nilai ini sangat tergantung dari jenis nilai apakah itu boolean, numerik, *string* atau *date*. Sedangkan format bilangan dan tanggal secara *default* menggunakan format yang ada pada sistem operasi, meskipun tidak menutup kemungkinan diubah menggunakan format tersendiri.

(Sumber : Algoritma Pemrograman 2 Menggunakan Visual Basic 6.0, Achmad basuki, 2006, 20-23)

### 2.3.4 Team Viewer

Team Viewer adalah sebuah aplikasi untuk melakukan kontrol penuh pada komputer/laptop yang mendukung akses jarak jauh menggunakan internet.

Team Viewer mempunyai keunggulan yaitu sebagai berikut :

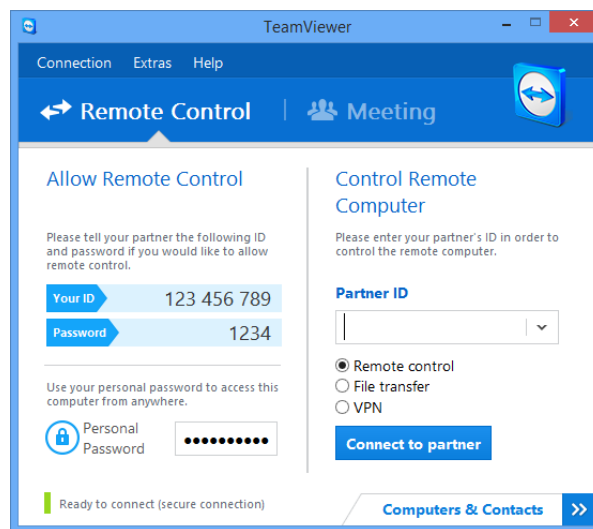
1. Bisa menggunakan internet untuk me-remote.

2. Kinerja aplikasi yang cepat asalkan koneksi internet stabil.
3. Sangat simple dan tidak ribet.
4. Punya fitur yang banyak.
5. Bisa membuat ruang rapat sampai 25 orang

(<http://malingpelajaran.blogspot.com/2013/12/pengertian-cara-menggunakan-team-viewer.html> : diakses tanggal 12/10/2013 pukul 11:55:00 PM)

#### 2.3.4.1 Remote Control

Cara Menggunakan Teamviewer 9 kita harus mengenal yang satu ini. Jika kita membuka teamviewer 9, pasti tampilannya akan seperti dibawah ini. Akan ada 2 buah tab yaitu, *Remote control* dan *meeting* pastikan komputer atau laptop sedang dalam keadaan hidup.

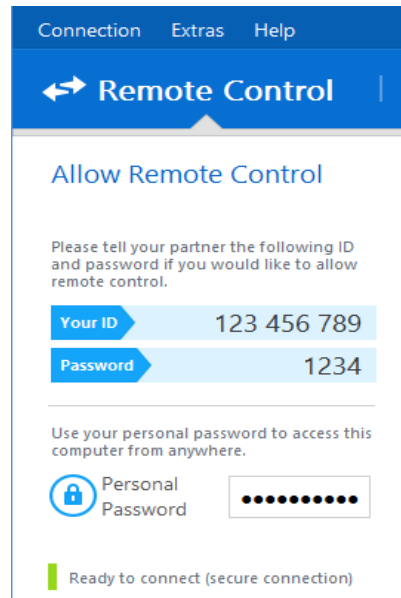


**Gambar 2.26** Tampilan Team Viewer  
( Sumber : Kuhnbergstrabe,Team Viewer 9 Manual)

#### 2.3.4.2 Allow Remote Control

*Allow Remote Control* adalah untuk memberikan akses kepada orang lain untuk meremote komputer/laptop kita. disini terdapat 2 buah *field* yang berisi ID dan juga *password* untuk mengendalikan komputer kita. Untuk ID tidak akan pernah bisa berubah, tapi untuk password kita bisa mengambil *password* baru dengan menekan tombol *refresh* disebalah kotaknya atau jika kita ingin

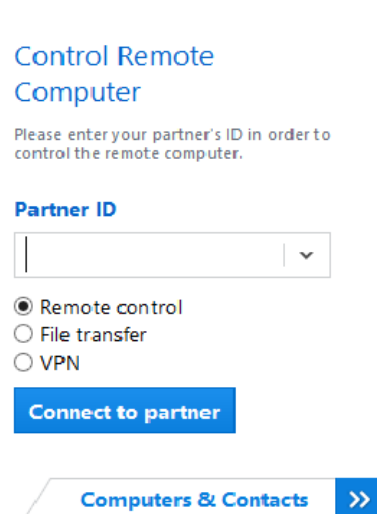
menggunakan *password* pribadi pun juga bisa dengan mengisi kotak personal di paling bawah.



**Gambar 2.27** Tampilan *Allow Control*  
( Sumber : RYTELYNE Technologies, 3013)

#### 2.3.4.3 Control Remote Control

*Control Remote Control* berfungsi dimana kita bisa mengendalikan komputer orang lain, asalkan kita bisa mendapatkan *Partner ID* atau ID dari team viewer pada komputer yang ingin kita kendalikan.



**Gambar 2.28** Control Remote Control  
( Sumber : RYTELYNE Technologies, 3013)