

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Wireless**

*Wireless Communication* atau dalam bahasa Indonesia di biasa dikenal dengan istilah Telekomunikasi Nirkabel adalah transfer informasi antara dua atau lebih titik yang tidak terhubung secara fisik. Jarak bisa pendek, seperti beberapa meter untuk remote control televisi, atau sejauh ribuan atau bahkan jutaan kilometer untuk ruang-dalam komunikasi radio. Ini meliputi berbagai jenis tetap, mobile, dan portabel radio dua arah, telepon seluler, *personal digital assistant* (PDA), dan jaringan nirkabel (Suprianto, 2006).

Pengertian wireless sendiri adalah teknologi tanpa kabel, dalam hal ini adalah melakukan telekomunikasi dengan menggunakan gelombang elektromagnetik sebagai media perantara pengganti kabel. Sekarang ini teknologi wireless berkembang sangat pesat sekali, secara kasat mata dapat kita lihat dengan semakin banyaknya penggunaan telepon selular, disamping itu berkembang juga teknologi wireless yang digunakan untuk akses internet. Contoh lain dari teknologi nirkabel termasuk GPS unit, pembuka pintu garasi atau pintu garasi, wireless mouse komputer, keyboard dan headset (audio), headphone, penerima radio, televisi satelit, siaran televisi tanpa kabel dan telepon.

#### **2.2 Efek Gitar**

Efek suara awalnya sering di pakai studio rekaman. Pada pertengahan tahun 1940-an, insinyur rekaman dan musisi eksperimental seperti Les Paul mulai memanipulasi rekaman pita *reel-to-reel* untuk menciptakan efek echo dan tidak biasa, suara futuristik. ("*miking*") teknik yang digunakan penempatan Mikrofon dalam ruang dengan sifat akustik yang dirancang khusus untuk mensimulasikan echo kamar.

Amplifier built-in adalah efek pertama yang akan digunakan secara teratur di luar studio oleh pemain gitar. Dari akhir 1940-an dan seterusnya, Gibson Guitar Corp mulai memasukan sirkuit vibrato di amplifier combo. Tahun 1950 Ray Butts amp Echo Sonic menjadi yang pertama untuk fitur suara echo "*slapback*", yang

dengan cepat menjadi populer dengan gitaris seperti Chet Atkins, Carl Perkins, Scotty Moore, Luther Perkins, dan Roy Orbison. Pada tahun 1950-an, tremolo, vibrato dan reverb banyak tersedia sebagai built-in efek pada gitar amplifier. Good Premier dan Gibson built ampli tube bertenaga dengan reverb Fender dimulai pada musim semi. manufaktur tremolo ampli Tremolux pada tahun 1955 dan Vibrolux pada tahun 1956. Distorsi tidak berpengaruh awalnya saat ditujukan oleh produsen amplifier, tetapi kadang bisa dengan mudah dipakai dengan "overdriving" pasokan listrik di amplifier tabung awal. Gitaris Johnny Burnette dan Willie Johnson adalah di antara yang pertama dengan sengaja meningkatkan keuntungan diluar batas sesuai dengan tujuannya untuk mencapai suara distorsi "Warm". Dave Davies dari The Kinks menjadi dokter untuk Penutur amp nya dengan memotong efek itu dengan pisau cukur untuk menghasilkan grittier suara gitar pada lagu 1964 " *You Really Got Me* ". Pada tahun 1965, Marshall Amplifikasi mulai menjual Marshall 1959, amplifier gitar mampu menghasilkan nada Warm dan terdistorsi "*crunch*" yang Ng-Rock. Efek Stand-alone tahun 1950-an dan 60-an awal seperti unit GA-VI vibrato Gibson dan Box reverb Fender, yang mahal dan tidak praktis, membutuhkan transformer besar dan tegangan tinggi. Unit-unit ahirnya berdiri sendiri selain pada permintaan konsumen sebagai efek yang hadir menyatu pada amplifier. Yang pertama populer yang berdiri sendiri adalah Watkins 1958 Copicat, tape relatif portabel efek gema dipopulerkan oleh band Inggris, The Shadows.

Elektronik transistor akhirnya berhasil memungkinkan untuk menjejalkan kreativitas aural dari studio rekaman menjadi lebih sederhana, unit stompbox sangat portabel. Transistor menggantikan tabung vakum, yang memungkinkan untuk format kompak yang lebih banyak dan stabilitas yang lebih besar. Dalam sejarah Efek gitar pertama *transistorized* adalah Maestro 1962 Fuzz Tone pedal, yang menjadi sensasi setelah digunakan dalam Rolling Stones 1965 hit "*(I Can't Get No) Satisfaction*". Warwick Electronics memproduksi wah wah pedal pertama, *The Clyde McCoy*, pada tahun 1967 dan tahun yang bersama Roger Mayer mengeluarkan efek oktaf pertama, Octavia, Pada tahun 1968, Univox mulai pemasaran Uni-Vibe pedal, Sebuah efek yang dirancang oleh insinyur audio Fumio Mieda yang menirukan pergeseran fasa aneh dan efek chorus dari Leslie rotating

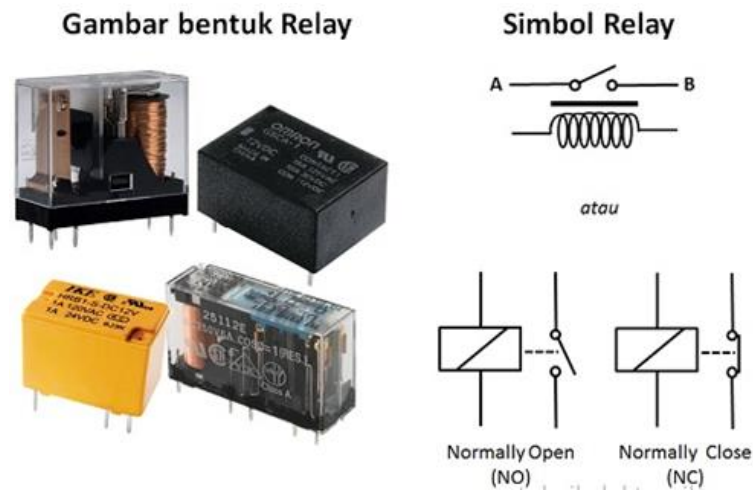
speakers digunakan dalam organ Hammond. pedal efek segera menjadi favorit gitaris Jimi Hendrix dan Robin Trower. Setelah pertamakali mendengar Octavia, Hendrix diduga bergegas kembali ke studio dan segera menggunakannya untuk merekam solo gitar pada "*Purple Haze*" dan "*Fire*". Pada pertengahan 1970-an berbagai efek solid-state pedal termasuk *flangers*, *chorus pedals*, *ring modulators* dan *phase-shifters* mulai tersedia. Pada 1980-an, unit rackmount digital mulai menggantikan stompboxes sebagai format efek pilihan. Seringkali musisi akan menggunakan "*Dry*", trek berubah di studio dan efek dalam akan ditambahkan pasca-produksi. Keberhasilan tahun 1991 album Nirvana's *Nevermind* membantu menyalakan kembali minat stompboxes. Sepanjang tahun 1990-an, musisi berkomitmen untuk sebuah "lo-fi" aesthetic seperti J Mascis dari Dinosaur Jr, Stephen Malkmus dari Perkerasan dan Robert Pollard dipandu terus oleh suara efek menggunakan non-digital (analog) pedal efek.

### 2.3 Relay

Menurut Bishop (2004:55) Relay adalah sebuah saklar yang di kendalikan oleh arus. Relay memiliki sebuah kumparan tegangan rendah yang dililitkan pada sebuah inti dan arus nominal yang harus dipenuhi output rangkaian pendriver atau pengemudinya. Arus yang digunakan pada rangkaian adalah arus DC.

Relay menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil (low power) dapat menghantarkan listrik yang bertegangan lebih tinggi. Sebagai contoh, dengan relay yang menggunakan elektromagnet 5V dan 50 mA mampu menggerakkan armature relay (yang berfungsi sebagai saklarnya) untuk menghantarkan listrik 220V 2A.

Bisa dilihat pada gambar 2.1 bentuk relay dan simbol relay.



**Gambar 2.1 Bentuk dan Simbol relay**

Pada dasarnya, Relay terdiri dari 4 komponen dasar yaitu:

- a. Electromagnet (Coil)
- b. Armature
- c. Switch Contact Point (Saklar)
- d. Spring

## 2.4 Mikrokontroler ATmega 8535

### 2.4.1 Pengertian Mikrokontroler

Menurut Chamim (2012) Mikrokontroler adalah sebuah sistem komputer yang seluruh atau sebagian besar elemennya dikemas dalam satu *chip IC*, sehingga sering disebut *single chip microcomputer*. Mikrokontroler merupakan sistem komputer yang mempunyai salah satu atau beberapa tugas yang sangat spesifik.

### 2.4.2 Pengertian Mikrokontroler ATmega 8535

Arsitektur mikrokontroler jenis AVR (*Alf and Vegard's Risc processor*) pertama kali dikembangkan pada tahun 1996 oleh dua orang mahasiswa *Norwegian Institute of Technology* yaitu *Alf-Egil Bogen* dan *Vegard Wollan*. Mikrokontroler AVR kemudian dikembangkan lebih lanjut oleh *Atmel*. Seri pertama AVR yang dikeluarkan adalah mikrokontroler 8 bit AT90S8515, dengan konfigurasi pin yang

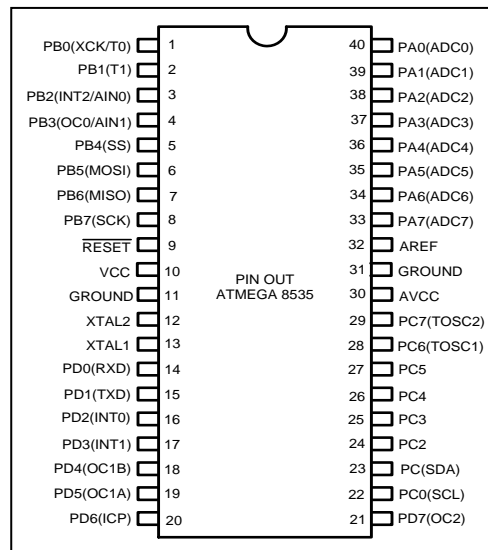
sama dengan mikrokontroler 8051, termasuk *address* dan *data bus* yang termultipleksi.

Mikrokontroler AVR menggunakan teknologi RISC (*Reduced Instruction Set Computer*) dimana set instruksinya dikurangi dari segi ukurannya dan kompleksitas *mode* pengalamatannya. Pada awal era industri komputer, bahasa pemrograman masih menggunakan kode mesin dan bahasa *assembly*. Untuk mempermudah dalam pemrograman para *desainer* komputer kemudian mengembangkan bahasa pemrograman tingkat tinggi yang mudah dipahami manusia. Namun akibatnya, instruksi yang ada menjadi semakin kompleks dan membutuhkan lebih banyak memori. Dan tentu saja siklus eksekusi instruksinya menjadi semakin lama. Dalam AVR dengan arsitektur RISC 8 bit, semua instruksi berukuran 16 bit dan sebagian besar dieksekusi dalam 1 siklus *clock*. (Agus Bejo,2008 : 3).

Dalam perkembangannya, AVR dibagi menjadi beberapa varian yaitu AT90Sxx, ATmega, AT86RFxx dan ATTiny. Pada dasarnya yang membedakan masing-masing varian adalah kapasitas memori dan beberapa fitur tambahan saja. Mikrokontroler AVR ATmega8535 merupakan IC CMOS 8-bit yang memanfaatkan daya rendah dalam pengoperasiannya dan berbasis pada arsitektur RISC AVR. ATmega 8535 dapat mengeksekusi satu instruksi dalam sebuah siklus *clock*, dan dapat mencapai 1 MIPS perMHz, sehingga para perancang dapat mengoptimalkan penggunaan daya rendah dengan kecepatan tinggi. (Wahyudin,2007:3)

#### **2.4.3 Konfigurasi Pin AVR ATmega 8535**

Mikrokontroler AVR ATmega 8535 mempunyai jumlah kaki sebanyak 40, dimana 32 kaki digunakan untuk keperluan port paralel yang dapat menjadi pin *input/output*. Pada 32 kaki tersebut terbagi atas 4 bagian (port), pada masing-masing port terdiri atas 8 kaki. Konfigurasi pin Mikrokontroler AVR ATmega 8535 dapat dilihat pada gambar 2.2.



**Gambar 2.2 Pin Out AVR ATmega 8535**

Penjelasan Pin :

- a. VCC : Tegangan Supply (5 Volt)
- b. GND : *Ground*
- c. Port A (PA0-PA7) : Port A berfungsi sebagai input analog ke ADC.  
Port A juga berfungsi sebagai suatu *port* I/O 8-bit bidirectional, jika ADC tidak digunakan. Pin port dapat menyediakan resistor pull-up internal (dipilih untuk setiap bit).
- d. Port B (PB0-PB7) : Port B merupakan *port* I/O 8 bit bidirectional dengan resistor pull-up internal (dipilih untuk setiap bit).
- e. Port C (PC0-PC7) : Port C merupakan port I/O 8 bit bidirectional dengan resistor pull-up internal (dipilih untuk setiap bit).
- f. Port D (PD0-PD7) : Port D merupakan port I/O 8 bit bidirectional dengan resistor pull-up internal (dipilih untuk setiap bit).
- g. Reset : Input reset. Level rendah pada pin ini selama lebih dari panjang pulsa minimum akan menghasilkan reset, walaupun *clock* sedang berjalan. Reset digunakan untuk mengulang ke posisi awal dan mengosongkan memori.

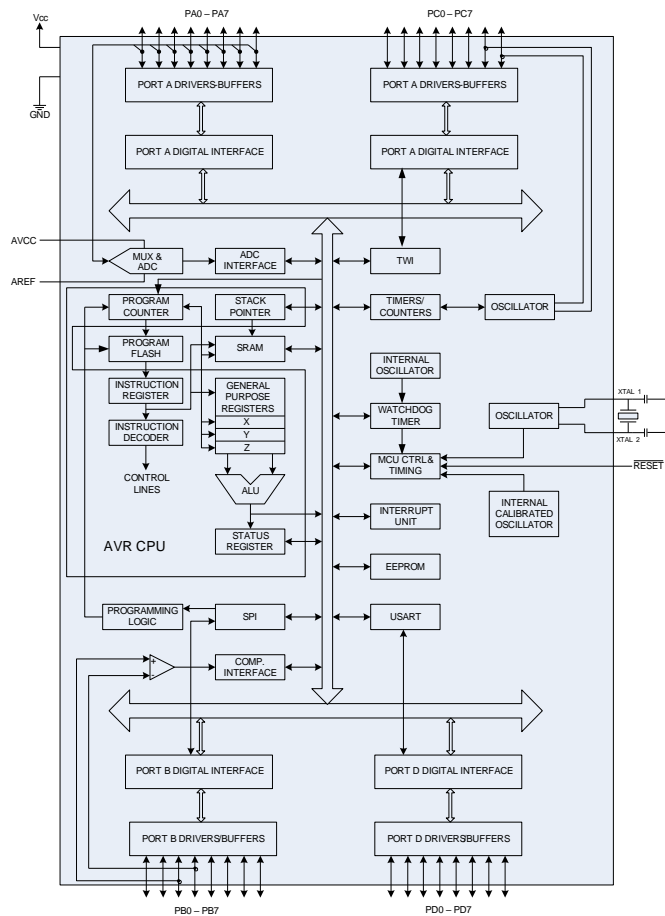
- h. XTAL1 : Input penguat osilator inverting dan input pada rangkaian operasi *clock* internal.
- i. XTAL2 : Output dari penguat osilator inverting.
- j. AVCC : AVCC adalah pin tegangan supply untuk *portA* dan ADC. Pin ini harus dihubungkan ke VCC walaupun ADC tidak digunakan. Jika ADC digunakan, maka pin ini harus dihubungkan ke Vcc melalui *low pass filter*.
- k. AREF : AREF adalah pin referensi tegangan analog untuk ADC.

Adapun fitur-fitur yang dimiliki oleh mikrokontroler AVR ATmega 8535 adalah sebagai berikut:

- a. 8 Kbyte *In-SystemProgrammable Flash*.
- b. 512 byte EEPROM dan SRAM.
- c. 32 *general purpose I/O* dan *register*.
- d. 3 buah *Timer/counter* dengan *mode compare*.
- e. *Interrupt* internal dan eksternal.
- f. Antar muka serial *Two-Wire* dengan orientasi byte.
- g. 8-channel ADC 10 bit.
- h. *Watchdog timer* yang dapat diprogram dengan osilator *internal*.

#### 2.4.4 Diagram Blok Mikro AVR ATmega 8535

Pada diagram blok Mikrokontroler AVR ATmega 8535 digambarkan 32 *general purpose working register* yang dihubungkan secara langsung dengan *Arithmetic Logical Unit (ALU)*, sehingga dimungkinkan dua register yang berbeda dapat di *aces* dalam satu siklus *clock*. Diagram blok Mikrokontroler AVR ATmega 8535 dapat dilihat pada gambar 2.3.



**Gambar 2.3 Diagram Blok AVR ATmega 8535**

#### 2.4.5 Memori Mikrokontroler AVR ATmega8535

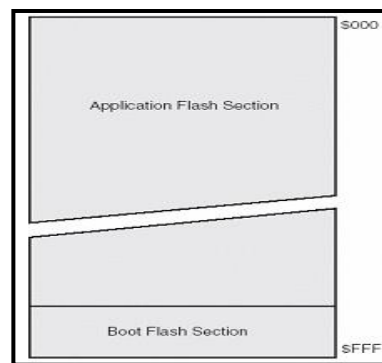
Mikrokontroler AVR ATmega 8535 mempunyai dua ruang memori utama yaitu Ruang Data Memori dan Ruang Program Memori, sebagai tambahan Mikrokontroler AVR ATmega 8535 memiliki fitur suatu EEPROM memori untuk penyimpanan data.

##### a. Program Memory

Mikrokontroler AVR ATmega 8535 memiliki *On-Chip In-System Reprogrammable Flash Memory* untuk menyimpan program. Untuk alasan keamanan, program memory dibagi menjadi dua bagian yaitu *Boot Flash Section* dan *Application Flash Section*. *Boot Flash Section* digunakan untuk menyimpan program *Boot Loader*, yaitu program yang harus dijalankan pada saat AVR reset atau pertama kali diaktifkan. *Application Flash Section* digunakan untuk



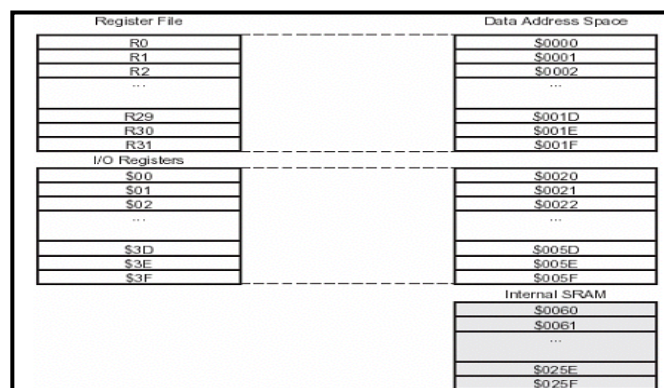
menyimpan program aplikasi yang dibuat *user*. AVR tidak dapat menjalankan program aplikasi ini sebelum menjalankan program *Boot Loader*. Besarnya memori *Boot Flash Section* dapat diprogram dari 128 *word* sampai 1024 *word* tergantung *setting* pada konfigurasi bit di register *BOOTSZ*. Jika *Boot Loader* diproteksi, maka program pada *Application Flash Section* juga sudah aman. Peta memori dapat dilihat pada gambar 2.4.



**Gambar 2.4 Peta Program Memori**

b. Data Memory

Gambar 2.4 menunjukkan peta data memori SRAM pada Mikrokontroler AVR ATmega 8535. Terdapat 608 lokasi *address* data memori. 96 lokasi *address* digunakan untuk *Register File* dan *I/O Memory* sementara 512 lokasi *address* lainnya digunakan untuk *internal data* SRAM. *Register File* terdiri dari 32 *general purpose working register*, *I/O register* terdiri dari 64 *register*. Peta data memori dapat dilihat pada gambar 2.5



**Gambar 2.5 Peta Data Memori**

c. EEPROM Data *Memory*

Mikrokontroler AVR ATmega 8535 memiliki EEPROM sebesar 512 *byte* untuk menyimpan data, memori ini diatur secara terpisah sehingga dapat dibaca dan ditulis per *byte*nya secara tersendiri. EEPROM ini memiliki daya tahan tulis atau hapus hingga 10.000 kali. Lokasinya terpisah dengan sistem *address register*, *data register* dan *control register* yang dibuat khusus untuk EEPROM.

## 2.5 Gitar

### 2.5.1 Pengertian Gitar

Gitar adalah alat musik berdawai yang dimainkan dengan jari-jemari tangan atau sebuahplektrum (alat petik gitar). Bunyinya dihasilkan dari senar-senar yang bergetar. Gitar bisa berupa gitar akustik atau listrik, atau gabungan keduanya.

Pengertian gitar akustik adalah jenis gitar dimana suara yang dihasilkan berasal dari getaransenar gitar yang dialirkan melalui sadel dan jembatan tempat pengikat senar ke dalam ruang suara. Suara di dalam ruang suara ini kan beresonansi terhadap kayu badan gitar. Jenis kayu akan mempengaruhi suara yang dihasilkan oleh gitar akustik.

Pengertian gitar listrik (elektrik) adalah gabungan komponen gitar itu sendiri dan bantuan berupa mic buat gitar /atau sering disebut *pick up*, yang di hubungkan dengan peralatan listrik lainnya seperti efek gitar untuk merubah jenis suara gitar itu sendiri dan biasanya menggunakan speker besar atau amplifier untuk menaikkan volume dari suara gitar itu sendiri.

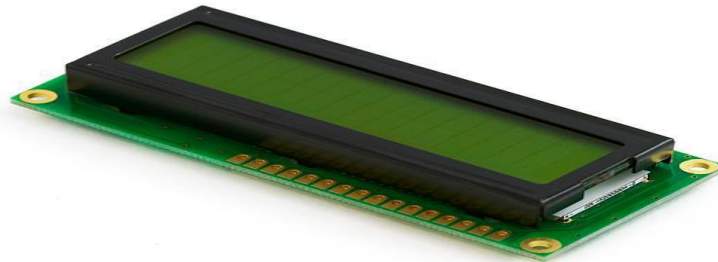
### 2.5.2 Sejarah Singkat Gitar

Kata 'gitar' atau gitar dalam bahasa Inggris, pada mulanya diambil dari nama alat musik petik kuno di wilayah Persia pada kira-kira tahun 1500 SM yang dikenal sebagai *citar* atau *sehtar*. Alat musik ini kemudian berkembang menjadi berbagai macam model gitar kuno yang dikenal dengan istilah umum *tanbur*. Pada tahun 300 SM *Tanbur* Persia dikembangkan oleh bangsa Yunani dan enam abad kemudian oleh bangsa Romawi (Bellow, 1970:5455).

Pada tahun 476M alat musik ini dibawa oleh bangsa Romawi ke Spanyol dan bertransformasi menjadi: (1) guitarra Morisca yang berfungsi sebagai pembawa melodi, dan (2) Guitarra Latina untuk memainkan akor. Tiga abad kemudian bangsa Arab membawa semacam gitar gambus dengan sebutan al ud ke Spanyol (Summerfield, 1982:12). Berdasarkan konstruksi al ud Arab dan kedua model gitar dari Romawi tersebut, bangsa Spanyol kemudian membuat alat musiknya sendiri yang disebut *vihuela*. Sebagai hasilnya, *vihuela* menjadi populer di Spanyol sementara alat-alat musik pendahulunya sedikit demi sedikit ditinggalkan. Walaupun demikian al ud dibawa orang ke negara-negara Eropa Barat dan menyaingi popularitas *vihuela* di Spanyol. Di Eropa al ud disambut dengan baik dan berkembang menjadi berbagai model lute Eropa hingga kira-kira akhir abad ke-17. Sementara itu *vihuela* berkembang terus menjadi berbagai macam gitar selama berabad-abad hingga akhirnya menjadi gitar klasik yang digunakan pada saat ini.

## **2.6 LCD (*Liquid Crystal Display*)**

Layar LCD merupakan suatu media penampilan data yang sangat efektif dan efisien dalam penggunaannya. Untuk menampilkan sebuah karakter pada layar LCD diperlukan beberapa rangkaian tambahan. Untuk lebih memudahkan para pengguna, maka beberapa perusahaan elektronik menciptakan modul LCD. Modul LCD memiliki 3 jalur kontrol yang bernama RS, R/W, dan E. RS digunakan untuk memberitahukan kepada LCD apakah data yang diberikan adalah kata instruksi (instruction word) atau kata data (data word). Jika akan mengirim instruksi, RS harus dibuat 0, sedangkan untuk mengirimkan data RS harus berlogika 1. Bentuk fisik LCD 16x2 ditunjukkan pada gambar 2.6 dibawah ini:



**Gambar 2.6 Bentuk Fisik LCD 16x2**

LCD dibagi menjadi dua bagian yaitu bagian depan panel LCD yang terdiri dari banyak dot atau titik LCD dan Microcontroller yang menempel pada bagian belakang panel LCD yang berfungsi untuk mengatur titik-titik LCD sehingga dapat menampilkan huruf, angka, dan simbol khusus yang dapat terbaca (Paulus dkk, 2004).

## **2.7 Komunikasi Serial**

Komunikasi serial adalah komunikasi yang pengiriman datanya per-bit secara berurutan dan bergantian. Komunikasi ini mempunyai suatu kelebihan yaitu hanya membutuhkan satu jalur dan kabel yang sedikit dibandingkan dengan komunikasi paralel. Pada prinsipnya komunikasi serial merupakan komunikasi dimana pengiriman data dilakukan per bit sehingga lebih lambat dibandingkan komunikasi paralel, atau dengan kata lain komunikasi serial merupakan salah satu metode komunikasi data di mana hanya satu bit data yang dikirimkan melalui seuntai kabel pada suatu waktu tertentu (Widodo, 2004). Berikut pada gambar 2.7 merupakan bagian-bagian dari port serial.



Keterangan mengenai saluran RS232 pada konektor DB-9 sebagai berikut:

- *Received Line Signal Detect*, dengan saluran ini DCE memberitahukan ke DTE bahwa pada terminal masukan ada data masuk.
- *Receive Data*, digunakan DTE menerima data dari DCE.
- *Transmit Data*, digunakan DTE mengirimkan data ke DCE.
- *Data Terminal Ready*, pada saluran ini DTE memberitahukan kesiapan sinyalnya.
- *Signal Ground*, saluran *Ground*.
- *Ring Indicator*, pada saluran ini DCE memberitahukan ke DTE bahwa sebuah stasiun menghendaki hubungan dengannya.
- *Clear to Send*, dengan saluran ini DCE memberitahukan bahwa DTE boleh mulai mengirimkan data.
- *Request to Send*, dengan saluran ini DCE diminta mengirim data oleh DTE.
- *DCE Ready*, sinyal aktif pada saluran ini menunjukkan bahwa DCE sudah siap.

## 2.8 Pengenalan *CodeVision-AVR*

*CodeVisionAVR* pada dasarnya merupakan perangkat lunak pemrograman mikrokontroler keluarga AVR berbasis bahasa C. Ada tiga komponen penting yang telah diintegrasikan dalam perangkat lunak ini: Compiler C, IDE dan Program generator.

Berdasarkan spesifikasi yang dikeluarkan oleh perusahaan pengembangnya, Compiler C yang digunakan hampir mengimplementasikan semua komponen standar yang ada pada bahasa C standar ANSI (seperti struktur program, jenis tipe data, jenis operator, dan library fungsi standar-berikut penamaannya). Tetapi walaupun demikian, dibandingkan bahasa C untuk aplikasi komputer, compiler C untuk microcontroller ini memiliki sedikit perbedaan yang disesuaikan dengan arsitektur AVR tempat program C tersebut ditanamkan (*embedded*). Khusus untuk library fungsi, disamping library standar (seperti fungsi-fungsi matematik, manipulasi String, pengaksesan memori dan sebagainya),

CodeVisionAVR juga menyediakan fungsi-fungsi tambahan yang sangat bermanfaat dalam pemrograman antarmuka AVR dengan perangkat luar yang umum digunakan dalam aplikasi kontrol. Beberapa fungsi library yang penting diantaranya adalah fungsi-fungsi untuk mengakses LCD, komunikasi I2C, IC RTC (Real time

Integrated Development Environment (IDE) telah diadaptasikan pada chip AVR yaitu In-System Programmer software, memungkinkan programmer untuk mentransfer program ke chip mikrokontroler secara otomatis setelah proses assembly/kompilasi berhasil. In-System Programmer software didesign untuk bekerja dan dapat berjalan dengan perangkat lunak lain seperti AVR Dragon, AVRISP, Atmel STK500, dan lain sebagainya.

CodeVisionAVR juga memiliki CodeWizardAVR sebagai generator program otomatis, yang memungkinkan kita untuk menulis, segala bentuk pengaturan Chip dalam waktu singkat, dan semua kode yang dibutuhkan. Pada gambar 2.8 bisa di lihat tampilan awal dari code vision AVR.

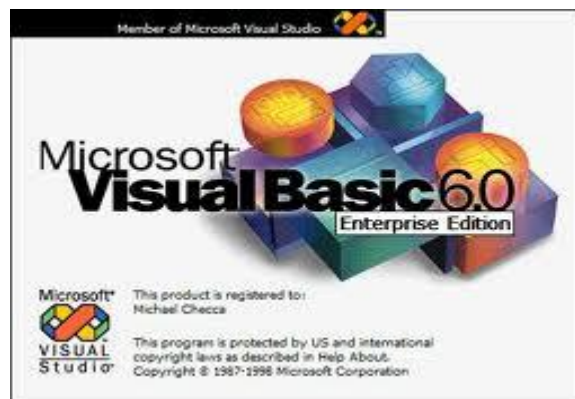


**Gambar 2.8 Tampilan Awal Code Vision AVR**

## 2.9 Microsoft Visual Basic 6.0

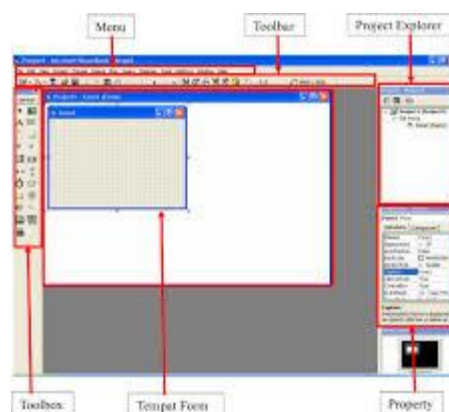
Pemrograman Visual (*Visual Basic*) merupakan salah satu bahasa pemrograman yang memanfaatkan suatu lingkungan/ tampilan secara visual sebagai antarmuka dengan usernya.

*Microsoft Visual Basic* adalah merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment (IDE) visual* untuk membuat program perangkat lunak berbasis sistem operasi *Microsoft Windows* dengan menggunakan model pemrograman BASIC dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat.



**Gambar 2.9 Tampilan Visual Basic 6.0**

### 2.9.1 Komponen Visual Basic 6.0



**Gambar 2.10 Tampilan Awal Visual Basic 6.0**

Komponen-komponen dari lingkungan Visual Basic tersebut antara lain adalah:



### 1. Baris Menu

Menu merupakan kumpulan perintah-perintah yang dikelompokkan dalam kriteria operasi yang dihasilkan. Visual Basic 6.0 menyediakan tiga belas menu.

### 2. Toolbar

Kehadiran tombol-tombol speed pada toolbar akan sangat membantudalam mempercepat akses perintah (yang bias jadi tersembunyi di dalam tingkat-tingkat hirarki). Sebab tombol speed berfungsi sama dengan perintah yang tersedia (dan tersembunyi) di dalam menu.

### 3. Form

Form adalah bahan untuk pembuatan window. Kita meletakkan kontrol pada form. Kontrol ini misalnya tombol, *check box*, *radio button*, *memo label*, panel dan sebagainya. Pada form tersedia tombol *minimize/restore* dan *close*, ketiganya terletak di pojok kanan atas.

Ukuran form bisa diubah dengan drag dan drop tiga titik di sebelah kanan, bawah dan pojok kanan bawah. Jika drag ke arah ke luar akan memperbesar dan sebaliknya akan memperkecil.

### 4. Toolbox

*Toolbox* adalah tempat penyimpanan kontrol yang akan kita gunakan pada program yang dipasangkan pada form.

### 5. Project Explorer

*Project Explorer* berfungsi berbagai saran pengakses bagian-bagian pembentuk *project*. Pada windows ini terdapat tiga tombol pengaktif untuk *Windows Code*, *Windows Object* dan *Toggle Folder*. Juga terdapat diagram yang menampilkan susunan *folder* penyimpanan *file-file project*.

### 6. Window Properties

*Window* ini bertugas menyiapkan segala properti dari objek yang diperlukan dalam perancangan user interface maupun pemrograman.

Pada *window* ini terdapat semua properti yang dimiliki oleh objek terpilih (cara memilih objek adalah klik objek langsung pada diagram di *project explorer* atau klik langsung pada objeknya, misalnya *form*).

Pada *windows* ini terdapat dua *tab* yang menampilkan properti dalam dua cara sesuai dengan nama tab yaitu *tab Alphabet* (diurutkan berdasarkan namanya

sesuai dengan abjad, ini merupakan pilihan default) dan *Categories* (diurutkan berdasarkan fungsinya). Masing-masing properties memiliki nilainya sendiri-sendiri yang telah disediakan *Visual Basic 6.0* atau kita isikan sesuai dengan kebutuhan.

### **7. Window Form Layout**

Bisa digunakan untuk mengatur tata letak form pada layar monitor. Seringkali kita salah menempatkan form sehingga untuk mendapatkan posisi yang kita inginkan, setiap kali kita harus menjalankan program untuk mengetahui posisi dari hasil penyetelan yang kita lakukan. Dengan adanya window form layout ini pekerjaan yang berulang-ulang yang tidak kita inginkan tersebut bisa dihindari.

### **8. Objek Komponen**

Merupakan komponen dalam program serta elemen antar muka yang di buat pada *form Visual Basic* menggunakan *Control Toolbox*. Kita harus memindahkan, mengubah ukurannya, mengatur letak objek (*Setting Property*). Objek memiliki *Inherent Functional* yaitu objek mengetahui cara beroperasi dan merespon pada kondisi tertentu.

Contoh – contoh Objek dan Komponen pada *Visual Basic* beserta fungsinya :

1. *TextBox* (tempat input/menampilkan teks -> bebas mengubah)
2. *Label* (teks pada form; tidak dapat langsung di ubah)
3. *Command Button* (menjalankan perintah/proses jika di tekan)
4. *Frame* (pengelompokan kontrol-kontrol secara visual/fungsional)
5. *CheckBox* (menampilkan kondisi Y/N atau T/F)
6. *Form* (tempat kerja program aplikasi)
7. *ComboBox* (kombinasi textbox & listbox, pilih dari list atau ketik)
8. *ListBox* (multiselect)
9. *PictureBox* (menampilkan gambar)
10. *Image* (menampilkan gambar, tapi tidak bisa sebagai kontainer kontrol)
11. *Option Button* (mirip dengan *checkbox*, tapi hanya satu pilihan yg dapat di pilih)
12. *HS/VS ScrollBar* (scrollbar Horizontal dan Vertikal)
13. *Line* (membentuk garis)

14. *Data* (data binding)
15. *OLE – Object Linking and Embedding* (lokasi program eksternal, ex : word ms, dll.)
16. *Project* (sekumpulan modul aplikasi itu sendiri)
17. *Module* (modul fungsi-fungsi kontrol)
18. *Code Editor* (tempat sintaks/coding)
19. *DBGrid* (menampilkan item-item database)
20. *Pointer* (untuk memilih kontrol)

Beberapa bahasa skrip seperti *Visual Basic for Applications* (VBA) dan *Visual Basic Scripting Edition* (VBScript), mirip halnya *Visual Basic*, tetapi cara kerjanya yang berbeda. Para programmer dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh *Microsoft Visual Basic* program-program yang ditulis dengan *Visual Basic* juga dapat menggunakan *Windows API*, tapi membutuhkan deklarasi fungsi luar tambahan.

*Visual Basic* juga telah berkembang menjadi beberapa versi, bagaimanapun juga *Visual basic 6* tetap menjadi versi yang populer karena mudah dalam membuat programnya dan ia tidak menghabiskan banyak memori (komputer). Sejarah *BASIC* ditangan *Microsoft* sebagai bahasa diinterpretasi (*BASICA*) dan juga bahasa yang dikompilasi (*BASCOM*) membuat *Visual Basic* diimplementasikan sebagai gabungan keduanya. Programmer yang menggunakan *Visual Basic* bisa memilih kode bahasa pemrograman yang dikompilasi atau kode yang harus bahasa pemrograman yang diinterpretasikan sebagai hasil Porting dari kode VB.

*Visual Basic* merupakan bahasa yang mendukung Pemrograman berorientasi objek, namun tidak sepenuhnya, Beberapa karakteristik objek tidak dapat dilakukan pada *Visual Basic*, seperti *Inheritance* tidak dapat dilakukan pada *class module*, *polymorphism* secara terbatas bisa dilakukan dengan mendeklarasikan *class module* yang memiliki Interpace tertentu. *Visual Basic* tidak bersifat *case sensitif*.

*Visual Basic* menjadi populer karena kemudahan desain form secara visual dan adanya kemampuan untuk menggunakan komponen-komponen *ActiveX* yang

dibuat oleh pihak lain. Namun kompone ActiveX memiliki masalahnya tersendiri yang dikenal sebagai DLL hell, pada Visual Basic .NET, Microsoft mencoba mengatasi masalah DLL hell dengan mengubah cara penggunaan komponen (menjadi independen terhadap registry).

Pembuatan program aplikasi menggunakan *visual basic* dilakukan dengan membuat tampilan aplikasi pada *form*, kemudian diberi sprict program di dalam komponen-komponen yang diperlukan. Form disusun oleh komponen-komponen yang berada do toolbox, dan setiap komponen yang dipakai harus diatur propertinya lewat jendela properti.

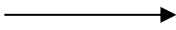
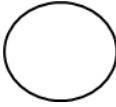

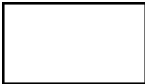
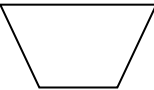
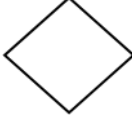
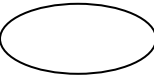


## **2.10 Flowchart**

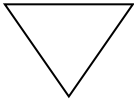
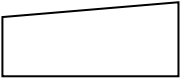

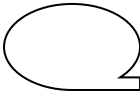
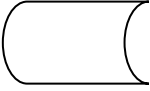


### **2.10.1 Pedoman Menggambar Flowchart**

Pedoman dalam menggambar suatu bagan alir, analis sistem atau pemrograman sebagai berikut;

- a. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
- b. Kegiatan didalam bagan alir harus ditunjukkan dengan jelas.
- c. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhirnya.
- d. Masing-masing kegiatan didalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan, misalnya “persiapkan” dokumen “hitung” gaji.
- e. Masing-masing kegiatan didalam bagan alir harus didalm urutan yang semestinya.
- f. Kegiatan yang terpotong dan akan disambung ketempat lain harus ditunjukkan dengan jelas menggunakan symbol penghubung.
- g. Gunakanlah symbol-simbol bagan alir yang standar.

Tabel 2.2 Simbol-Simbol *Flowchart*

No	Simbol	Keterangan
1		Simbol arus / <i>flow</i> , yaitu menyatakan jalannya arus suatu proses
2		Simbol connector, berfungsi menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
3		Simbol offline connector, menyatakan sambungan dari proses ke proses lainnya dalam halaman berbeda
4		Simbol process, yaitu menyatakan suatu tindakan (proses) yang dilakukan oleh komputer
5		Simbol manual, yaitu menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
6		Simbol decision, yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya atau tidak
7		Simbol terminal, yaitu menyatakan permulaan atau akhir suatu program
8		Simbol predefined process, yaitu menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
9		Simbol keying operation, menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard

10		Simbol offline-storage, menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
11		Simbol manual input, memasukkan data secara manual dengan menggunakan online keyboard
12		Simbol input/output, menyatakan proses input atau output tanpa tergantung jenis peralatannya
13		Simbol magnetic tape, menyatakan input berasal dari pita magnetis atau output disimpan ke dalam pita magnetis
14		Simbol disk storage, menyatakan input berasal dari disk atau output disimpan ke dalam disk
15		Simbol document, mencetak keluaran dalam bentuk dokumen (melalui printer)
16		Simbol punched card, menyatakan input berasal dari kartu atau output ditulis ke kartu.