BAB II

DASAR TEORI

2.1 Pengenalan Ucapan

Sistem Pengenalan Ucapan (*Speech Recognition Sistem*) adalah sistem yang berfungsi untuk mengubah bahasa lisan menjadi bahasa tulisan. Masukan sistem adalah ucapan manusia, selanjutnya sistem akan mengidentifikasikan kata atau kalimat yang diucapkan dan menghasilkan teks yang sesuai dengan apa yang diucapkan. Sinyal ucapan pertama kali akan dilewatkan pada bagian penganalisis ucapan untuk mendapatkan besaran-besaran atau ciri-ciri yang mudah diolah pada tahap berikutnya. Untuk setiap ucapan yang berbeda akan dihasilkan pola ciri yang berbeda

Penganalisis sintaks akan melakukan transformasi sinyal ucapan dari domain waktu ke domain frekuensi. Pada domain frekuensi, untuk kurun waktu yang singkat, setiap sinyal dapat terlihat memiliki ciri-ciri yang unik. Namun demikian, pengucapan suatu unit bunyi ucapan (*fonem*) seringkali bervariasi antar orang yang berbeda, juga terpengaruh oleh fonem-fonem disekitarnya, kondisi emosi, *noise*, dan faktor-faktor lainnya. Sistem *Speech Recognition* akan melakukan pengenalan untuk setiap unit bunyi pembentuk ucapan (*fonem)*, selanjutnya mencoba mencari kemungkinan kombinasi hasil ucapan yang paling dapat diterima. Sistem yang lebih sederhana adalah sistem yang hanya dapat mengenal sejumlah kata yang jumlahnya terbatas. Sistem ini biasanya lebih akurat

dan lebih mudah dilatih, tetapi tidak dapat mengenal kata yang berada di luar kosa kata yang pernah diajarkan.

Ada 2 tipe *Speech Recognition*, dilihat dari ketergantungan pembicara yaitu:

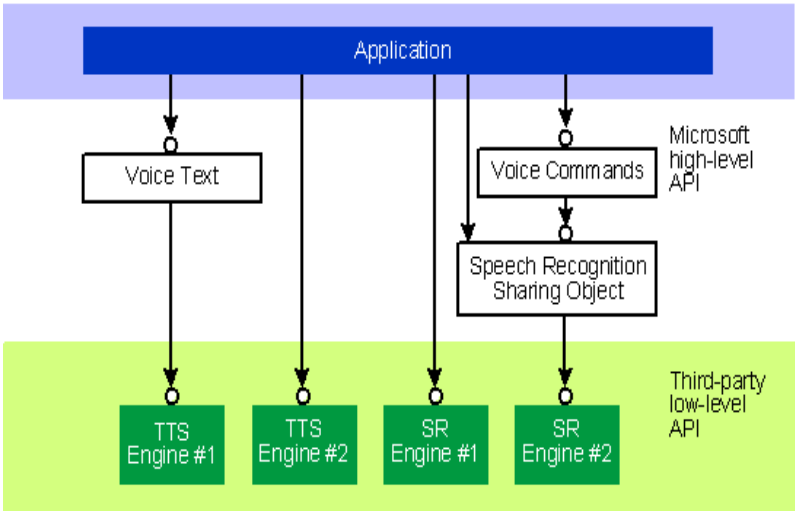
a. *Independent Speech Recognition*, yaitu sistem pengenal ucapan tanpa terpengaruh dengan siapa yang berbicara, tetapi mempunyai keterbatasan dalam jumlah kosakata. Model ini akan mencocokan setiap ucapan dengan kata yang dikenali dan memilih yang ”sepertinya” cocok. Untuk mendapatkan kecocokan kata yang diucapkan maka digunakan model statistic yang dikenal dengan nama *Hidden Markov Model* (HMM)

b. *Dependent Speech Recognition*, yaitu sistem pengenal ucapan yang memerlukan pelatihan khusus dari pembicara, dimana hasil pelatihan dari masing-masing pembicara akan disimpan dalam sebuah profil. Profil inilah yang nantinya digunakan untuk berinteraksi dengan sistem pengenal ucapan dan sistem akan bergantung siapa yang berbicara. Sistem ini biasanya lebih mudah untuk dikembangkan, dimana contoh suara sudah dibuat sebelumnya dan disimpan dalam database (basis data) dan jumlah kosakatanya lebih besar dibandingkan dengan *independent speech recognition*. Proses pengenalan ucapan dengan cara membandingkan ucapan pembicara dengan contoh suara yang sudah ada.

2.2 *Microsoft SAPI 5.1*

Speech Application Programming Interface (SAPI) adalah sebuah API yang dikembangkan oleh Microsoft yang digunakan sebagai pengenal suara didalam lingkungan pemrograman aplikasi Windows. Sampai saat ini SAPI

dikemas baik berupa SDK (*Sistem Development Kit*) maupun disertakan dalam sistem operasi Windows itu sendiri. Aplikasi yang telah menggunakan SAPI antara lain *Microsoft Office*, dan *Windows Vista*. Secara arsitektur pemrograman SAPI dapat dilihat sebagai sebuah *middleware* yang terletak antara aplikasi dan *speech engine*. Di dalam SAPI versi 1 sampai dengan 5, aplikasi dapat berkomunikasi langsung dengan *speech engine* seperti tampak pada gambar berikut:



Gambar 2.1 Arsitektur Microsoft SAPI

Komponen utama di dalam SAPI 5 adalah sebagai berikut:

a. *Voice Command*, sebuah obyek level tinggi untuk perintah dan kontrol menggunakan pengenalan suara.

b. *Voice Dictation*, sebuah obyek level tinggi untuk continous *dictation speech recognition.*

c. *Voice Talk*, sebuah obyek level tinggi untuk *speech synthesis*.

d. *Voice Telephony*, sebuah obyek untuk menulis aplikasi telepon berbasiskan pengenalan suara.

e. *Direct Speech Recognition*, sebuah obyek sebagai mesin untuk mengontrol pengenalan suara (*direct control of recognition engine*)

f. *Direct Text to Speech*, sebuah obyek sebagai mesin yang mengontrol synthesis.

g. *Audio Object*, untuk membaca dari audio device atau sebuah *file audio*

Option Explicit

1.3 CodeVisionAVR

Dewasa ini penggunanaan bahasa pemrograman tingkat tinggi (seperti C, Basic, Pascal, Forth dan sebagainya) semakin popular dan banyak digunakan untuk memprogram sistem mikrokontroler. Berdasarkan sifatnya yang sangat *fleksibel* dalam hal kelulusan pemrogram untuk mengakses perangkat keras,

Bahasa C dikembangkan pertama kali oleh Dennis Ritchie dan Ken Thomson pada tahun 1972, bahasa C merupakan salah satu pemrograman yang paling populer untuk pengembangan program-program aplikasi yang berjalan pada sistem *microprocessor* (komputer). Karena kepopulerannya, vendor-vendor perangkat lunak kemudian mengembangkan compiler C sehingga menjadi beberapa varian berikut: Turbo C, Borland C, Microsoft C, Power C, Zortech C dan lain sebagainya.

Untuk menjaga portabilitas, *compiler-compiler* C tersebut menerapkan ANSI C *American National Standards Institute* (ANSI) sebagai standar bakunya. Perbedaan antara *compiler-compiler* tersebut umumnya hanya terletak pada pengembangan fungsi-fungsi *library* serta fasilitas *Integrated Development Environment* (IDE) saja.

Dibandingkan dengan bahasa assembler, penggunaan bahsa C dalam pemrograman memiliki beberapa kelebihan berikut, memepercepat waktu pengembangan, bersifat modular dan terstriktur, sedangkan kelemahannya adalah kode program hasil kompilasi akan relative lebih besar (dan sebagai konsekuensinya hal ini terkadang akan mengurangi kecepatan eksekusi). Khusus pada mikrokontroler AVR, untuk mereduksi konsekuensi negatif diatas, perusahaan atmel merancang sedemikian sehingga arsitektur AVR ini efisien dalam mendekode serta mengeksekusi instruksi-instruksi yang umum dibangkitkan oleh *compiler* C. Dalam kenyataannya, pengembangan arsitektur AVR ini tidak dilakukan sendiri oleh perusahaan atmel tetapi ada kerja sama dengan salah satu vendor pemasok *compiler* C untuk mikrokontroler tersebut, yaitu IAR C *software* atau pemrograman. Bagian tersebut digunakan dalam pengembangan robot dengan komponen *Programmable Logic Control* (PLC).

Komponen *programmable* adalah komponen yang terbuat dari bahan semikonduktor dan merupakan integrasi dari berbagai komponen elektronika didalamnya dan dapat deprogram ulang demi tujuan pengendalian robot. Komponen *programmable* digunakan oleh robot-robot yang dibuat untuk melakukan tugas-tugas beragam dan biasanya bekerja secara otomatis. Dengan demikian, robot-robot yang bersifat *unprogrammable* (sederhana), yang bisa bergerak hanya dengan rangkaian elektronika analog biasa, tidak memerlukan program atau *software* tersebut. Adapun instruksi-instruksi pada bahasa C di CodeVision AVR:

1. Header File

   Adalah berkas yang berisi *prototype* fungsi definisi konstanta dan definisi *variable*. Fungsi adalah kumpulan *code* C yang diberi nama dan ketika nama tersebut dipanggil maka kumpulan kode tersebut dijalankan.

   Contoh:

   stdio.h

   math.h

2. *#include*

   Preprosesor directive adalah bagian yang berisi pengikutsertaan file atau berkasberkas fungsi maupun pendefinisian konstanta.

   Contoh:

   #include <stdio.h>

   #include phi 3.14

3. Void

   artinya fungsi yang mengikutinya tidak memiliki nilai kembalian *(return)*.

4. Main ( )

   Fungsi main ( ) adalah fungsi yang pertama kali dijalankan ketika program dieksekusi tanpa fungsi main suatu program tidak dapat dieksekusi namun dapat dikompilasi.

5. Statement

   Statement adalah instruksi atau perintah kepada suatu program ketika program itu dieksekusi untuk menjalankan suatu aksi. Setiap statement diakhiri dengan titik-koma (;).

6. Tipe Data Dasar

Tabel 2.1 Tipe Data

Tipe Data	Ukuran (Byte)	Range	Keterangan
char	1	-128 s/d +127	Karakter/string
Int	2	-32768 s/d +32127	Integer/bilangan bulat
Float	4	3.4E-38 s/d 3.4E38	Float/pecahan
Long	4	-2.147.438.648 s/d +2.147.438.647	-
Double	8	1.7E-308 s/d 1.7E308	Pecahan presisi ganda
Long Double	10	3.4E-4932 s/d 3.4E4932	-

7. Operator

Tabel 2.2 Daftar Operator Aritmatika

Operator	Deskripsi
+	Penjumlahan (*add*)
-	Pengurangan (*substract*)
*	Perkalian (*multiply*)
/	Pembagian (*divide*)
%	Sisa pembagian integer (modulus)
~	Megasi (*negate*)

Tabel 2.3 Daftar Operator Kondisi

Operator	Keterangan
==	Sama dengan (bukan assignment)
!=	Tidak sama dengan
>	Lebih besar
<	Lebih kecil
>=	Lebih besar atau sama dengan
<=	Lebih kecil atau sama dengan

Tabel 2.4 Daftar Operator Logika

Opearator	Keterangan
&&	Logic AND
||	Logic OR
!	Logic NOT

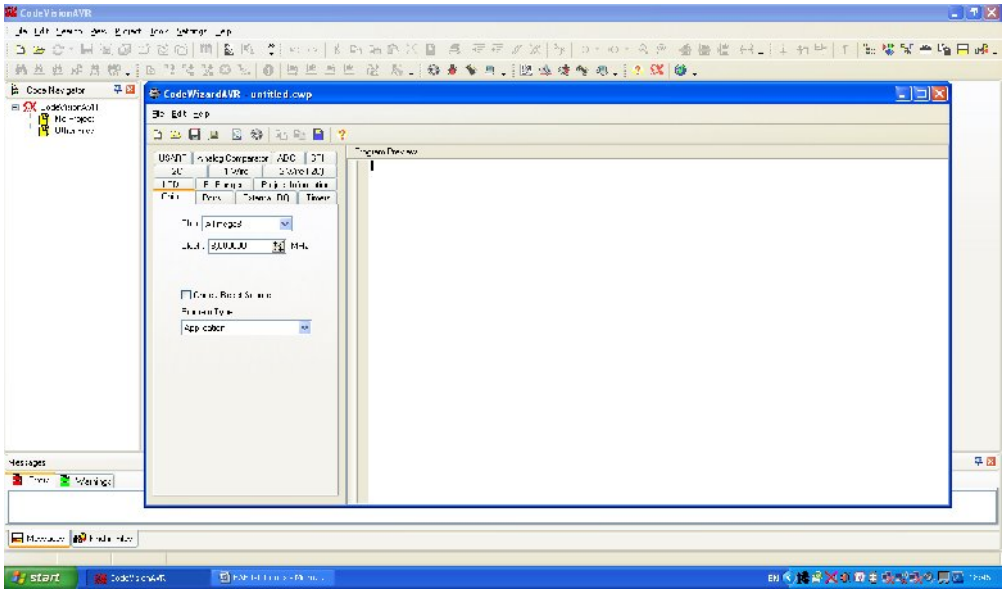
Tabel 2.5 Daftar Operator Bitwise

Operator	Keterangan
&	Bitwise AND
\| | Bitwise OR |
^ | Bitwise XOR |
~ | Bitwise NOT |

8. Pernyataan *if*

   Sebuah pernyataan yang dapat dipakai untuk mengambil keputusan berdasarkan suatu kondisi. Bentuk pernyataan ini ada dua macam:

   - *If* ,
   - *Else if*



Gambar 2.2 Tampilan CodeVisionAVR

CodeVision AVR merupakan *software* C-cross compiler, dimana program dapa ditulis menggunakan bahasa-C. Dengan menggunakan pemrograman bahasa-C diharapkan waktu disain (*developing time*) akan menjadi lebih singkat. Setelah program dalam bahasa-C ditulis dan dilakukan kompilasi tidak terdapat kesalahan (*error*) maka proses *download* dapat dilakukan. Mikrokontroler AVR mendukung

sistem *download* secara *In Sistem Programming* (ISP). Untuk selanjutnya fasilitas-fasilitas lainnya dapat di*setting* sesuai kebutuhan dari pemrograman. Setelah selesai dengan CodeWizard AVR, selanjutnya pada menu *File*, pilih *Generate*, *Save and Exit* dan simpan pada direktori yang diinginkan.

1.4 Visual Basic 6.0

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.

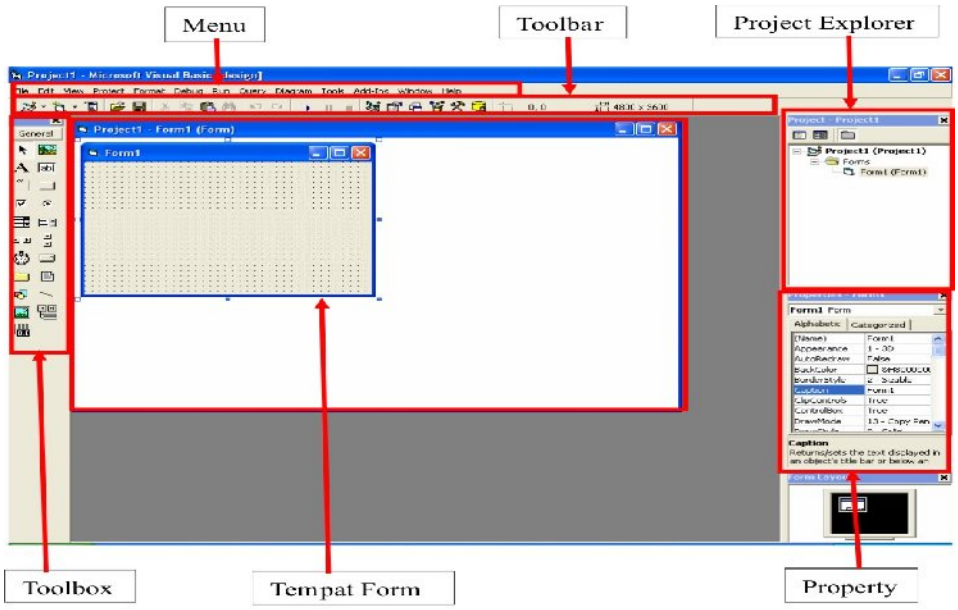
Ledakan pemakaian Visual Basic ditandai dengan kemampuan Visual Basic untuk dapat berinteraksi dengan aplikasi lain di dalam sistem operasi Windows dengan komponen *ActiveX Control.* Dengan komponen ini memungkinkan penguna untuk memanggil dan menggunakan semua model data yang ada di dalam sistem operasi 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.

2.4.1 Interface Visual Basic 6.0

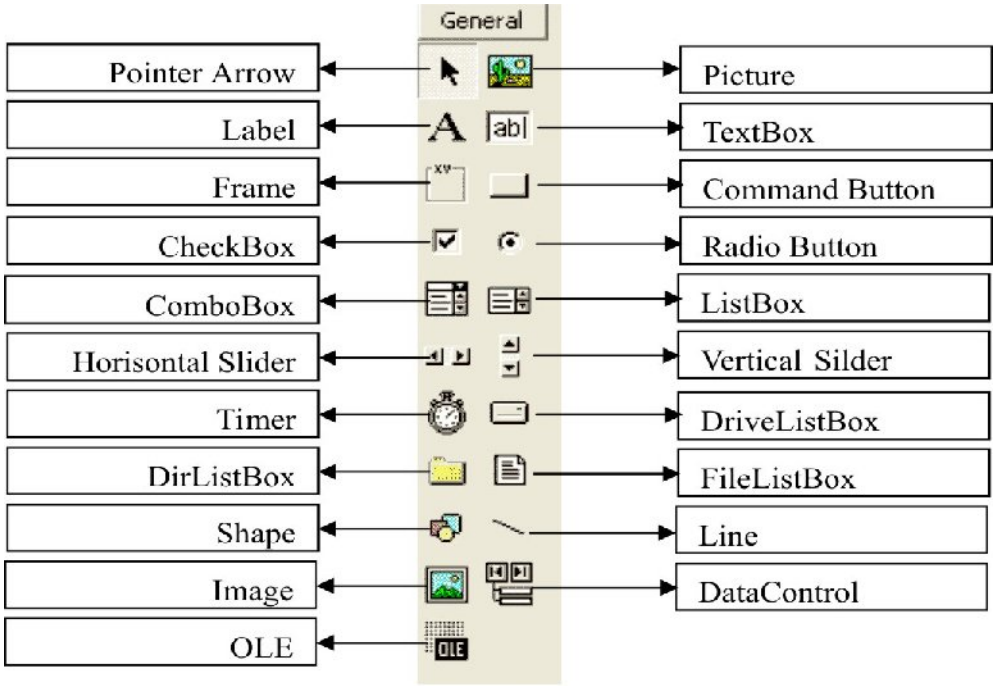
Interface Visual Basic 6.0, berisi menu, toolbar, toolbox, form, project explorer dan property seperti terlihat pada gambar 2.3. berikut:



Gambar 2.3. Interface Visual Basic 6.0

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]. 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.

Toolbox berisi komponen-komponen yang bisa digunakan oleh suatu project aktif, artinya isi komponen dalam toolbox sangat tergantung pada jenis project yang dibangun. Komponen standar dalam toolbox dapat dilihat pada gambar 2.4 berikut ini.



Gambar 2.4. Komponen Standar Dalam Toolbox

2.4.2 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 variable (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

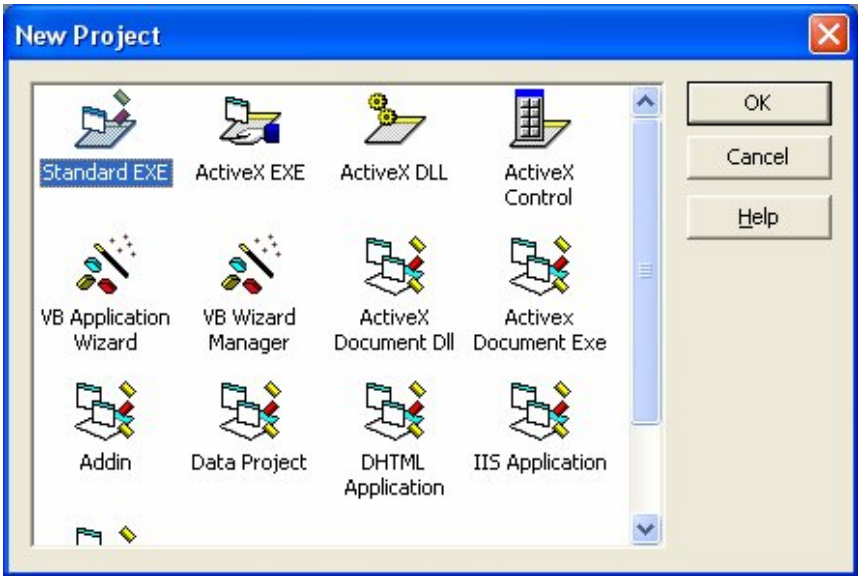
Timer1.Enable=True

(2) Metode: bahwa jalannya program dapat diatur sesuai aplikasi dengan menggunakan metode pemrograman yang diatur sebagai aksi dari setiap komponen. Metode inilah tempat untuk mengekpresikan 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.

2.4.3 Membuat Project Baru

Untuk memulai pembuatan program aplikasi di dalam Visual Basic, yang dilakukan adalah membuat project baru. Project adalah sekumpulan form, modul, fungsi, data dan laporan yang digunakan dalam suatu aplikasi. Membuat projrct baru dapat dilakukan dengan memilih menu [File] >> [New Project] atau dengan menekan ikon [new project] pada Toolbar yang terletak di pojok kiri atas. Setelah itu akan muncul konfirmasi untuk jenis project dari program aplikasi yang akan dibuat seperti terlihat pada gambar 2.5. berikut.



Gambar 2.5 Layar Pemilihan Jenis Project

Visual Basic 6.0 menyediakan 13 jenis project yang bisa dibuat seperti terlihat pada gambar 2.5 di atas. Ada beberapa project yang biasa digunakan oleh banyak pengguna Visual Basic, antara lain:

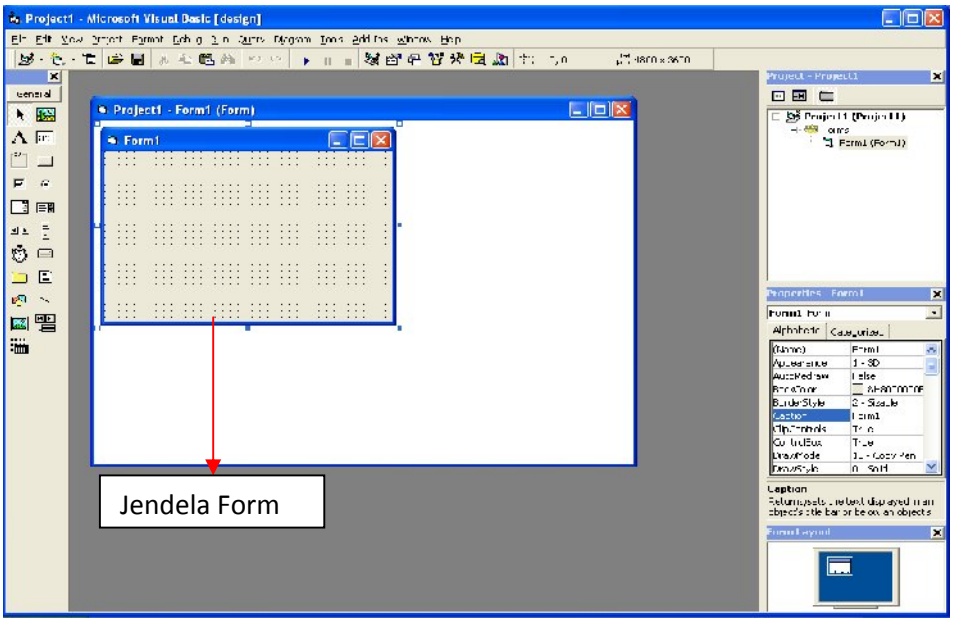
a. *Standard EXE*: Project standar dalam Visual Basic dengan komponen-komponen standar. Jenis project ini sangat sederhana, tetapi memiliki keunggulan bahwa semua komponennya dapat

diakui oleh semua unit komputer dan semua user meskipun bukan administrator. Pada buku ini akan digunakan project Standard EXE ini, sebagai konsep pemrograman visualnya.

b. *ActiveX EXE*: Project ini adalah *project ActiveX* berisi komponen-komponen kemampuan intuk berinteraksi dengan semua aplikasi di sistem operasi windows.
c. *ActiveX DLL*: Project ini menghasilkan sebuah aplikasi library yang selanjutnya dapat digunakan oleh semua aplikasi di sistem operasi windows.
d. *ActiveX Control*: Project ini menghasilkan komponen-komponen baru untuk aplikasi Visual Basic yang lain
e. *VB Application Wizard*: Project ini memandu pengguna untuk membuat aplikasi secara mudah tanpa harus pusing-pusing dengan perintah-perintah pemrograman.
f. *Addin*: Project seperti *Standard EXE* tetapi dengan berbagai macam komponen tambahan yang memungkinkan kebebasan kreasi dari pengguna.
g. *Data project*: Project ini melengkapi komponennya dengan komponen-komponen database. Sehingga bisa dikatakan project ini memang disediakan untuk keperluan pembuatan aplikasi database.
h. *DHTML Application*: Project ini digunakan untuk membuat aplikasi internet pada sisi client (*client side*) dengan fungsi-fungsi DHTML.

i. *IIS Application*: Project ini menghasilkan apliaksi internet pada sisi server (*server side*) dengan komponen-komponen CGI (*Common Gateway Interface*).

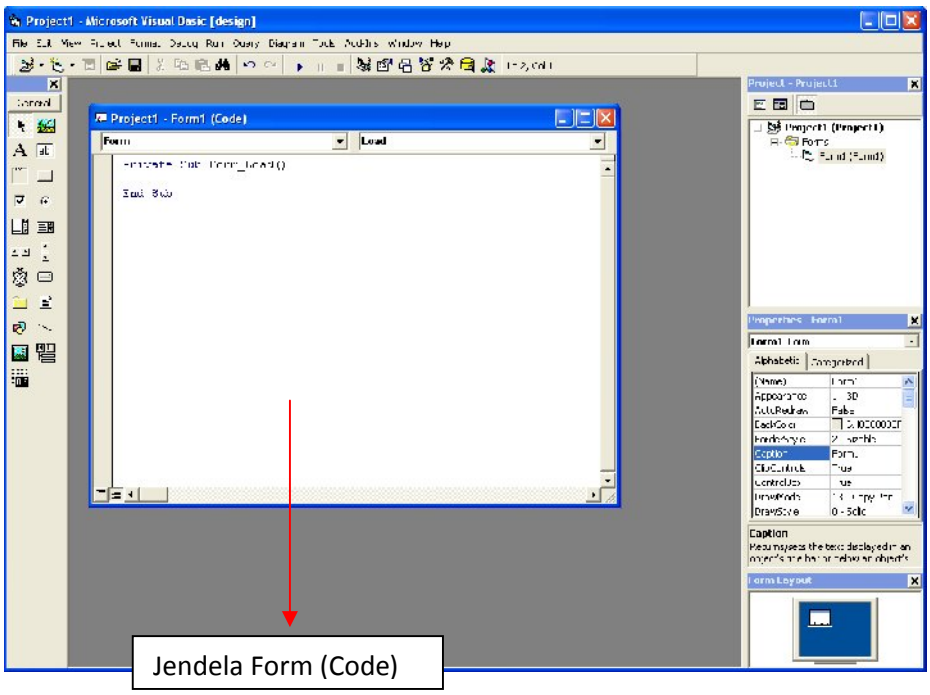
Selanjutnya pilih *Standard EXE* dan tekan [Ok]. Lalu muncul tampilan dari *Standard Exe* seperti pada gambar 2.2. Dengan demikian project sudah siap dibuat. Dalam pembuatan project sebelumnya *double click* pada form yang terbuat maka akan terlihat jendela tersembunyi (*hidden windows*) yang berupa jendela untuk pembuatan program atau jendela kode (*code windows*). Hal ini Dapat dilakukan dengan cara memilih ikon jendela form atau jendela kode yang ada di [Project Explorer]. Hal ini dapat dilihat pada gambar 2.6 dan gambar 2.7



Gambar 2.6. Jendela Form

Pada jendela form, digunakan dalam membangun tampilan dari program aplikasi yang akan dibuat dengan mengatur komponen-komponen baik letak, properti dan eventnya. Untuk mengambil suatu komponen dari [Toolbox] dapat

dilakukan dengan klik komponen tersebut, kemudian klik atau tarik pada posisi yang benar pada form.



Gambar 2.7. Jendela Kode

Sedangkan, pada jendela kode, digunakan untuk menuliskan program dari komponen-komponen yang sudah deletakkan pada jendela form sesuai dengan aplikasi yang akan dibuat.

2.5 Mikrokontroler AVR ATmega8535

Mikrokontroler adalah IC yang dapat diprogram berulang kali, baik ditulis atau dihapus. Biasanya digunakan untuk pengontrolan otomatis dan manual pada perangkat elektronika. Beberapa tahun terakhir, mikrokontroler sangat banyak digunakan terutama dalam pengontrolan robot. Seiring perkembangan elektronika, mikrokontroler dibuat semakin kompak dengan bahasa pemrograman yang juga ikut berubah. Salah satunya adalah mikrokontroler AVR (*Alf and Vegard's Risc*

processor) ATmega8535 yang menggunakan teknologi RISC (*Reduce Instruction Set Computing*) dimana program berjalan lebih cepat karena hanya membutuhkan satu *siklus clock* untuk mengeksekusi satu instruksi program. Secara umum, AVR dapat dikelompokkan menjadi 4 kelas, yaitu kelas ATtiny, keluarga AT90Sxx, keluarga ATmega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, mereka bisa dikatakan hampir sama.

Mikrokontroler AVR ATmega8535 memiliki fitur yang cukup lengkap. Mikrokontroler AVR ATmega8535 telah dilengkapi dengan ADC *internal*, *EEPROM internal*, *Timer/Counter, PWM, analog comparator,* dan lain-lain. Sehingga dengan fasilitas yang lengkap ini memungkinkan kita belajar mikrokontroler keluarga AVR dengan lebih mudah dan efisien, serta dapat mengembangkan kreativitas penggunaan mikrokontroler ATmega8535.

Fitur-fitur yang dimiliki oleh mikrokontroler ATmega8535 adalah sebagai berikut:

1. Saluran I/O sebanyak 32 buah, yaitu port A, port B, port C, dan port D.
2. ADC internal sebanyak 8 saluran.
3. Tiga buah Timer/Counter dengan kemampuan pembandingan.
4. CPU yang terdiri atas 32 buah register.
5. SRAM sebesar 512 byte.
6. Memori Flash sebesar 8 kb dengan kemampuan Read While Write.
7. Port antarmuka SPI
8. EEPROM sebesar 512 byte yang dapat diprogram saat operasi.
9. Antarmuka komparator analog.

10. Port USART untuk komunikasi serial.

11. Sistem mikroprosesor 8 bit berbasis RISC dengan kecepatan maksimal 16 MHz.

12. Dan lain-lainnya.

2.5.1 Konstruksi ATmega8535

Mikrokontroler ATmega8535 memiliki 3 jenis memori, yaitu memori program, memori data dan memori EEPROM. Ketiganya memiliki ruang sendiri dan terpisah.

a. Memori program

ATmega8535 memiliki kapasitas memori progam sebesar 8 Kbyte yang terpetakan dari alamat 0000h – 0FFFh dimana masing-masing alamat memiliki lebar data 16 bit. Memori program ini terbagi menjadi 2 bagian yaitu bagian program *boot* dan bagian program aplikasi.

b. Memori data

ATmega8535 memiliki kapasitas memori data sebesar 608 byte yang terbagi menjadi 3 bagian yaitu register serba guna, register I/O dan SRAM. ATmega8535 memiliki 32 byte register serba guna, 64 byte register I/O yang dapat diakses sebagai bagian dari memori RAM (menggunakan instuksi LD atau ST) atau dapat juga diakses sebagai I/O (menggunakan instruksi IN atau OUT), dan 512 byte digunakan untuk memori data SRAM.

c. Memori EEPROM

ATmega8535 memiliki memori EEPROM sebesar 512 byte yang terpisah dari memori program maupun memori data. Memori EEPROM ini hanya dapat

diakses dengan menggunakan register-register I/O yaitu register EEPROM *Address*, register EEPROM *Data*, dan register EEPROM *Control*. Untuk mengakses memori EEPROM ini diperlakukan seperti mengakses data eksternal, sehingga waktu eksekusinya relatif lebih lama bila dibandingkan dengan mengakses data dari SRAM.

ATmega8535 merupakan tipe AVR yang telah dilengkapi dengan 8 saluran ADC internal dengan fidelitas 10 bit. Dalam mode operasinya, ADC ATmega8535 dapat dikonfigurasi, baik secara *single ended input* maupun *differential input*. Selain itu, ADC ATmega8535 memiliki konfigurasi pewaktuan, tegangan referensi, mode operasi, dan kemampuan filter derau yang amat fleksibel, sehingga dengan mudah disesuaikan dengan kebutuhan ADC itu sendiri.

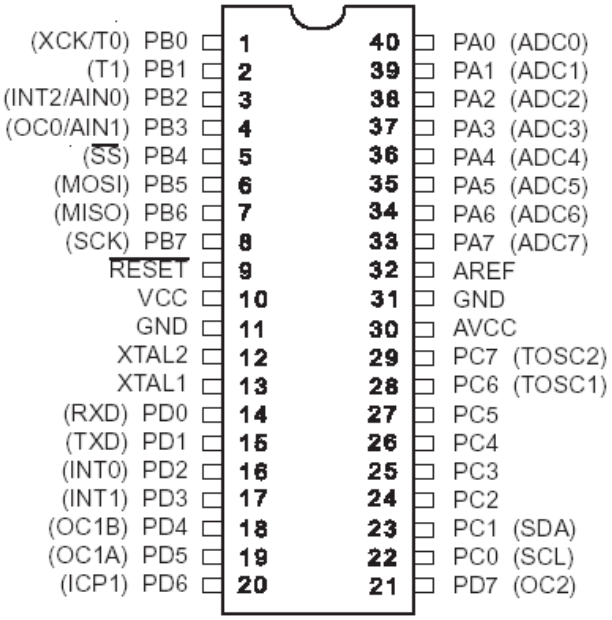
ATmega8535 memiliki 3 modul timer yang terdiri dari 2 buah timer/counter 8 bit dan 1 buah timer/counter 16 bit. Ketiga modul timer/counter ini dapat diatur dalam mode yang berbeda secara individu dan tidak saling mempengaruhi satu sama lain. Selain itu, semua timer/counter juga dapat difungsikan sebagai sumber interupsi. Masing-masing timer/counter ini memiliki register tertentu yang digunakan untuk mengatur mode dan cara kerjanya.

Serial Peripheral Interface (SPI) merupakan salah satu mode komunikasi serial *syncrhronous* kecepatan tinggi yang dimiliki oleh ATmega8535. *Universal Syncrhronous and Asyncrhronous Serial Receiver and Transmitter* (USART) juga merupakan salah satu mode komunikasi serial yang dimiliki oleh ATmega8535. USART merupakan komunikasi yang memiliki fleksibilitas tinggi, yang dapat digunakan untuk melakukan transfer data baik antar mikrokontroler maupun dengan modul-modul eksternal termasuk PC yang memiliki fitur UART.

USART memungkinkan transmisi data baik secara *syncrhronous* maupun *asyncrhronous,* sehingga dengan memiliki USART pasti kompatibel dengan UART. Pada ATmega8535, secara umum pengaturan mode *syncrhronous* maupun *asyncrhronous* adalah sama. Perbedaannya hanyalah terletak pada sumber clock saja.

Jika pada mode *asyncrhronous* masing-masing peripheral memiliki sumber clock sendiri, maka pada mode *syncrhronous* hanya ada satu sumber clock yang digunakan secara bersama-sama. Dengan demikian, secara hardware untuk mode *asyncrhronous* hanya membutuhkan 2 pin yaitu TXD dan RXD, sedangkan untuk mode *syncrhronous* harus 3 pin yaitu TXD, RXD dan XCK.

2.5.2 Pin-pin Pada Mikrokontroler ATmega8535



Gambar 2.8 Konfigurasi Pin ATmega8535

Konfigurasi *pin* ATmega8535 dengan kemasan 40 pin DIP (*Dual Inline Package*) dapat dilihat pada gambar 2.8. Dari gambar di atas dapat dijelaskan fungsi dari masing-masing *pin* Atmega8535 sebagai berikut:

1. VCC merupakan *pin* yang berfungsi sebagai masukan catu daya.
2. GND merukan *pin Ground*.
3. *Port* A (PortA0…PortA7) merupakan *pin input/output* dua arah dan *pin* masukan ADC.
4. *Port* B (PortB0…PortB7) merupakan *pin input/output* dua arah dan dan *pin* fungsi khusus
5. *Port* C (PortC0…PortC7) merupakan *pin input/output* dua arah dan *pin* fungsi khusus
6. *Port* D (PortD0…PortD7) merupakan *pin input/output* dua arah dan *pin* fungsi khusus
7. RESET merupakan *pin* yang digunakan untuk me-*reset* mikrokontroler.
8. XTAL1 dan XTAL2 merupakan *pin* masukan *clock* eksternal.
9. AVCC merupakan *pin* masukan tegangan untuk ADC.
10. AREFF merupakan *pin* masukan tegangan referensi ADC.

2.6 Komunikasi Data Serial

Standart RS232 ditetapkan oleh Electronic Industry Association dan Telecomunication Industry Association pada tahun 1962. Nama lengkapnya adalah EIA/TIA-232 Interface Between Data Terminal Equipment and Data Terminal equipment Employing Serian Binary Data Interchage. Dengan demikian standard ini hanya menyangkut komunikasi data antara komputer (Data Terminal

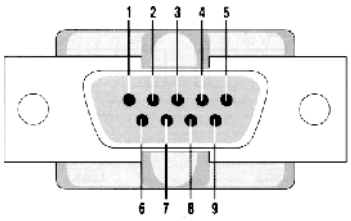
Equipment - DTE) dengan alat-alat pelengkap komputer (Data Circuit Terminal Equipment - DCE). Dalam banyak literatur, DCE sering diartikan sebagai Data Communication Equipment, hal ini bisa dibenarkan tetapi pengertiannya menjadi lebih sempit karena sebagai Data Communication Equipment yang dimaksud dengan DTE hanya sebatas peralatan untuk komunikasi, misalnya modem. Padahal yang dimaksud dengan Data Circuit Terminal Equipment bisa meliputi macam-macam alat pelengkap komputer yang dihubungkan ke komputer dengan standard RS232, misalnya Printer, Optical Mark Reader, Card Register dan alat-alat lainnya yang bisa dihubungkan ke komputer.

Pada prinsipnya, serial ialah pengiriman data dilakukan per bit, sehingga lebih lambat dibandingkan parallel seperti pada port printer yang mampu mengirim 8 bit sekaligus dalam sekali detak. Beberapa contoh serial ialah mouse, scanner dan sistem akuisisi data yang terhubung ke port COM1/COM2.

Device pada serial port dibagi menjadi 2 (dua) kelompok yaitu Data Communication Equipment (DCE) dan Data Terminal Equipment (DTE). Contoh dari DCE ialah modem, plotter, scanner dan lain lain sedangkan contoh dari DTE ialah terminal di komputer.Spesifikasi elektronik dari serial port merujuk pada Electronic Industry Association (EIA) :

1. “Space” (logika 0) ialah tegangan antara + 3 hingga +25 V.
2. “Mark” (logika 1) ialah tegangan antara –3 hingga –25 V.
3. Daerah antara + 3V hingga –3V tidak didefinisikan /tidak terpakai
4. Tegangan open circuit tidak boleh melebihi 25 V.
5. Arus hubungan singkat tidak boleh melebihi 500mA.

Komunikasi serial membutuhkan port sebagai saluran data. Berikut tampilan port serial DB9 yang umum digunakan sebagai port serial.



Gambar 2.9 Port DB9

Keterangan

· Pin 1 = Data Carrier Detect (DCD)

· Pin 2 = Received Data (RxD)

· Pin 3 = Transmitted Data (TxD)

· Pin 4 = Data Terminal Ready (DTR)

· Pin 5 = Signal Ground (common)

· Pin 6 = Data Set Ready (DSR)

· Pin 7 = Request To Send (RTS)

· Pin 8 = Clear To Send (CTS)

· Pin 9 = Ring Indicator (RI)

Pada UART (Universal Asynchronous Receiver Transmitter), kecepatan pengiriman data (atau yang sering disebut dengan Baud Rate) dan fase clock pada sisi transmitter dan sisi receiver harus sinkron. Untuk itu diperlukan sinkronisasi antara Transmitter dan Receiver. Hal ini dilakukan oleh bit “Start” dan bit “Stop”. Ketika saluran transmisi dalam keadaan idle, output UART adalah dalam keadaan logika “1”.

Ketika Transmitter ingin mengirimkan data, output UART akan diset dulu ke logika “0” untuk waktu satu bit. Sinyal ini pada receiver akan dikenali sebagai sinyal “Start” yang digunakan untuk menyinkronkan fase clocknya sehingga sinkron dengan fase clock transmitter. Selanjutnya data akan dikirimkan secara serial dari bit yang paling rendah (bit0) sampai bit tertinggi. Selanjutnya akan dikirimkan sinyal “Stop” sebagai akhir dari pengiriman data serial. Untuk dapat menggunakan port serial harus diketahui dahulu alamat dari port serial tersebut. Biasanya tersedia dua port serial pada CPU, yaitu COM1 dan COM2. Base Address COM1 biasanya 1016 (3F8h) dan COM2 biasanya 760 (2F8h). Alamat tersebut adalah alamat yang biasa digunakan, tergantung komputer yang digunakan.Tepatnya kita bisa melihat pada peta memori tempat menyimpan alamat tersebut, yaitu memori 0000.0400h untuk COM1 dan 0000.0402h untuk COM2. Berikut adalah nama-nama register yang digunakan beserta alamatnya.

Tabel 2.6 Nama Register Dan Alamat Register

Nama Register	COM1	COM2
TX Buffer	3F8h	2F8h
RX Buffer	3F8h	2F8h
Baud Rate Dicisor latch LSB	3F8h	2F8h
Baud Rate Dicisor latch MSB	3F9h	2F9h
Interrupt Enable Register	3F9h	2F9h
Interrupt Identification Register	3FAh	2FAh
Line Control Register	3FBh	2FBh
Modem Control Register	3FCh	2FCh
Line Status Register	3FDh	2FDh
Modem Sttus Register	3FEh	2FEh

Keterangan Register :

- *RX Buffer* , digunakan untuk menampung dan menyimpan data dari DCE.
- *TX Buffer* , digunakan untuk menampung dan menyimpan data yang akan dikirim ke port serial.
- *Baud Rate Divisor Latch LSB* , digunakan untuk menampung byte bobot rendah untuk pembagi clock pada IC UART agar didapat baud rate yang tepat.
- *Baud Rate Divisor Latch MSB* , digunakan untuk menampung byte bobot tinggi untuk pembagi clock pada IC UART sehingga total angka pembagi adalah 4 byte yang dapat dipilih dari 0001h sampai FFFFh.

Berikut adalah tabel angka pembagi yang sering digunakan :

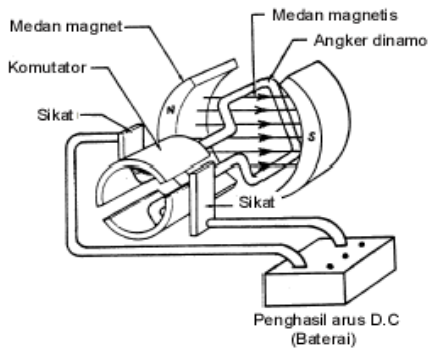
Tabel 2.7 Tabel Baud Rate

Baud Rate (bit/detik)	Angka Pembagi
300	0180h
600	0C00h
1200	0060h
1800	0040h
2400	0030h
4800	0018h
9600	000ch

2.7 Motor DC

Motor listrik merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Motor DC memerlukan suplai tegangan yang

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



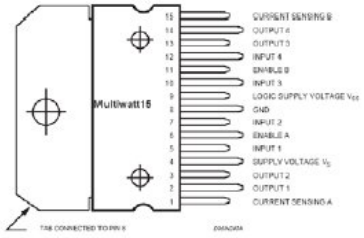
Gambar 2.10 Mekanik Motor DC

Mekanisme kerja untuk seluruh jenis motor secara umum :

- Arus listrik dalam medan magnet akan memberikan gaya.
- Jika kawat yang membawa arus dibengkokkan menjadi sebuah lingkaran / *loop*, maka kedua sisi *loop*, yaitu pada sudut kanan medan magnet, akan mendapatkan gaya pada arah yang berlawanan.
- Pasangan gaya menghasilkan tenaga putar / *torque* untuk memutar kumparan.

- Motor-motor memiliki beberapa *loop* pada dinamonya untuk memberikan tenaga putaran yang lebih seragam dan medan magnetnya dihasilkan oleh susunan elektromagnetik yang disebut kumparan medan.

IC L298 digunakan sebagai rangkaian *driver*. Cukup dihubungkan ke mikrokontroler dan diberi tegangan sebesar 5 volt dengan arus minimal 2 ampere rangkaian *driver* berbasis L298 sudah dapat digunakan. Selain itu, *supply* IC L298 dapat diberi tegangan sampai 50 Volt.



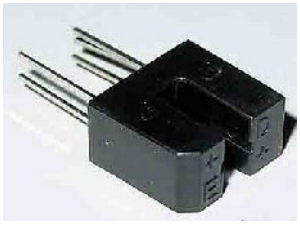
Gambar 2.11 Konfigurasi Pin IC L298

Untuk menjalankan motor, pin *enable* A dan *enable* B pada IC L298 harus diberi logika 1. Current sensing A dan current sensing B dihubungkan ke *ground*. Input 1 dan input 2 masing-masing berlogika 1 dan 0, output 1 dan output 2 dihubungkan ke motor.

2.8 Optocoupler

Optocoupler merupakan salah satu jenis komponen yang memanfaatkan sinar sebagai pemicu on/off-nya. Opto berarti optic dan coupler berarti pemicu. Sehingga bisa diartikan bahwa optocoupler merupakan suatu komponen yang bekerja berdasarkan picu cahaya optic opto-coupler termasuk dalam sensor,

dimana terdiri dari dua bagian yaitu transmitter dan receiver. Dasar rangkaian dapat ditunjukkan seperti pada gambar dibawah ini:



Gambar 2.12 Optocoupler

Bagian pemancar atau transmitter dibangun dari sebuah led infra merah untuk mendapatkan ketahanan yang lebih baik daripada menggunakan led biasa. Sensor ini bisa digunakan sebagai isolator dari rangkaian tegangan rendah kerangkaian tegangan tinggi. Selain itu juga bisa dipakai sebagai pendeteksi adanya penghalang antara transmitter dan receiver dengan memberi ruang uji dibagian tengah antara led dengan photo transistor.