BAB II

DASAR TEORI

2.1 Pengenalan Mikrokontroler

Mikrokontroler adalah suatu alat elektronika digital yang berupa keping IC dan merupakan suatu terobosan teknologi mikroprosesor dan mikrokomputer terbaru yang hadir memenuhi kebutuhan pasar (*market needed*). Mikrokontroler mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data.

Sebagai teknologi terbaru dengan teknologi semikonduktor yang mengandung transistor yang lebih banyak namun hanya membutuhkan ruang kecil sebagai wadah penempatannya dan dapat diproduksi secara massal sehingga harganya lebih murah dan dapat terjangkau oleh hampir seluruh kalangan masyarakat. Oleh karena itu mikrokontroler sangat cocok diterapkan untuk mengontrol berbagai peralatan-peralatan yang lebih canggih dibandingkan dengan komputer PC, karena effektivitas dan kefleksibelannya yang tinggi.

Kelebihan utama dari mikrokontroler ialah telah tersedianya RAM dan peralatan I/O pendukung sehingga ukuran board mikrokontroler menjadi sangat ringkas. Terdapat berbagai jenis mikrokontroler dari berbagai *vendor* yang digunakan secara luas di dunia.

Diantaranya yang terkenal ialah dari Intel, Maxim, Motorolla , dan ATMEL. Beberapa seri mikrokontroler yang digunakan secara luas ialah 8031, 68HC11, 6502 , 2051, 89S51 dan 89S52.

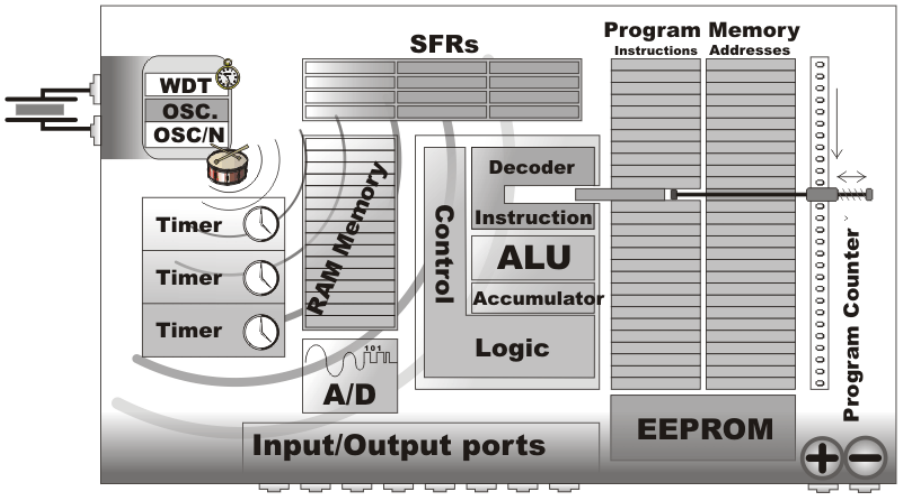
2.1.1 Keuntungan Mikrokontroler

Beberapa keuntungan mikrokontroler dalam sistem kendali, antara lain :

- Sistem elektronik akan menjadi lebih ringkas.
- Rancang bangun sistem elektronik akan lebih cepat karena sebagian besar dari sistem adalah perangkat lunak yang mudah dimodifikasi.
- Pencarian gangguan lebih mudah ditelusuri karena sistemnya yang kompak.

2.1.2 Arsitektur Mikrokontroler

Mikrokontroler adalah mikroprosesor yang telah dilengkapi komponen-komponen pendukung secara internal yaitu seperti: CPU, ROM, RAM, dan I/O yang membentuk mikrokomputer tunggal yang dapat disebut juga sebagai versi mini dan untuk aplikasi khusus dari mikrokomputer atau komputer dikemas dalam bentuk IC.



Gambar 2.1 Arsitektur Mikrokontroler

ALU, Instruction Decoder, Accumulator dan Control merupakan **Otak**-nya mikrokontroler yang bersangkutan. Jantungnya berasal dari detak OSC. Sedangkan di sekeliling 'Otak' terdapat berbagai macam periferal seperti SFR (*Special Function Register* yang bertugas menyimpan data-data sementara selama proses berlangsung), Memori RAM (tugas hampir sama seperti SFR hanya saja tidak berhubungan langsung selama proses operasional mikrokontroler), ADC (untuk mengubah data-data analog menjadi digital untuk diolah atau diproses lebih lanjut), EEPROM (sama seperti RAM hanya saja tetap akan menyimpan data walaupun tidak mendapatkan sumber listrik/daya) dan port-port I/O untuk masukan/luaran, untuk melakukan komunikasi dengan pihak-pihak eksternal mikrokontroler (sensor dan aktuator).

Ciri khas mikrokontroler lainnya, antara lain:

- 'Tertanam' (atau *embedded*) dalam beberapa piranti (umumnya merupakan produk konsumen) atau yang dikenal dengan istilah *embedded system* atau *embedded controller*.

- Hanya membutuhkan daya yang rendah (*low power*) sekitar 50 mWatt sedangkan komputer membutuhkan daya yang bisa mencapai 50 Watt lebih.
- Memiliki beberapa keluaran maupun masukan yang terdedikasi, untuk tujuan atau fungsi-fungsi khusus.
- Kecil dan relatif lebih murah.
- Seringkali tahan-banting, terutama untuk aplikasi-aplikasi yang berhubungan dengan mesin atau otomotif atau militer.

2.1.3 Bahasa Pemrograman Mikrokontroler

Setiap mikrokontroler memiliki bahasa-bahasa pemrograman yang berbeda. Terdapat beberapa bahasa pemrograman mikrokontroller yang digunakan untuk membuat perangkat lunak program mikrokontroller, tetapi pada umumnya bahasa yang sering digunakan tergantung dari kemampuan orang – orang yang ingin menggunakannya. Adapun beberapa jenis bahasa pemrograman mikrokontroller yang banyak digunakan dalam pembuatan program kendali adalah bahasa Assembler, BASIC, Pascal, dan bahasa C.

2.1.4 Mikrokontroller AT89S52

Sebuah mikroprosesor yang digabungkan dengan input-output (I/O) dan memori (*Random Access Memory/Read Only Memory*) akan membentuk sebuah sistem mikrokomputer.

Dari pemikiran CPU yang dapat dikonstruksi dalam sebuah IC tunggal, maka sebuah mikroprosesor, I/O dan memori dapat pula dibangun dalam tingkatan IC. Konstruksi ini menghasilkan *Single Chip Microcomputer* (SCM). SCM inilah yang disebut mikrokontroler.

AT89S52 adalah salah satu anggota dari keluarga MCS-51/52 yang dilengkapi dengan internal 8 Kbyte *Flash* PEROM (*Programmable and Erasable Read Only Memory*), yang memungkinkan memori program untuk dapat diprogram kembali.

2.1.4.1 Arsitektur Mikrokontroller AT89S52

Mikrokontroler AT89S52 memiliki :

1. Memiliki CPU(*Central Processing Unit*) 8 bit.
2. Oscilator internal dengan rangkaian tambahan untuk pewaktu.
3. RAM internal 256 byte.
4. *Flash Memory* untuk menyimpan program sebesar 8 Kbyte.
5. *In-System Programable(ISP) Flash Memory.*
6. Memiliki 32 buah jalur I/O yang *programable*.
7. Memiliki 3 buah Timer 16 bit.
8. Mendukung 6 buah sumber *interrupt*.
9. Memiliki kanal serial UART yang *programable*.

2.1.4.2 CPU (*Central Processing Unit*)

Bagian ini berfungsi mengendalikan seluruh operasi pada mikrokontroler. Unit ini terbagi atas dua bagian, yaitu unit pengendali atau CU (*Control Unit*) dan unit aritmatika dan logika atau ALU (*Aritmetic logic Unit*).

Fungsi utama unit pengendali adalah mengambil instruksi dari memori (*fetch*) kemudian menterjemahkan susunan instruksi tersebut menjadi kumpulan proses kerja sederhana (*decode*), dan melaksanakan urutan instruksi sesuai dengan langkah-langkah yang telah ditentukan program (*execute*).

Unit aritmatika dan logika merupakan bagian yang berurusan dengan operasi aritmatika seperti penjumlahan, pengurangan, serta manipulasi data secara logika seperti operasi AND, OR, dan perbandingan.

2.1.4.3 Bagian Masukan/Keluaran (I/O)

Bagian ini berfungsi sebagai alat komunikasi serpih tunggal dengan piranti di luar sistem. Sesuai dengan namanya, perangkat I/O dapat menerima maupun memberi data dari/ke serpih tunggal.

Ada dua macam piranti I/O yang digunakan, yaitu piranti untuk hubungan serial UART (*Universal Asynchronous Receiver Transmitter*) dan piranti untuk hubungan pararel yang disebut dengan PIO (*Pararel Input Output*). Kedua jenis I/O tersebut telah tersedia di dalam serpih tunggal AT89S52.

2.1.4.4 Perangkat Lunak

Serpih tunggal keluarga MCS-51 memiliki bahasa pemrograman khusus yang tidak dipahami oleh jenis serpih tunggal yang lain. Bahasa pemrograman ini dikenal dengan nama bahasa assembler yang memiliki 256 perangkat instruksi.

Berikut adalah contoh *listing* program mikrokontroler :

```
$MOD51     ; instructions for assembler
           ; variables

           ARED   equ    P1.0
           AYEL   equ    P1.1
           AGRE   equ    P1.2
           BRED   equ    P1.3
           BYEL   equ    P1.4
           BGRE   equ    P1.5

           ORG    0000H
           LJMP   BEGGIN
           ORG    0030H

BEGGIN:
           LCALL  CLEARALL
           SJMP   MAIN

MAIN:
           LCALL  SIT1
           LCALL  SIT3
           LCALL  SIT2
           LCALL  SIT4
           SJMP   MAIN

; empat situasi

SIT1:
           SETB   ARED
           SETB   BGRE
           MOV    R0, #0AH     ; 10 detik
           LCALL  WAIT
           LCALL  CLEARALL
           RET

SIT2:
           SETB   AGRE
```

```
            SETB   BRED
            MOV    R0, #0AH     ; 10 detik
            LCALL  WAIT
            LCALL  CLEARALL
            RET

SIT3:
            SETB   ARED
            SETB   BYEL
            MOV    R0, #02H     ; 2 detik
            LCALL  WAIT
            LCALL  CLEARALL
            RET

SIT4:
            SETB   AYEL
            SETB   BRED
            MOV    R0, #02H     ; 2 detik
            LCALL  WAIT
            LCALL  CLEARALL
            RET

; counters

WAIT:
            LCALL  ONESEC       ; menentukan 1 detik
            DJNZ   R0, WAIT     ; ulangi waktu R0
            RET

ONESEC:
            MOV    R1, #14H     ; menentukan 1 detik

CLEARALL:
            MOV    P1, #00H     ; hapus P1
            CLR    A            ; hapus ACC
            RET

End
```

Perangkat lunak pada mikrokontroler dapat dibagi menjadi lima kelompok sebagai berikut :

1. Instruksi Transfer Data

   Instruksi ini berfungsi memindahkan data, yaitu antar *register*, dari memori ke memori, dari *register* ke memori dan lain lain.

2. Instruksi Aritmatika

Instruksi ini melaksanakan operasi aritmatika yang meliputi penjumlahan, pengurangan, penambahan satu (*increment*), pengurangan satu (*decrement*), perkalian dan pembagian.

3. Instruksi Logika dan Manipulasi Bit

Berfungsi melaksanakan operasi logika AND, OR, XOR, perbandingan, penggeseran dan komplemen data.

4. Instruksi Percabangan

Berfungsi untuk mengubah urutan normal pelaksanaan suatu program. Dengan instruksi ini, program yang sedang dilaksanakan akan meloncat ke suatu alamat tertentu.

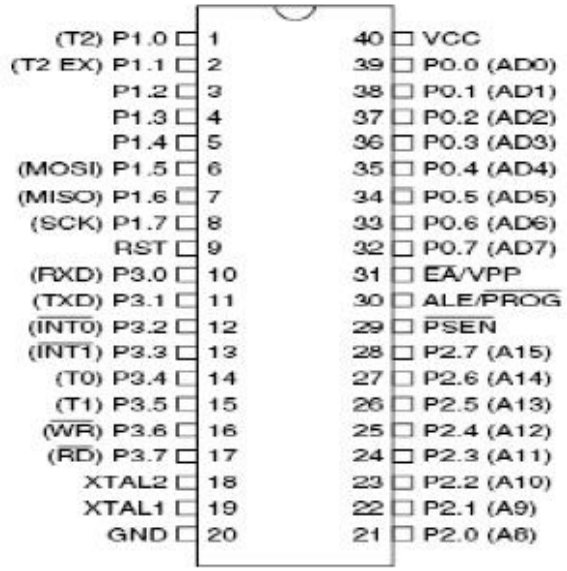
5. Instruksi Stack, I/O, dan Kontrol

Instruksi ini mengatur penggunaan stack, membaca/menulis port I/O, serta pengontrolan.

Pada penelitian ini, AT89S52 berfungsi untuk menampung sinyal hasil ADC, selanjutnya mengirimkan sinyal tersebut ke port serial.

2.1.4.5 Konfigurasi Pin

Mikrokontroler AT89S52 mempunyai 40 pin dengan catu daya tunggal 5 Volt. Ke - 40 pin tersebut digambarkan sebagai berikut :



Gambar 2.2 Konfigurasi Pin AT89S52

Fungsi dari masing-masing pin AT89S52 adalah :

1. Pin 1 sampai 8 (Port 1) merupakan port pararel 8 bit dua arah (*bidirectional*) yang dapat digunakan untuk berbagai keperluan (*general purpose*).

2. Pin 9 merupakan pin reset, reset aktif jika mendapat catuan tinggi.

3. Pin 10 sampai 17 (Port 3) adalah port pararel 8 bit dua arah yang memiliki fungsi pengganti sebagai berikut :

   - P3.0 (10) : RXD (port serial penerima data)
   - P3.1 (11) : TXD (port serial pengirim data)
   - P3.2 (12) : INT0 (input interupsi eksternal 0, aktif *low*)
   - P3.3 (13) : INT1 (input interupsi ekstrernal 1, aktif *low*)
   - P3.4 (14) : T0 (eksternal input *timer / counter* 0)

- P3.5 (15) : T1 (eksternal input *timer / counter* 1)
- P3.6 (16) : WR (*Write*, aktif *low*) Sinyal kontrol penulisan data dari port 0 ke memori data dan input-output eksternal.
- P3.7 (17) : RD (*Read*, aktif *low*) Sinyal kontrol pembacaan memori data input-output eksternal ke port 0.

4. Pin 18 sebagai XTAL 2, keluaran osilator yang terhubung pada kristal.
5. Pin 19 sebagai XTAL 1, masukan ke osilator berpenguatan tinggi, terhubung pada kristal.
6. Pin 20 sebagai Vss, terhubung ke 0 atau ground pada rangkaian.
7. Pin 21 sampai 28 (Port 2) adalah port pararel 8 bit dua arah. Port ini mengirim byte alamat bila pengaksesan dilakukan pada memori eksternal.
8. Pin 29 sebagai PSEN (*Program Store Enable*) adalah sinyal yang digunakan untuk membaca, memindahkan program memori eksternal (ROM / EPROM) ke mikrokontroler (aktif *low*).
9. Pin 30 sebagai ALE (*Address Latch Enable*) untuk menahan alamat bawah selama mengakses memori eksternal. Pin ini juga berfungsi sebagai PROG (aktif *low*) yang diaktifkan saat memprogram internal *flash* memori pada mikrokontroler (*on chip*).

10. Pin 31 sebagai EA (*External Accesss*) untuk memilih memori yang akan digunakan, memori program internal (EA = Vcc) atau memori program eksternal (EA = Vss), juga berfungsi sebagai Vpp (*programming supply voltage*) pada saat memprogram internal *flash* memori pada mikrokontroler.

11. Pin 32 sampai 39 (Port 0) merupakan port pararel 8 bit dua arah. Berfungsi sebagai alamat bawah yang dimultipleks dengan data untuk mengakses program dan data memori eksternal.

12. Pin 40 sebagai Vcc, terhubung ke +5 V sebagai catuan untuk mikrokontroler.

2.1.4.6 Organisasi Memori

Semua serpih tunggal dalam keluarga MCS-51 memiliki pembagian ruang alamat untuk program dan data. Pemisahan memori program dan memori data memperbolehkan memori data untuk diakses oleh alamat 8 bit. Sekalipun demikian, alamat data memori 16 bit dapat dihasilkan melalui register DPTR (*Data Point Register*). Memori program hanya bisa dibaca tidak bisa ditulis karena disimpan dalam EPROM. Dalam hal ini EPROM yang tersedia di dalam serpih tunggal AT89S52 sebesar 8 Kbyte.

a. **Memori Program**

Pada EPROM 8 Kbyte, jika EA (*External Access*) bernilai tinggi, maka program akan menempati alamat 0000 H sampai 0FFF H secara internal. Jika EA bernilai rendah maka program akan menempati alamat 1000 H sampai FFFF H ke program eksternal.

b. **Memori data**

Memori data internal dipetakan seperti pada gambar di bawah ini Ruang memorinya dibagi menjadi tiga blok yaitu bagian 128 bawah, 128 atas, dan ruang SFR (*Special Function Register*). Bagian RAM 128 byte bawah dipetakan menjadi 32 byte bawah dikelompokkan menjadi 4 bank dan 8 register (R0 sampai R7). Pada bagian 16 byte berikutnya, di atas bank-bank register, membentuk suatu blok ruang memori yang bisa teralamati per bit (*bit addressable*).

Alamat-alamat bit ini adalah 00 H hingga 7F H. Semua byte yang berada di dalam 128 bawah dapat diakses baik secara langsung maupun tidak langsung. Bagian 128 atas hanya dapat diakses dengan pengalamatan tidak langsung. Bagian 128 atas dari RAM hanya ada di dalam piranti yang memiliki RAM 256 byte.

2.2 Sensor

Sensor adalah jenis tranduser yang digunakan untuk mengubah besaran mekanis, magnetis, panas, sinar, dan kimia menjadi tegangan dan arus listrik. Sensor sering digunakan untuk pendeteksian pada saat melakukan pengukuran atau pengendalian.

Beberapa jenis sensor yang banyak digunakan dalam rangkaian elektronik antara lain : sensor cahaya, sensor suhu, sensor tekanan, dan sensor suara.

2.2.1 Sensor Suara

Sensor suara adalah sensor yang cara kerjanya yaitu merubah besaran suara menjadi besaran listrik, dan dipasaran sudah begitu luas penggunaannya. Komponen yang termasuk dalam Sensor suara yaitu :

Microphone adalah komponen elektronika dimana cara kerjanya yaitu membran yang digetarkn oleh gelombang suara akan menghasilkan sinyal listrik, dan lain – lain.

Sensor yang digunakan pada saat ini adalah sensor suara dengan menggunakan IC tipe LM 567.

2.2.2 IC LM 567

IC tipe LM 567 biasanya disebut juga *tone detector*, *tone detector* dapat digunakan untuk merubah input suara menjadi data digital. Frekuensi yang akan diambil dapat kita setting sendiri dengan merubah nilai R dan C yang telah ditetapkan sesuai dengan datasheet. Dengan LM 567 kita bisa membuat sebuah sensor pendeteksi suara.

Sensor pendeksi suara ini memanfaatkan *condensor microfon* (CM) kemudian akan diolah dengan penguatan sinyal agar suara yang diterima bisa dikuatkan, suara yang akan diolah pada mikrokontroller tentunya tidak bisa diolah secara langsung. Untuk itu frekuensi sinyal yang diterima akan diolah kembali dengan menggunakan sistem tone decoder memanfaatkan IC LM 567. Dari IC LM 567 semua sinyal yang masuk akan di filter untuk mengurangi noise ataupun suara yang tidak diinginkan.

Kelebihan dari IC LM 567 diantara adalah :

1. 20 ke 1 range frekwensi dengan penambahan resistor tambahan.

2. Output digital dengan keluaran arus sebesar 100mA

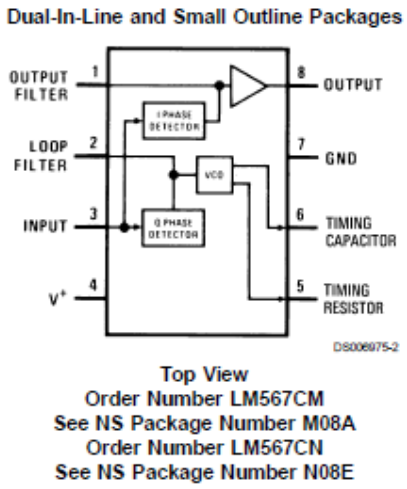
3. *Bandwidth* dapat diset antara 0 – 14%

4. Mampu menahan *noise* yang masuk

5. Mampu menahan kerusakan sinyal

6. Kestabilan pada *center frequency*

7. *center frequency* dapat diset diatara 0,01 Hz hingga 500 KHz.



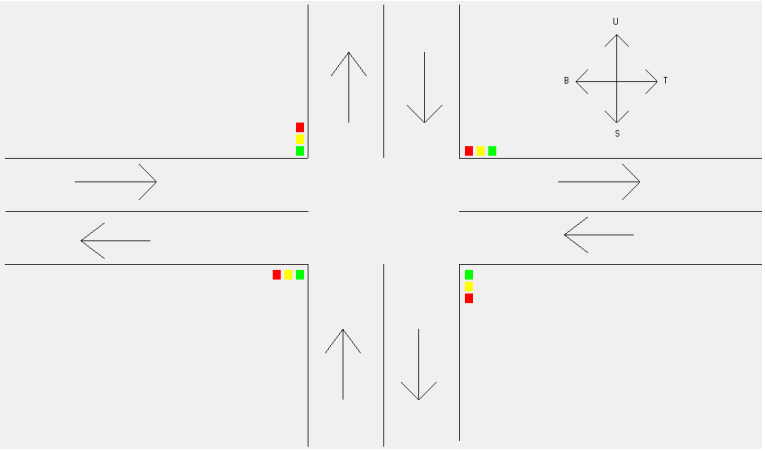
Gambar 2.3 Pandangan Atas LM567

2.3 Lampu Lalu Lintas

Menurut Penjelasan UU Lalu Lintas No. 14 tahun 1992 pasal 8 ayat 1 huruf C menyebutkan bahwa “Pengertian alat pemberi isyarat lalu lintas adalah peralatan teknis berupa isyarat lampu yang dapat dilengkapi dengan bunyi untuk

memberi peringatan atau mengatur lalu lintas orang dan/atau kendaraan di persimpangan, persilangan sebidang ataupun pada arus jalan".

Jadi lampu lalu lintas dapat diartikan sebagai lampu yang digunakan untuk mengatur kelancaran lalu lintas di suatu persimpangan jalan dengan cara memberi kesempatan pengguna jalan dari masing–masing arah untuk berjalan secara bergantian.



Gambar 2.4 Pandangan Atas Perempatan Lampu Lalu Lintas

Pada setiap lampu lalu lintas terdapat 3 buah lampu yang berwarna merah, kuning, dan hijau. Merah berarti berhenti, kuning berarti hati-hati, sedangkan hijau berarti jalan.

Secara *default*, setiap lampu lalu lintas akan mengatur laju kendaraan yang akan berjalan lurus dan berbelok ke kanan. Sedangkan belok kiri diperbolehkan langsung kecuali ada lampu lalu lintas atau rambu-rambu lalu lintas lain yang mengatur belokan ke kiri. Hal itu telah diatur di Penjelasan UU Lalu Lintas No.14 tahun 1992.

Adapun arah arus kendaraan pada setiap jalur yaitu, arus kendaraan dapat berbelok ke arah kiri secara langsung kemudian arus kendaraan dengan arah lurus dan berbelok ke arah kanan harus sesuai dengan lampu lalu lintas. Misalkan pada jalur timur, arah mobil yang berbelok ke arah kiri (menuju jalur selatan) dapat langsung berbelok sedangkan mobil yang akan berbelok ke arah kanan dan lurus harus sesuai dengan lampu lalu lintas yang telah diprogram sebelumnya. Proses tersebut akan berlangsung sama pada jalur barat, jalur utara, dan jalur selatan juga.

Pada uji alat *hardware* dan *software* ini, untuk mensimulasikan arus kendaraan akan diberi penunjuk arah arus kendaraan tersebut dengan tanda panah. Untuk uji gerak mobil pada alat *hardware* dan *software*, dilakukan secara manual yaitu dengan menggerakkan mobil yang telah terpasang suara sirine *ambulance* untuk melewati salah satu sensor suara. Sedangkan untuk gerak mobil dan mensimulasikan kepadatan kendaraan dilakukan secara otomatis yang disimulasikan pada *visual basic*.

2.4 Visual Basic

Visual BASIC (*Beginners All-Purpose Symbolic Instruction Code*) merupakan Bahasa pemrograman *Integrated Development Environment* (IDE), yaitu bahasa pemrograman visual yang digunakan untuk membuat program aplikasi atau software berbasis sistem operasi Microsoft Windows, dengan menggunakan model pemrograman *Common Object Model* (COM) visual basic merupakan turunan bahasa pemrograman BASIC yang menawarkan pengembangan perangkat lunak komputer bebasis grafik dengan cepat.

Dengan menggunakan bahasa pemrograman *visual basic* para programmer dapat membangun aplikasi dengan menggunakan komponen-komponen yang di sediakan *visual basic*. Visual Basic merupakan salah satu bahasa pemrograman yang dapat digunakan untuk menyusun dan membuat program aplikasi pada lingkungan sistem operasi Windows. Dengan menggunakan *visual basic*, kemampuan Windows dapat dimanfaatkan secara optimal.

Kecanggihan yang dimiliki oleh *visual basic* akan menjadikan betapa mudahnya menyusun program aplikasi dengan tampilan grafis yang menawan dalam waktu yang relatif singkat. Program aplikasi dapat berupa program database, program grafis, program kendali, dan lain sebagainya.

Adapun beberapa keuntungan mengapa menggunakan *visual basic* sebagai tampilan perempatan jalan dan lampu lalu lintas daripada bahasa pemrograman yag lain diantaranya :

1) Tampilan grafis (*under Windows*) sehingga lebih "bersahabat".

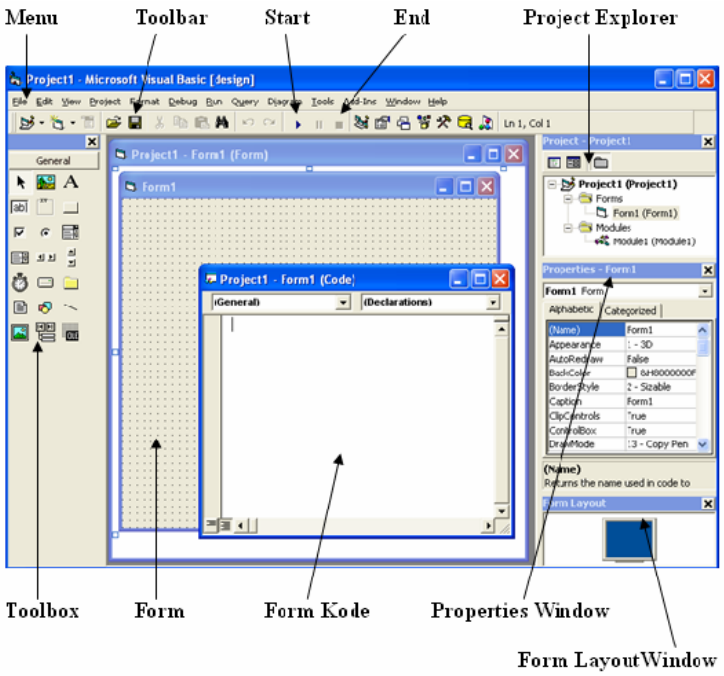
2) Cara pemrograman relatif lebih mudah sehingga cocok untuk segala tingkat programmer.

3) Hubungan dengan perangkat luar (*hardware*) tidak begitu rumit sehingga cukup mudah untuk meng-implementasikan sebagai pengendali peralatan elektronik.

2.4.1 IDE Visual Basic

Langkah pertama dalam membuat program aplikasi dengan Visual Basic 6.0 adalah membuat sebuah project. Pembuatan sebuah project dapat dilakukan dengan beberapa cara, diantaranya dengan meng-klik **Start |Program | Microsoft Visual Studio 6.0 | Microsoft Visual Basic 6.0**.

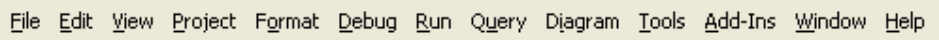
Setelah itu akan terlihat tampilan pilihan jenis *New Project*, pilih *Standart EXE* maka akan terlihat tampilan IDE *(Integrated Development Environment)* Visual Basic.



Gambar 2.5 Tampilan IDE Visual Basic

1) Menu Bar

Visual Basic mempunyai tiga belas menu dan masing-masing menu mempunyai fungsi yang berbeda.



Gambar 2.6 Menu Bar

2) Toolbar

Toolbar mempunyai fungsi yang sama dengan menu, hanya saja berupa icon-icon gambar dan digunakan sebagai jalan pintas.



Gambar 2.7 Toolbar

3) Toolbox

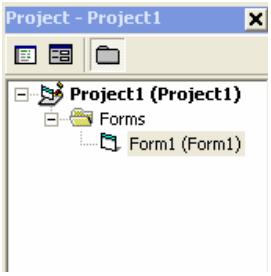
Toolbox merupakan tempat kontrol-kontrol yang akan digunakan untuk membantu pembuatan program aplikasi.



Gambar 2.8 Toolbox

4) Project Explorer

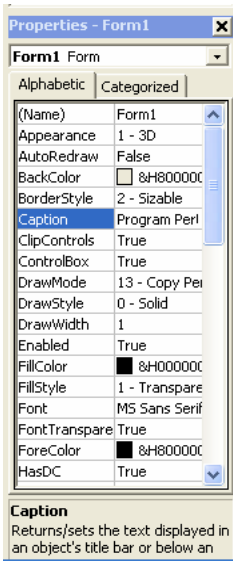
Project Explorer merupakan tempat yang digunakan untuk melihat daftar forms, modules, class modules, dan designers.



Gambar 2.9 Project Explorer

5) Properties Window

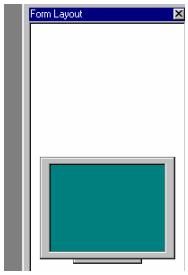
Properties Window berfungsi untuk mengatur properti dari setiap objek kontrol atau form. Pada Properties Window semua objek kontrol dapat diatur karakteristiknya.



Gambar 2.10 Property Window

6) Form Layout Window

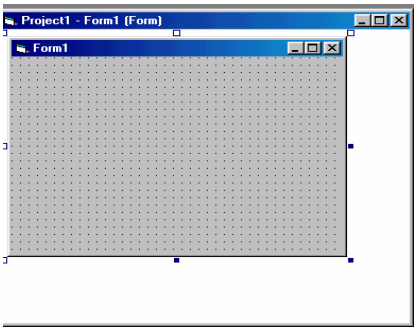
Form layout window berfungsi untuk melihat atau mengetahui posisi tampilan form saat program dijalankan.



Gambar 2.11 Form Layout Window

7) Form Objek

Form objek digunakan untuk menempatkan atau meletakkan objek dari kontrol-kontrol yang akan digunakan untuk merancang dan membuat program aplikasi.



Gambar 2.12 Form Objek

8) Form Kode Window

Form kode digunakan sebagai tempat untuk menulis kode-kode program aplikasi.



Gambar 2.13 Kode Window

2.4.2 Bahasa Pemrograman Visual Basic

a. Variabel

Setiap melakukan pemrograman, akan selalu memerlukan tempat penyimpanan data, misalnya untuk menampung data hasil perhitungan, menampung data hasil pembacaan register, atau lainnya. Tempat penyimpanan data itu dinamakan variabel yang merupakan pointer yang menunjuk pada alamat memori fisik tertentu di komputer.

Dalam penggunaannya variabel harus mempunyai nama dan tipe data tertentu. Nama variabel menunjuk pada suatu tempat pada memori komputer, sedangkan tipe data mengontrol besarnya memori yang disediakan untuk variabel tersebut.

Pada *visual basic* terdapat dua cara untuk mendeklarasikan sebuah variabel, yaitu dengan cara deklarasi eksplisit dan cara deklarasi implisit. Deklarasi eksplisit menggunakan pernyataan "Dim" diikuti nama dan tipe datanya, sedangkan deklarasi implisit menggunakan simbol dibelakang nama variabel yang mempresentasikan tipe data yang digunakan.

b. Kontrol Program

Dengan kontrol program, alur eksekusi program dapat dikendalikan serta dapat menentukan keputusan apa yang harus dikerjakan oleh program pada kondisi tertentu.

Kontrol program pada *visual basic* meliputi kontrol pertimbangan kondisi dan keputusan, kontrol pengulangan serta kontrol penyaluran alternatif.

Beberapa kontrol program pada Visual Basic yang digunakan pada pemrograman ini :

1) If ... Then

Pernyataan ini mengetes suatu kondisi berdasarkan syarat kondisi kemudian menentukan suatu tindakan jika kondisi tersebut dipenuhi yang berupa pernyatan.

2) If ... Then ... Else

Pernyataan ini hampir sama dengan If ... Then ..., yaitu digunakan untuk mengetes suatu kondisi tertentu. Hanya saja, jika suatu kondisi tidak terpenuhi, maka alur program akan mengeksekusi pernyataan yang lain kemudian menentukan suatu tindakan jika salah satu kondisi tersebut terpenuhi.

3) Select ... Case

Pada dasarnya perintah ini sama dengan perintah If ... Then ... Else, yaitu akan mengeksekusi satu blok pernyataan dari beberapa pilihan blok pernyataan. Hanya saja penulisannya lebih ringkas dan lebih mudah dimengerti.

4) Do ... Loop

Perintah Do ... Loop digunakan untuk perulangan suatu blok pernyataan sampai dipenuhinya syarat kondisi yang ditetapkannya.

5) For ... Next

Perintah ini sama dengan melakukan perulangan seperti perintah Do ... Loop, tetapi dengan For ... Next bisa ditentukan nilai awal dan nilai akhir perulangan serta nilai kenaikannya.

c. Prosedur

Pembuatan program akan lebih mudah dengan memecah program menjadi blok-blok komponen yang lebih kecil yang disebut prosedur. Prosedur sangat berguna ketika sering melakukan tugas yang sama berulang-ulang atau bermaksud membagikannya pada program yang lain.

1) Sub Procedure

Salah satu jenis prosedur yang ada didalam *visual basic* adalah *Sub Procedure*. *Sub Procedure* adalah blok kode yang dijalankan sebagai tanggapan atas terbentuknya even, baik even itu merupakan even pemanggilan dari prosedur lain maupun even yang terjadi dari pemakaian program, misal even penekanan tombol kiri mouse. Setiap kali prosedur dipanggil, blok pernyataan yang ada di antara Sub dan End Sub akan dijalankan.

2) Operator

Operator digunakan pada *visual basic* untuk memanipulasi data maupun untuk melakukan perhitungan. Operator pada *visual basic* dapat dikelompokkan menjadi tiga kelompok, yaitu operator matematik, operator perbandingan, dan operator logika.

- Operator Matematik

Operator matematik digunakan untuk melakukan perhitungan matematik.

Tabel 2.1 Operator Matematik

Operator	Operasi	Contoh pemakaian
^	Pemangkatan	Nilai% = 2^2 'menghasilkan 4
-	Tanda negatif	Nilai% = -5 'menghasilkan negatif 5
*, /	Perkalian dan pembagian	Nilai% = (2*3)/6 'menghasilkan 1
\	Pembagian integer	Nilai% = 13\4 'menghasilkan 3
Mod	Modulus (sisa pembagian)	Nilai% = 10 Mod 5 'menghasilkan 2
+, -	Penambahan dan pengurangan	Nilai% = 6 + 2 - 4 'menghasilkan 4
&	Penggabungan string	Teks$ = "aw" & "an" 'menghasilkan "awan"

- Operator Perbandingan

Operator perbandingan digunakan untuk membandingkan suatu ekspresi dengan ekspresi yang lain dan akan menghasilkan nilai Boolean (*False* atau *True*).

Tabel 2.2 Operator Perbandingan

Operator	Operasi	Contoh pemakaian
=	Sama dengan	Nilai = (1 + 2) = 3 'menghasilkan True
<>	Tidak sama dengan	Nilai = (1 + 2) <> 3 'menghasilkan False
<	Lebih kecil	Nilai = 2 < 3 'menghasilkan True
>	Lebih besar	Nilai = 2 > 3 'menghasilkan False
<=	Lebih kecil atau sama dengan	Nilai = 2 <= 3 'menghasilkan True
>=	Lebih besar atau sama dengan	Nilai = 2 >= 3 'menghasilkan False
Like	Mempunyai ciri yang sama	Nilai = "abba" Like "a*" 'True Nilai = "abba" Like "a??a" 'True Nilai = "abba" Like "a?a" 'False Nilai = "a" Like "[a-z]" 'True
Is	Mempunyai referensi obyek yang sama	Nilai = Command1 Is Label1 'menghasilkan False

- Operator Logika

  Operator logika biasanya digunakan untuk mengekspresikan satu atau lebih ekspresi logika yang akan menghasilkan nilai Boolean.

Tabel 2.3 Operator Logika

Operator	Keterangan	Tabel Kebenaran Operasi	
		Operan	Hasil
Not	Akan menghasilkan nilai kebalikan dari nilai operan	Not True	False
		Not False	True
And	Akan menghasilkan True jika kedua operan-nya berlogika True	True And True	True
		True And False	False
		False And True	False
		False And False	False
Or	Akan menghasilkan True jika salah satu operan-nya berlogika True	True Or True	True
		True Or False	True
		False Or True	True
		False Or False	False
Xor	Akan menghasilkan True jika operan-nya berlogika berbeda	True Xor True	False
		True Xor False	True
		False Xor True	True
		False Xor False	False
Eqv	Akan menghasilkan True jika operan-nya berlogika sama	True Eqv True	True
		True Eqv False	False
		False Eqv True	False
		False Eqv False	True

Berikut adalah contoh *listing* program pada visual basic :

```
Private Sub Timer1_Timer()
If Shape1.Visible Then
Shape2.Visible = True
Shape1.Visible = False
Shape3.Visible = False
Timer2 = True
Timer3 = True
ElseIf Shape2.Visible Then
Shape3.Visible = True
Shape2.Visible = False
```

```
Shape1.Visible = False
Timer2 = True
Timer3 = True
Else
Shape1.Visible = True
Shape2.Visible = False
Shape3.Visible = False
Timer2 = False
Timer3 = False
End If
End Sub
Private Sub Timer2_Timer()
Image1.Left = Image1.Left + 200
If Image1.Left > 10000 Then
Image1.Left = -2500
End If
End Sub
Private Sub Timer3_Timer()
Image2.Left = Image2.Left + 200
If Image2.Left > 10000 Then
Image2.Left = -2500
End If
End Sub
Private Sub Timer4_Timer()
p = p + 1
warna = warna + 300
If p = 20 Then p = 0
Label1.ForeColor = warna
End Sub
```