# BAB II

# DASAR TEORI

## 2.1 Sistem Pengendali

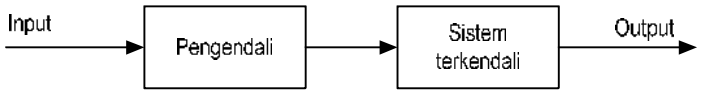
Dalam sistem pengendali ini terdiri dari alat mekanik dan elektronik. Sistem pengendali dibagi menjadi dua, yaitu loop terbuka dan loop tertutup.

### 2.1.1 Sistem Kendali Loop Terbuka

Sistem kendali terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya. Karena sistem sifatnya terbuka dan terpengaruh oleh lingkungan luarnya, maka suatu sistem harus mempunyai suatu sistem pengendali yang baik.

Walaupun tidak memberikan umpan balik sebagai pembanding untuk validasi, sistem kendali terbuka masih tetap memungkinkan untuk menyelesaikan perintah dengan baik asalkan objek terkendali tidak mengalami perubahan yang tidak terpredeksi sebelumnya.

Sistem pengendalian dengan loop terbuka lebih sederhana dan lebih murah dari pada loop tertutup yang lebih rumit, dengan catatan implementasinya harus disesuaikan pada situasi dimana semua pengaruh pada objek terkendali dapat diprediksi serta tidak tergantung faktor-faktor luar.



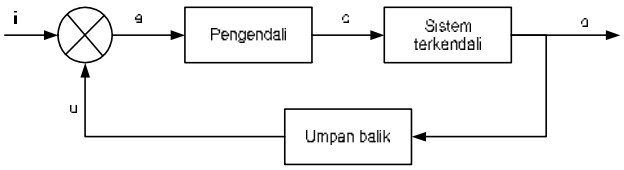

**Gambar 2.1** Diagram Blok untuk Kendali Terbuka

Sistem pengendali sangat berkaitan erat dengan kemajuan teknologi komputer. Dengan menggunakan suatu bahasa pemrograman tertentu komputer memungkinkan untuk dapat mengendalikan kinerja peralatan lain, sehingga dapat diperintah sesuai dengan kondisi yang diinginkan.

### 2.1.2 Sistem Kendali Tertutup

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tampa adanya turut campur tangan dari pihak diluarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup).

Sistem kendali loop tertutup ini terdapat umpan balik dari sinyal sistem terkendali yang berfungsi sebagai validasi hasil keluaran terhadap sinyal input.




**Gambar 2.2** Diagram Blok untuk Kendali Tertutup

Keterangan:

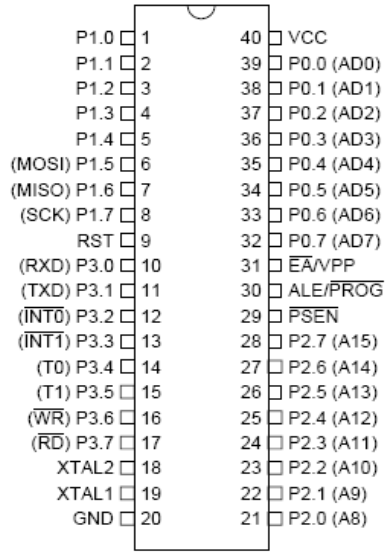
i : masukan referensi yang menentukan suatu nilai bagi sistem yang dikendalikan
u : sinyal umpan balik dari keluaran
c : sinyal kontrol

o :  sinyal keluaran

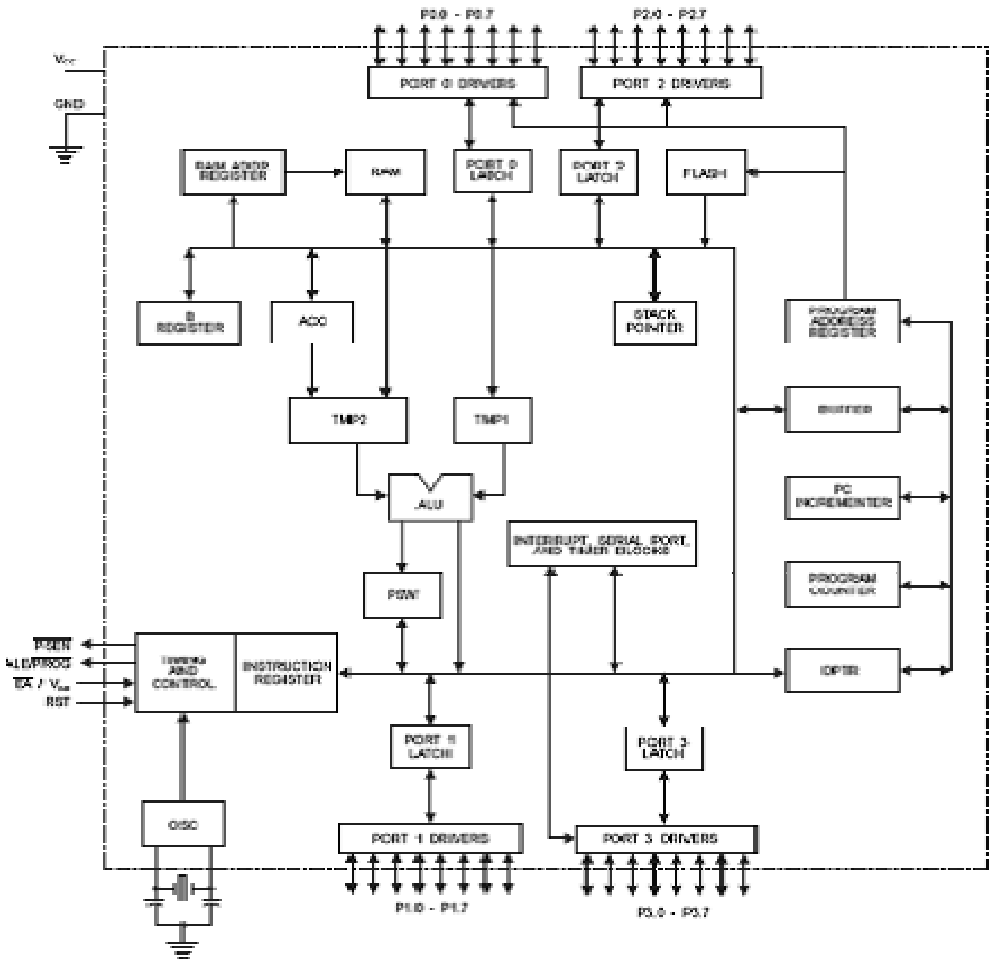
## 2.2 Mikrokontroler Atmel AT89S51

Mikrokontroler ialah sebuah chip yang berfungsi sebagai pengontrol rangkaian elektronik dan umumnya menyimpan program didalamnya.

Salah satu jenis mikrokontroler adalah Atmel AT89S51, AT89S51 ialah mikrokontroler CMOS 8 bit dengan 4KB *Flash Programmable and Erasable Read Only Memory* (PEROM). Mikrokontroler ini mempunyai memori yang terdiri dari RAM internal dan *Special Function Register*. Produk keluaran Atmel ini memiliki 40 kaki. Dari 40 kaki yang tersedia, 32 kaki dibagi menjadi 4 buah pararel port. Port-port tersebut yaitu P0, P1, P2 dan P3. Masing-masing port memiliki 8 kaki input/ output yang berfungsi sebagai input dan output program kontrol. Bentuk mikrokontroler AT89S51 dapat dilihat pada Gambar 2.3




**Gambar 2.3** Mikrokontroler AT 89S51




**Gambar 2.4** Arsitektur Mikrokontroler

Selain kaki-kaki I/O yang berjumlah 32 kaki, AT89S51 juga memiliki kaki-kaki yang lain, yaitu RST, XTAL1, XTAL2, GND, ALE/PROG, EA/VPP, PSEN dan VCC.

Fungsi dari kaki-kaki tersebut adalah :

1. P0.0 – P0.7 Port 0 dapat berfungsi sebagai I/O biasa, *low order* multiplex *addres* atau data ataupun menerima kode byte pada saat *Flash Programming*. Pada fungsi sebagai I/O biasa port ini dapat memberikan output sink ke

delapan buah TTL input atau dapat diubah sebagai input dengan memberikan logika 1 pada port tersebut.

2. P1.0 – P0.7 Port 1 berfungsi sebagai I/O biasa, *low order* multiplex *addres* atau data ataupun menerima kode byte pada saat *Flash Programming*. Port ini mempunyai internal pull up dan berfungsi sebagai input dengan memberikan logika 1.
3. P2.0 – P2.7 Port 2 berfungsi sebagai I/O biasa atau *high order address*, pada saat mengakses memori secara 16 bit ( Movx @ DPTR ). Pada saat mengakses memori secara 8 bit, ( Mov @Rn ) port ini akan mengeluarkan isi dari P2 *Special Function Register*.
4. P3.0 (RXD) berfungsi untuk serial input, P3.1 (TXD) untuk serial output, P3.2 (INT0) untuk External Interrupt 0, P3.3 (INT1) untuk External Interrupt 1, P3.4 (T0) untuk External Timer 0 Input, P3.5 (T1) untuk External Timer 1 input, P3.6 (WR) untuk External Data Memory Write Strobe, P3.7 (RD) untuk External Data Memory Write Strobe.
5. RST (kaki 9) merupakan masukan reset (aktif tinggi), pulsa transisi dari rendah ke tinggi me-reset mikrokontroler.
6. XTAL1 (kaki 19) dan XTAL2 (kaki 18) diperlukan untuk menghubungkan mikrokontroler dengan kristal.
7. GND (kaki 20) merupakan ground sumber tegangan.
8. PSEN (*Program Store Enable*) (kaki 29) adalah kaki yang berfungsi memberikan sinyal pengontrol untuk mengakses program memori eksternal

masuk ke dalam bus selama proses pemberian atau pengambilan intruksi (*fetching*).

9. ALE/ PROG (kaki 30) adalah kaki yang mempunyai dua fungsi. Fungsi yang pertama adalah *Address Latch Enable* (ALE) yaitu penahan alamat memori eksternal. Fungsi kedua yaitu sebagai pemberi sinyal atau pulsa input pemrograman (PROG) selama proses pemrograman.

10. EA/ VPP (kaki 31) adalah kaki yang mempunyai dua fungsi . Fungsi pertama adalah sebagai kaki *External Access Enable* (EA), yaitu kaki yang memberi sinyal kontrol untuk pembacaan memori program. Fungsi kedua sebagai tegangan pemrograman (+12V) selama proses pemrograman (VPP).
11. VCC (kaki 40) merupakan sumber tegangan positif.

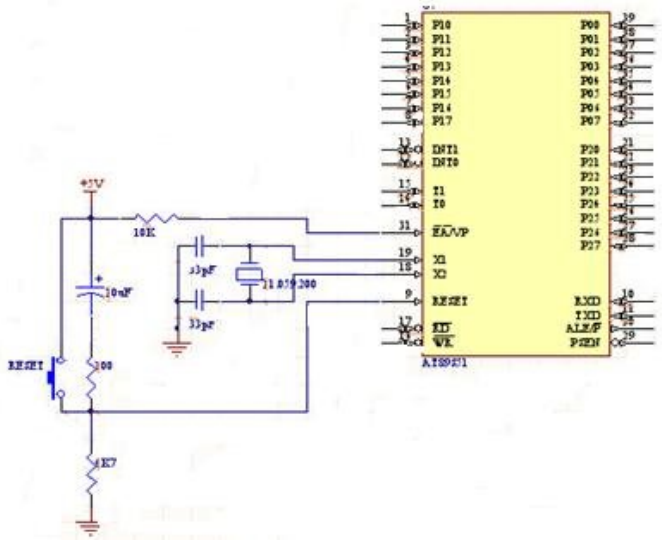
**2.3 Organisasi Memori**

Pada mikrokontroler AT89S51 terdapat dua macam memori, yaitu *program memory* dan *data memory*. *Program memory* adalah tempat untuk menyimpan program yang berisi intruksi-intruksi (*intruction storage*), sedangkan *data memory* adalah tempat untuk menyimpan data (*data storage*).

*Program memory* dapat berbentuk ROM internal dan eksternal, sedangkan *data memory* dapat berbentuk RAM internal dan eksternal.

Mikrokontroler AT89S51 mempunyai kemampuan pemrograman berulang dengan flash memori. Dengan kemampuan ini AT89S51 ini dapat menyimpan dan menghapus program beberapa kali. Batas proses pemrograman ulang ini sampai 1000

kali. AT89S51 mempunyai RAM internal dengan kapasitas 128x8 bit dengan frekuensi pengoperasian hingga 24 MHz. Agar mikrokontroler dapat berkerja, maka beberapa kaki mikrokontroler ini harus dihubungkan dengan komponen elektronika. Berikut adalah rangkaian dasar mikrokontroler :



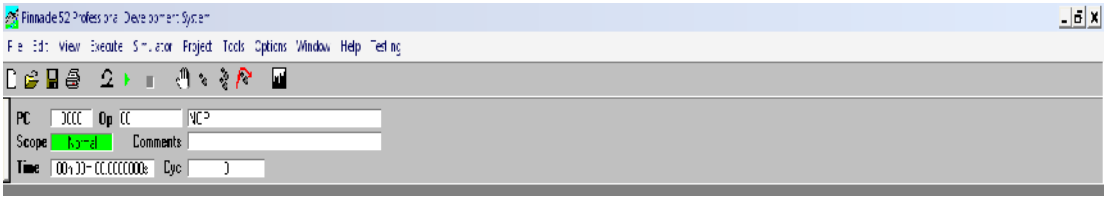
**Gambar 2.5** Rangkaian Dasar Mikrokontroler AT89S51

## 2.4 Program Pinnacle 52

Program pinnacle 52 merupakan suatu program yang dapat mensimulasikan proses yang terjadi dalam mikrokontroler AT89S51 dan menampilkan kondisi-kondisi pada memorinya serta dapat mengkompilasi program yang dibuat ke dalam bentuk OBJ, LST dan HEX.

Berikut ini akan dijelaskan beberapa menu yang sering digunakan pada pembuatan dan simulasi program. Pada bagian diatas program aplikasi pinnacle 52

terdapat beberapa menu utama, diantaranya: Menu File, Edit, View, Execute, Simulator, Project, Tools, Options, Windows, dan Help.

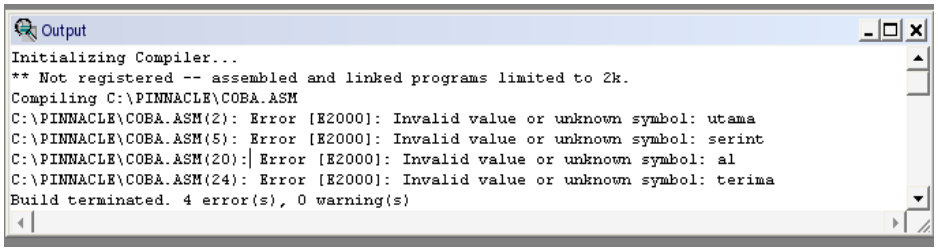


**Gambar 2.6** Tampilan menu utama program Pinnacle 52

Pembuatan program dilakukan pada editor. Setelah program selesai dibuat, sebaik nya program disimpan dan selanjutnya dilakukan kompilasi agar diperoleh file OBJ. Atau langsung dilakukan kompilasi dan link agar diperoleh file OBJ dan HEX (file HEX digunakan untuk dimasukan ke dalam IC AT89S51).

### 2.4.1 Kompilasi Program

Proses kompilasi dapat dilakukan melalui menu project kemudian pilih Compile <namafile.asm> atau langsung menekan tombol F2. Berikut ini contoh tampilan hasil kompilasi file Coba.asm.

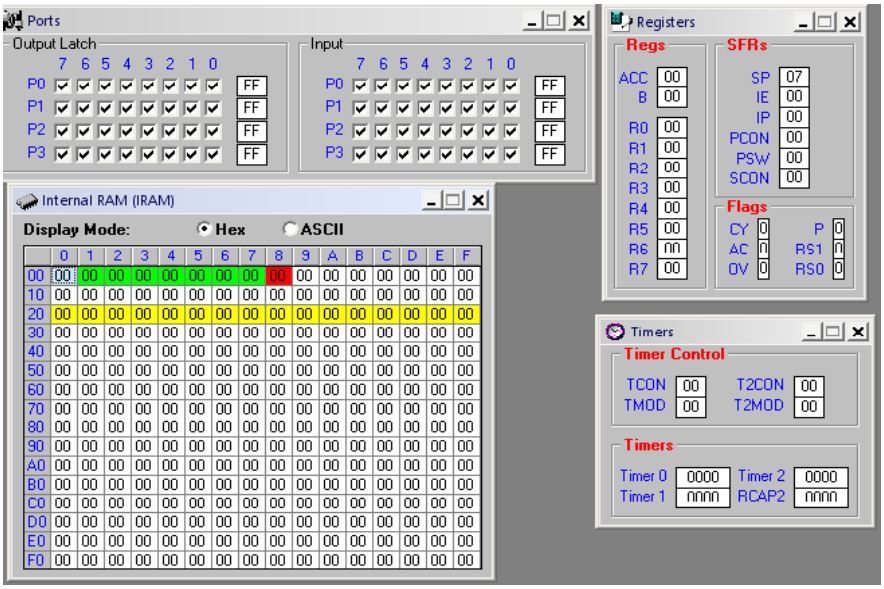



**Gambar 2.7** Compile dan Link

### 2.4.2 Simulasi Program

Selanjutnya program dapat disimulasikan melalui menu *execute* kemudian pilih Run atau langsung menekan tombol F5. Selain itu simulasi dapat dilakukan secara per instruksi dengan menekan tombol F8 untuk setiap instruksi.

Menu View digunakan untuk menampilkan beberapa tampilan seperti register, port, timer dan internal RAM. Berikut tampilan yang dapat dipanggil melalui menu View.




**Gambar 2.8** Beberapa Tampilan dari menu View

### 2.4.3 Mode Pengalamatan

Pengalamatan ( *Addressing* ) adalah proses mengakses sebuah lokasi, baik untuk pembacaan ataupun penulisan. Cara untuk mengakses lokasi memori dapat

digolongkan kedalam beberapa kelompok, antara lain: *direct*, *indirect*, dan *immediate*.

1. Mode pengelamatan segera ( *Immediate Addressing Mode* )

   Yaitu proses mengisi suatu alamat memori dengan suatu nilai tertentu. Cara ini biasanya menggunakan konstanta.

   Contoh : mov A, #30h

   Nilai 30h akan langsung diisikan kedalam *accumulator* A

2. Mode pengalamatan langsung ( *Direct Addressing Mode* )

   Yaitu proses sebuah lokasi memori secara langsung dengan menggunakan alamatnya. Cara ini dipakai untuk menunjuk data yang berada disuatu lokasi memori dengan cara menyebut lokasi ( alamat ) tempat data tersebut berada.

   Contoh : Mov A, 47h

   Isi dialamat 47h akan langsung diisikan ke *accumulator*

3. Mode pengalamatan tidak langsung ( *Indirect Addressing Mode* )

   Yaitu proses mengakses alamat secara tidak langsung. Cara ini dipakai untuk mengakses data yang berada didalam memori, tetapi lokasi memori tidak disebut secara langsung tapi di-“titip”-kan keregister lain.

   Contoh : mov A, @R0

   *accumulator* A akan terisi data yang berada dialamat yang ditunjukan oleh register R0.

   tanda “@” dipakai untuk menandai lokasi memori yang tersimpan didalam R0.

4. Mode pengalamatan Register ( *Register Addressing Mode* )

Yaitu proses mengakses memori dengan label antara R0 sampai R7

Contoh : mov A, R0

Data dalam register serbaguna R0 disalin ke *accumulator* A.

5. Mode pengalamatan bit ( *Bit Addressing Mode* )

   Pada pengalamatan ini, operand menunjuk ke alamat internal RAM atau register fungsi khusus yang dapat dialamat secara bit (bit addressable)

   Contoh : setb P3.1 : lakukan set pada port 3 bit ke 1

### 2.4.4 Pengarah Assembler

1. ORG ( *Origin* )

   Digunakan untuk mengarahkan lokasi tempat instruksi yang ada dibawah nya.

   Contoh :

   ORG 100 h :  alamat awal untuk instruksi dibawah berada dialamat 100h

   MOV A, #10h : instruksi mov A, #10h berada di alamat 100h

2. END

   Digunakan sebagai tanda akhir dari program assembler

3. BIT

   Digunakan untuk mendefinisikan sebuah lambang yang menunjuk kelokasi bit pada memory yang dapat dialamati secara pengalamatan bit

Contoh :

key_1 bit P1.0 : konstansta “key_1” didefinisikan dengan port 1 bit 0

### 2.4.5 Instruksi Transfer Data

1. Mov A,Rn

   Melakukan pemindahan data dari Rn ( R0…..R7 ) menuju ke akumulator A

   Contoh :

   Mov A, R1 : Data dari R1 pindah ke akumulator A

2. Mov A, direct

   Melakukan pemindahan data dari alamat langsung ke akumulator A

3. Mov Rn, A

   Melakukan pemindahan data dari akumulator A menuju ke Rn ( R0….R7 )

4. Mov Rn, direct

   Melakukan pemindahan data dari alamat langsung menuju ke Rn ( R0…R7)

   Contoh :

   Mov R7,10h : Data di alamat 10h dipindah kedalam R7

5. Mov Rn, #Data

Melakukan pemindahan data dari immediat*e* menuju ke Rn ( R0…R7 )

Contoh :

Mov R7, #05h : Data 05h dipindah ke dalam R7

### 2.4.6 Contoh Program

```
Org 100h

Mov A, #16h        : Salin 16 H ke Akumulator

Mov R1, A          : Salin data A ke R1

Mov A, #32h        : Salin 32 H ke Akumulator

Mov @R1, #64h      : Salin data 64h ke alamat yang ditunjukan R1

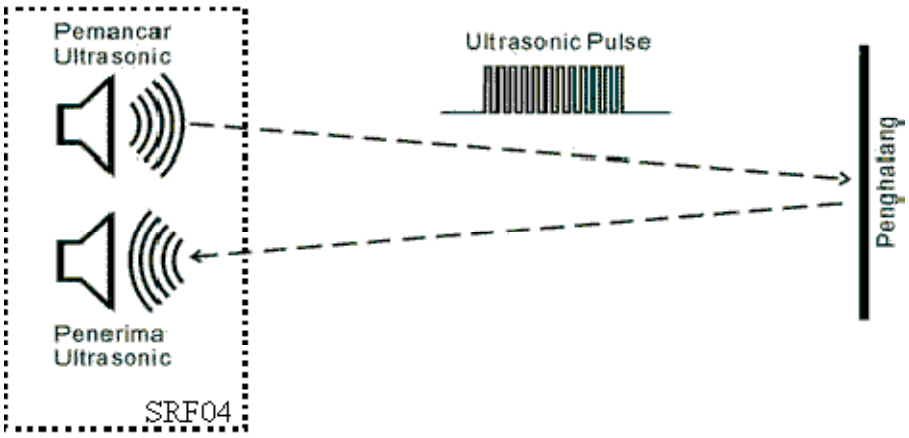
Mov P1, 16h        : Salin data dialamat 16 ke port 1
```

**Table 2.1** Instruksi Bahasa Assembly

| No | Nama Perintah | Penjelasan |
|---|---|---|
| 1 | MOV | Salin |
| 2 | ORG | alamat awal |
| 3 | CLR | Hapus |
| 4 | NOP | Tidak ada operasi |
| 5 | ACALL | Panggilan |
| 6 | LCALL | lama panggilan |
| 7 | JB | Melompat jika bit diatur |
| 8 | DJNZ | Pengurangan, melompat jika tidak nol |
| 9 | CJNE | Bandingkan, melompat jika tidak sama |
| 10 | JMP | Melompat |
| 11 | DELAY | Penundaan |
| 12 | DIV | Membagi |
| 13 | RET | Kembali dari subroutine |
| 14 | SJMP | Pendek melompat |
| 15 | SETB | Set bit |
| 16 | LJMP | Lompat jauh |
| 17 | TMOD | Timer register |
| 18 | JNB | Melompat jika bit tidak diatur |
| 19 | ANL | Logis |
| 20 | CPL | Melengkapi (complement ) |
| 21 | DEC | Pengurangan ( decrement ) |
| 22 | INC | Kenaikan ( increment ) |
| 23 | JBC | Melompat jika mengatur sedikit, maka bit yang jelas |
| 24 | JNC | Melompat jika membawa tidak diatur |
| 25 | JNZ | Melompat jika akumulator tidak nol |
| 26 | JZ | Melompat jika akumulator adalah nol |
| 27 | MOVC | Memindah kan kode byte |
| 28 | MOVX | Bergerak byte eksternal |
| 29 | ORL | Logical OR |
| 30 | POP | Pop nilai dari stack |
| 31 | PUSH | Mendorong nilai ke dalam stack |
| 32 | SUBB | Menguragi |
| 33 | XRL | Logis eksklusif OR |
| 34 | RL, RLC | Akumulator memutar kekiri |
| 35 | RR, RRC | Memutar akumulator kekanan |
| 36 | BUFFER | Penyangga (tempat penyimpanan semantara) |

## 2.5 Ultrasonik ( SRF04 )

Ultrasonik adalah merupakan sensor yang berisi *transmitter* dan *receiver* ultrasonic, sensor dapat digunakan untuk mengukur jarak. Sensor ini mengukur jarak dengan cara menghitung selisih waktu antara saat pemancaran sinyal dan saat penerimaan sinyal pantul. Seperti diketahui, kecepatan rambat suara diudara adalah 34399,22 cm/detik, berarti untuk merambat sejauh 1 cm suara membutuhkan waktu 29 mikro detik. Misalkan waktu antara pengiriman dan penerimaan sinyal ultrasonic adalah 5800 mikro detik, maka jarak antara sensor dan benda ( penghalang ) adalah 100 cm. ( 2 x 100 cm x 29 mikro detik/cm = 5800 mikro detik ).



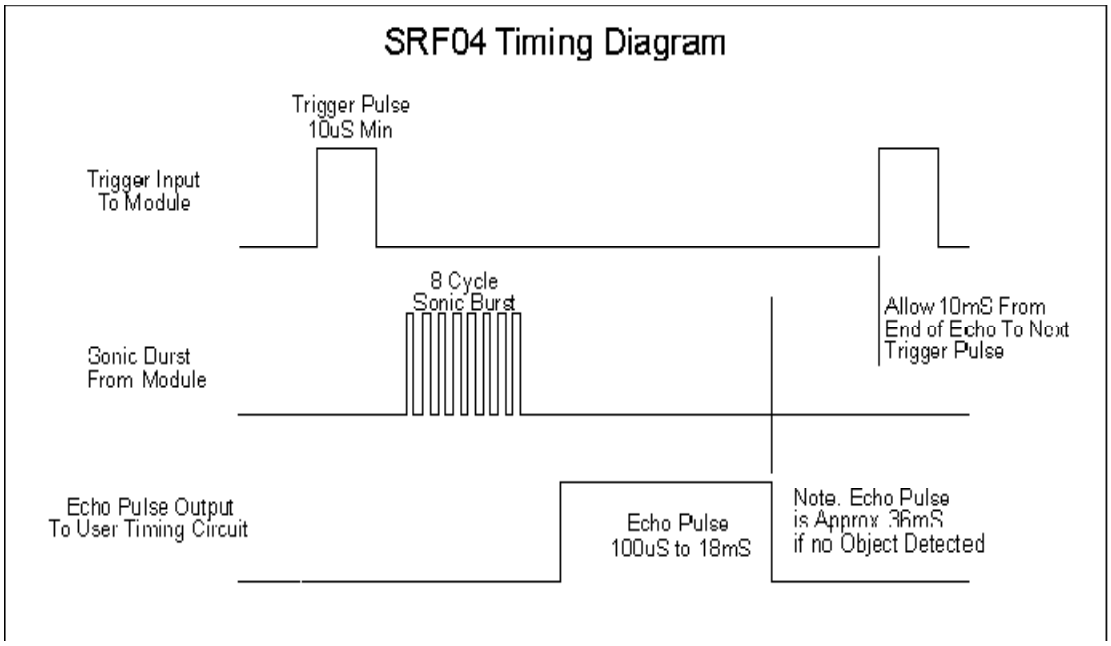

**Gambar 2.9** Prinsip kerja sensor jarak Ultrasonik

SRF04 hanya menggunakan 2 port I/O untuk berhubungan dengan mikrokontroler, sehingga sangat ideal untuk aplikasi-aplikasi robotika, SRF04 dapat mengukur jarak antara 3 cm sampai 3 m, dan dapat mengukur benda dengan diameter 3 cm pada jarak kurang dari 2 meter.

Pulsa Ultrasonik yang dikirim oleh SRF04 adalah sinyal ultrasonic dengan frekuensi 40 KHz sebanyak 8 periode setiap kali pengiriman. Ketika pulsa mengenai

benda penghalang, maka pulsa ini akan dipantulkan kembali dan diterima kembali oleh penerima ultrasonic. Dengan mengukur selang waktu antara saat pulsa dikirim dan pulsa pantul diterima, maka jarak benda penghalang bisa dihitung.

Apabila PI ( triger pulsa input ) diberi logika 1( high )selama minimal 10 uS maka SRF04 akan memancarkan sinyal ultrasonic, setelah itu pin PO ( echo pulsa output ) akan berlogika high selama 100 uS – 18 mS. ( tergantung jarak sensor dan penghalang ) dan apabila tidak ada penghalang maka PO akan berlogika 1 selama kurang lebih 38 mS.



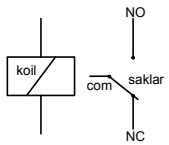

**Gambar 2.10** Timing diagram SRF04

Misalkan lama Echo Pulse adalah T, maka untuk mengetahui jaraknya dapat diketahui dengan cara membagi T dengan 58 (T/58) untuk satuan senti meter dan dibagi dengan 148 (T/148) untuk satuan inchi. Misalkan panjang Echo pulse adalah 5800 mikro detik maka jarak benda adalah 1 meter ( 5800/58=100 cm=1 meter ).

## 2.6 Relay

Relay merupakan komponen elektronika yang prinsip kerjanya hampir sama dengan saklar. Relay mempunyai lima kaki. Dua kaki koil dan tiga kaki saklar. Kaki koil dalam pemasangannya boleh terbalik. Kaki saklar terdiri dari kaki *normally open (NO), normally close (NC)* dan *common (COM).*

Dalam keadaan arus listrik mengalir (koil hidup), maka kaki COM akan terhubung ke kaki NO dan sebaliknya dalam keadaan koil relay mati maka kaki COM akan terhubung ke kaki NC. Relay dapat digunakan untuk memisahkan rangkaian yang satu dengan yang lainnya dengan atau tanpa sambungan sama sekali. Bila arus masuk ke kaki relay hal ini mengakibatkan koil relay hidup dan menimbulkan medan magnet. Medan magnet ini yang kemudian akan menggerakan saklar. Gambar simbol relay dapat dilihat pada Gambar 2.11

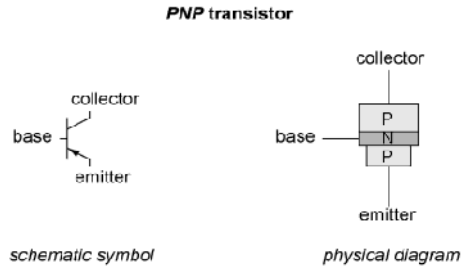



**Gambar 2.11** Simbol Relay

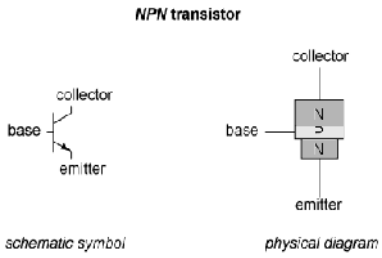
## 2.7 Transistor

Transistor adalah alat semikonduktor yang dipakai sebagai penguat arus listrik dan mempunyai tiga kaki, dengan prinsip kerja suatu arus listrik atau tegangan listrik kecil yang dipasang pada satu kaki dapat mengontrol arus yang melewati kedua kaki lainnya. Kaki-kaki tersebut adalah *basis*, *emitor* dan *collector*. Ada dua jenis

transistor yaitu transistor NPN (Negatif Positif Negatif) dan transistor PNP (Positif Negatif Positif).




**Gambar 2.12** Simbol Transistor PNP




**Gambar 2.13** Simbol Transistor NPN

Pada jenis NPN, kaki emittor selalu lebih negatif terhadap kolektornya. Pada jenis PNP berlaku hal sebaliknya. Transistor dapat juga disebut sebagai penahan arus yang dapat diatur karena dengan transistor kita dapat mempengaruhi kekuatan suatu arus. Transistor mengatur arus secara elektronis. Tidak ada suatu tombol atau lainnya yang digunakan untuk pengaturannya. Sebagian transistor disusun dalam bentuk kecil dengan tiga buah kaki.

## 2.8 Komunikasi Serial Port

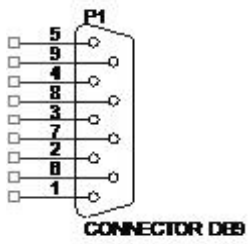
Komunikasi serial merupakan hal yang penting dalam sistem *embedded*, karena dengan komunikasi serial kita dapat dengan mudah menghubungkan mikrokontroler dengan alat lainnya. Port serial pada mikrokontroler terdiri atas dua pin yaitu RXD dan TXD, RXD berfungsi untuk menerima data dari komputer/perangkat lainnya, TXD berfungsi untuk mengirim data ke komputer atau perangkat lainnya. Standar komunikasi serial untuk komputer ialah RS-232, RS-232 mempunyai standar tegangan yang berbeda dengan serial port mikrokontroler,

sehingga agar sesuai dengan RS-232 maka dibutuhkan suatu rangkaian level converter, IC yang digunakan bermacam-macam, tetapi yang paling mudah dan sering digunakan ialah IC MAX232.

Hardware pada komunikasi 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.

Konektor port serial terdiri dari 2 jenis, yaitu konektor 25 pin (DB25) dan 9 pin (DB9) yang berpasangan (jantan dan betina).




**Gambar 2.14** Serial Port DB9

Setiap kaki dari serial port DB9 ini mempunyai fungsi masing-masing yang dijelaskan dalam tabel di bawah ini

**Tabel 2.2** Keterangan Sinyal dan Kaki Serial Port DB9

| Nama Sinyal | Arah Sinyal | Nomor Kaki Konektor |
|---|---|---|
| Data Carrier Detect (DCD) | Dari DCE | 1 |
| Received Data (RD) | Dari DCE | 2 |
| Transmitted Data (TD) | Dari DTE | 3 |
| DTE Ready (DTR) | Dari DTE | 4 |
| Ground | - | 5 |
| DCE Ready (DSR) | Dari DCE | 6 |
| Request to Send (RTS) | Dari DTE | 7 |
| Clear to Send (CTS) | Dari DCE | 8 |
| Ring Indicator (RI) | Dari DCE | 9 |

Keterangan mengenai fungsi saluran pada konektor DB9 adalah sebagai berikut:

1. *Data Carrier Detect*, dengan saluran ini DCE memberitahukan ke DTE bahwa pada terminal masukkan ada data masuk.
2. *Receive Data*, digunakan DTE menerima data dari DCE.
3. *Transmitted Data*, digunakan DTE mengirimkan data ke DCE.
4. *Data Terminal Ready*, pada saluran ini DTE memberitahukan kesiapan terminalnya.
5. *Signal Ground*, saluran ground.
6. *DCE Ready*, sinyal aktif pada saluran ini menunjukkan bahwa DCE sudah siap.
7. *Request To Send*, dengan saluran ini DCE diminta mengirim data oleh DTE.
8. *Clear To Send*, dengan saluran ini DCE memberitahukan bahwa DTE boleh mulai mengirim data.
9. *Ring Indicator*, pada saluran ini DCE memberitahukan ke DTE bahwa sebuah stasiun menghendaki berhubungan dengannya.

## 2.9 Short Message Service (SMS)

SMS merupakan salah satu layanan *data service* GSM yang menyediakan fasilitas untuk menyampaikan pesan singkat antara *Mobile Station* (MS) melalui *Service Centre* (SC) atau yang disebut dengan *Short Message Service Centre* (SMSC). SMSC berfungsi menyampaikan pesan antara MS dan *Short Message Entity* (SME).

SMS adalah pesan yang terbatas besarnya (pesan singkat) yang dapat dikirim atau diterima oleh sebuah MS yang berupa data dalam bentuk *string* atau tesk. Tesk tersebut dapat terdiri dari kombinasi kata-kata, nomor-nomor atau penggaubungan huruf dan angka, yang mempunyai panjang maksimum 160 karakter dengan menggunakan huruf latin.

Dalam mengirim dan menerima data sms, digunakan standar ETSIGSM 03.38, dimana data yang tertulis dilayar *handphone* berupa huruf atau angka yang kemudian diterjemahkan menjadi data 7-*bit* (*septet*). Data 7 bit tersebut menurut standar ETSIGSM 03.38 diolah dulu menjadi data 8 bit (*octet*), baru dikirimkan, dimana besarnya data 7 bit maksimum adalah 160 karakter yang jika diolah menjadi data *octet* akan menjadi 140 *octet*.

## 2.10 AT Command

AT Command merupakan satu set perintah yang dapat digunakan untuk mengoperasikan GSM *mobile phone* melalui serial *interface* (kabel data). Menurut formatnya perintah-perintah yang diberikan harus selalu diawali dengan ’AT’ dan diakhiri dengan ’<CR>’ yang merupakan *Carriage Return*.

AT Command dapat digunakan antara lain untuk menginstruksikan perintah-perintah sebagai berikut :

1. Mengirim dan menerima SMS.
2. Mendapatkan informasi mengenai alat, misalnya nama manufaktur, nomor IMEI dan lain-lain.
3. Mendapatkan status alat, misalnya status aktivitas, status regestrasi *network*, kekuatan sinyal dan status batrei.
4. Penulisan dan pencarian *phonebook*.
5. Mengembalikan dan menyimpan konfigurasi.

Tidak semua alat mengimplementasikan AT Command, pada umumnya modem GSM lebih mendukung banyak AT Command. AT Command umumnya ditulis dengan huruf besar, tetapi banyak pila modem GSM dan ponsel yang mengizinkan penulisan AT Command dalam huruf besar maupun huruf kecil.

Terdapat dua tipe AT Command, yaitu *basic command* dan *extended command*. *Basic command* adalah AT Command yang tidak menggunakan tanda '+', misalnya ATH, ATD dan ATA. Sebaliknya AT Command yang menggunakan tanda '+' merupakan *extendedcommand*. Setiap *extended command* memiliki paremeter atau perintah tes yaitu =?, yang akan mengembalikan informasi mengenai perintah tersebut.

Beberapa perintah AT *Command* yang digunakan untuk keperluan SMS (pengiriman, penerimaan) adalah sebagai berikut :

1. AT+CMGS

Perintah AT *Command* ini digunakan untuk mengirimkan SMS. Format yang digunakan adalah "AT+CMGS = <*length*> <CR> <PDU *is given*>". Apabila pengiriman sukses dilakukan, format respon yang diterima adalah "+CMGS : <mr>", dengan "<mr>" adalah *message reference* dari SMSC. Sedangkan jika pengiriman gagal dilakukan, respon yang diterima adalah "+*CMS error*".

2. AT+CMGR

Perintah ini digunakan untuk membaca sebuah SMS pada indeks tertentu. Format yang digunakan adalah "AT+CMGR = <*index*>". Apabila perintah ini berhasil diesekusi, format respon yang diterima adalah "+CMGR: <*stat*>,,<*length*><CR><LF><pdu>". "<*stat*>".

3. AT+CMGD

Perintah ini digunakan menghapus sebuah SMS pada *memory* SMS. Format yang digunakan adalah "AT=CMGD=<*index*>", respon yang diterima adalah "*OK/ERROR/*+CMS *ERROR* "

4. AT+CMGL

Perintah ini digunakan untuk membaca daftar SMS sesuai parameter tertentu. Format yang digunakan adalah "AT+CMGL [=<*stat*>]".

Respon yang diterima adalah :

“+CMGL:*<index>,<stat>,<oa/da>,[<alpha>],[<scts>][,<tooa/toda>,<length>]<*

*CR><LF><data>[<CR><LF>*

*+CMGL:*

*<index>,<stat>,<da/oa>,[<alpha>],[<scts>][,<tooa/toda>,<length>]<*

*CR><LF>*

*<data>[...]] OK*”

Atau “*+CMS ERROR : <err>*”. “*[<alpha>]*”, adalah deretan alfanumerik yang merepresentasikan nomor pengirim atau penerima.

Yang terpenting adalah Anda mengetahui perintah AT Command untuk terima, kirim, dan delete SMS. Kemudian perintah tersebut dimasukkan ke dalam coding program yang anda buat.

**Tabel 2.3** adalah daftar perintah dalam AT Command.

| Jenis AT Command | AT Command | Fungsi |
|---|---|---|
| Untuk Kontrol | ATD | Melakukan dial |
| | ATH | Hang up panggilan masuk |
| | ATA | Menjawab panggilan |
| Bersifat Umum | AT+CGMI | Mengambil informasi manufaktur |
| | AT+CGMM | Mengambil informasi model |
| | AT+CGMR | Mengambil informasi revisi |
| | AT+CGSN | Mengambil informasi serial number alat |
| | AT+CSCS | Memilih set karakter |
| Untuk Layanan Jaringan | AT+CNUM | Nomor subscriber |
| | AT+CREG | Registrasi jaringan |
| | AT+COPS | Pemilihan operator |
| | AT+CLCK | Fasilitas lock |
| | AT+CPWD | Penggantian password |
| | AT+CCWA | Call waiting |
| Untuk Layanan Jaringan | AT+CPMS | Menentukan penyimpanan pesan |
| | AT+CMGF | Format pesan |
| | AT+CSCA | Nomor service provider |
| | AT+CNMI | Pengaktifan indikasi pesan baru |
| | AT+CMGL | Daftar pesan |
| | AT+CMGR | Membaca pesan |
| | AT+CMGS | Mengirim pesan |
| | AT+CMGD | Menghapus pesan |
| | AT+CMMS | Mengirimkan lebih banyak pesan |
| | +CMTI | Notifikasi delivery pesan |
| | +CDSI | Notifikasi status repot |

## 2.11 Pemrograman

Agar komputer dapat bekerja sesuai dengan yang kita inginkan, maka komputer harus diberikan program sehingga komputer yang terdiri dari perangkat keras ini mampu membantu aktivitas pemakainya. Program adalah sekumpulan instruksi-instruksi yang disimpan dalam memori dan kemudian dibaca, diterjemahkan serta dilaksanakan oleh mikroprosesor.

### 2.11.1 Borland Delphi

Delphi adalah suatu bahasa pemograman (development language) yang digunakan untuk merancang suatu aplikasi program.

Delphi termasuk dalam pemrograman bahasa tingkat tinggi (*high level lenguage*). Maksud dari bahasa tingkat tinggi yaitu perintah-perintahnya programnya menggunakan bahasa yang mudah dipahami oleh manusia. Bahasa pemrograman Delphi disebut bahasa prosedural artinya mengikuti urutan tertentu. Dalam membuat aplikasi perintah-perintah, Delphi menggunakan lingkungan pemrograman visual.

Delphi merupakan generasi penerus dari Turbo Pascal. Pemrograman Delphi dirancang untuk beroperasi dibawah sistem operasi Windows. Program ini mempunyai beberapa keunggulan, yaitu produktivitas, kualitas, pengembangan perangkat lunak, pola desain yang menarik serta diperkuat dengan bahasa perograman yang terstruktur dalam struktur bahasa perograman *Object Pascal.*

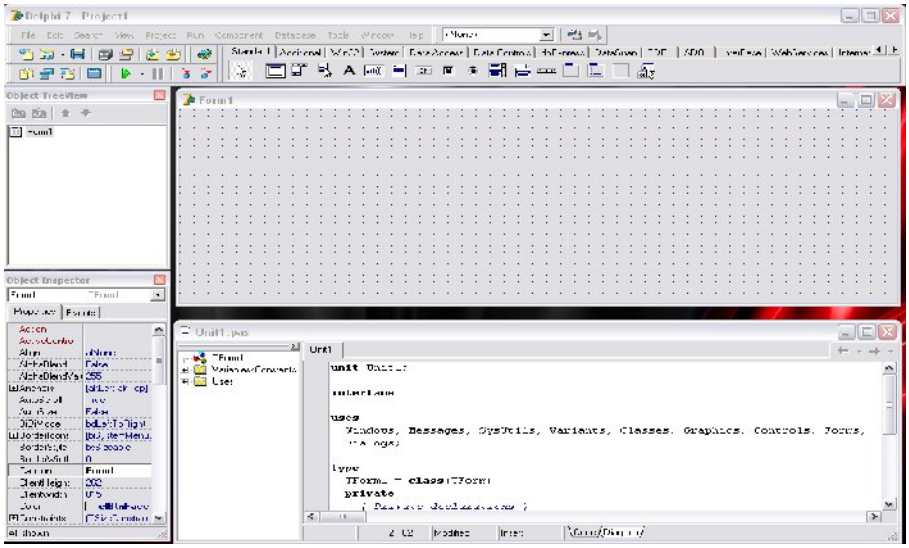
Sebagaian besar pengembang Delphi menuliskan dan mengkompilasi kode program di dalam lingkungan pengembang aplikasi atau *Integrated Development Environment* (IDE). Lingkungan kerja IDE ini menyediakan sarana yang diperlukan untuk merancang, membangun, mencoba, mencari atau melacak kesalahan, serta mendistribusikan aplikasi. Sarana-sarana inilah yang memungkinkan pembuatan prototipe aplikasi menjadi lebih mudah dan waktu yang diperlukan untuk mengembangkan aplikasi menjadi lebih singkat.

### 2.11.2 Beberapa keunggulan dari program Delphi adalah :

1. IDE (Integrated Development Environment) atau lingkungan pengembangan aplikasi sendiri adalah satu dari beberapa keunggulan delphi, didalamnya terdapat menu – menu yang memudahkan kita untuk membuat suatu proyek program.
2. Proses Kompilasi cepat, pada saat aplikasi yang kita buat dijalankan pada Delphi, maka secara otomatis akan dibaca sebagai sebuah program, tanpa dijalankan terpisah.
3. Mudah digunakan, source kode delphi yang merupakan turunan dari pascal, sehingga tidak diperlukan suatu penyesuain lagi.
4. Bersifat multi *purphase*, artinya bahasa pemograman Delphi dapat digunakan untuk mengembangkan berbagai keperluan pengembangan aplikasi.

### 2.11.3 Tampilan Delphi

Berikut adalah gambar dari keseluruhan tampilan Delphi.



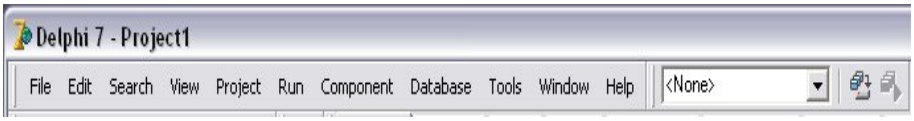
**Gambar 2.15** Tampilan Delphi

### 2.11.4 IDE (Integrated Development Environment)

Kemudian, hal yang paling pertama dalam mengenal Delphi adalah harus mengetahui IDE. IDE (Integrated Development Environment) merupakan lingkungan/wilayah dimana seluruh *tools* atau komponen-komponen yang dibutuhkan untuk merancang atau membangun aplikasi program. Secara umum IDE Delphi di kelompokkan kepada 8 bagian yaitu :

1. Main Menu

   Merupakan penunjuk ke seluruh fasilitas yang disediakan aplikasi Delphi.




**Gambar 2.16** Menu Pemrograman Delphi

2. *Toolbar / Speedbar*

   Merupakan Icon (Sortcut) yang dirancang untuk lebih memudahkan menjangkau fasilitas yang ada pada Delphi.




**Gambar 2.17** Toolbar Pemrograman Delphi

3. Component Palette

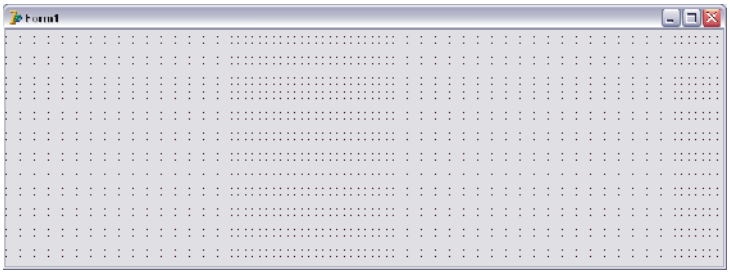
   Merupakan komponen-komponen VCL(Visual Component Library) yang dikelompokkan kedalam tab-tab, komponen komponen inilah yang akan digunakan untuk merancang interface atau antar muka aplikasi.




**Gambar 2.18** Component Palette Pemrograman Delphi

4. Form Designer

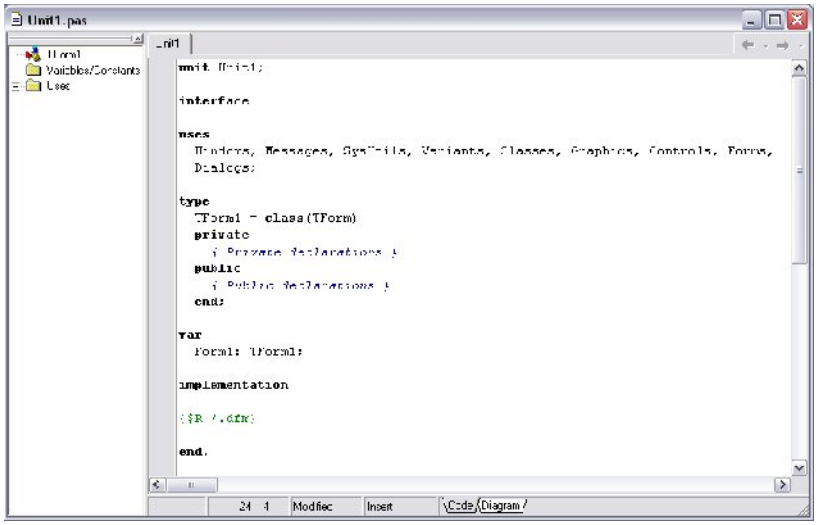
   Merupakan interface (antar muka) apalikasi yang akan dibangun, Form akan menampung seluruh komponen yang akan digunakan dalam proses perancangan sebuah aplikasi dengan Delphi.




**Gambar 2.19** Form Designer Pemrograman Delphi

5. Code Editor

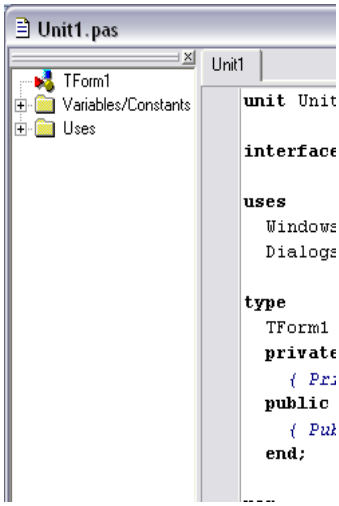
   Code editor merupakan tempat untuk menuliskan kode program menggunakan bahasa object Pascal. Kode program tidak perlu di tulis secara keseluruhan karena Delphi sudah menyediakan blok atau kerangka untuk menulis kode program.




**Gambar 2.20** Code Editor Pemrograman Delphi

6. Code Explorer

Digunakan untuk memudahkan berpindah antar file unit di dalam jendela code editor. Code explorer berisi daftar yang menampilkan semua tipe,class, properti , method, variabel global, rutin global yang telah didefinisikan di dalam unit. Saat memilih sebuah item dalam code explorer, kursor akan berpindah menuju implementasi dari item yang dipilih di dalam code editor.




**Gambar 2.21** Code Explorer Pemrograman Delphi

7. Object inspector

   Object inspector digunakan untuk mengubah properti atau karakteristik dari suatu komponen. Terdiri dari 2 tab yaitu :

   a. Properties

      Digunakan untuk menentukan seting suatu objek. Satu objek memiliki beberapa properti yang dapat diatur langsung dari object inspector maupun melalui kode program. Seting ini mempengaruhi cara kerja objek tersebut saat aplikasi dijalankan.

   b. Event

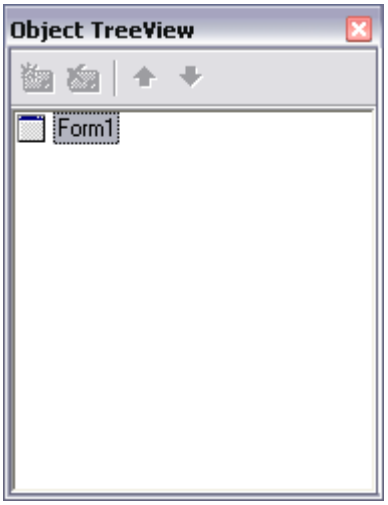
      Merupakan bagian yang dapat diisi dengan kode program tertentu yang berfungsi untuk menangani event-event (berupa sebuah procedure) yang dapat direspon oleh sebuah komponen.Event adalah peristiwa atau kejadian yang diterima oleh suatu objek, misal : klik, drag, dan lain-lain. Event yang diterima objek akan memicu Delphi menjalankan kode program yang ada didalamnya. Misalnya ingin sesuatu dikerjakan pada saat form ditutup, maka untuk menyatakan tindakan tersebut (berupa sebuah procedure) menggunakan *OnClose***.**




**Gambar 2.22** Object Inspector Pemrograman Delphi

8. Object Tree View

Object tree view berisi daftar komponen yang sudah diletakkan di form designer.




**Gambar 2.23** Object Tree ViewPemrograman Delphi

### 2.11.5 XCOMM

XComm adalah komponen tambahan seperti MSComm dengan fungsi yang sama, XComm ini digunakan untuk komunikasi antara computer PC (*personal computer*) dengan *handphone* sebagai penerima SMS. XComm inilah yang mengatur dan membaca jika ada SMS masuk atau keluar dari *handphone* yang selanjut nya dikomunikasikan ke PC.

### 2.11.6 QCOMM

QComm adalah komponen tambahan yang terdapat pada Borland Delphi yang menyediakan fasilitas komunikasi data melalui serial port, sehingga dapat berkomunikasi dengan rangkaian elektronik mikrokontroler. MSComm inilah yang menjadi jembatan penghubung antara computer dengan mikrokontroler