BAB II
LANDASAN TEORI

2.1. Dasar Komunikasi

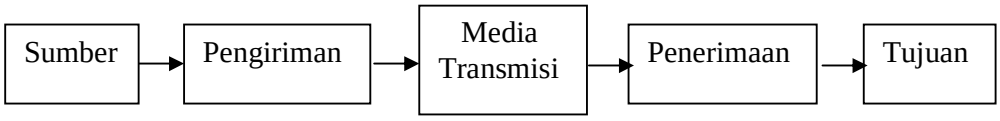
Komunikasi adalah suatu proses penyampaian berita dari suatu pihak ke pihak lain dengan mempergunakan suatu sarana untuk mendapatkan saling pengertian antara kedua belah pihak.

Proses yang mendasar dalam komunikasi adalah penggunaan bersama atau dengan kata lain ada yang memberi informasi (mengirim) dan ada yang menerima informasi. Penggunaan bersama disini tidak harus yang memberi dan yang menerima harus saling berhadapan secara langsung akan tetapi bisa melalui media lain, seperti tulisan, isyarat, maupun yang berupa kode-kode tertentu yang bisa dipahami.

Kesimpulannya, bahwa dalam proses komunikasi, pihak-pihak peserta dalam komunikasi menciptakan pesan-pesan yang berupa informasi bisa berbentuk pola, isyarat ataupun simbol, dengan harapan akan mengutarakan suatu makna tertentu bagi peserta-peserta lain (penerima).

2.1.1. Komunikasi data

Komunikasi data adalah proses pertukaran data atau pengiriman data dari sumber ke tujuan. Hal yang penting dalam melakukan komunikasi data adalah jenis komunikasi yang digunakan, apakah menggunakan kabel, nirkabel (infrared,frekuensi radio) atau menggunakan frekuensi radio. Secara umum diagram blok komunikasi data sederhana seperti pada Gambar 2.1.



Gambar 2.1. Blok diagram model komunikasi sederhana

Berikut penjelasan dari blok diagram di atas :

1. Sumber

   Masukan data atau informasi yang akan dikirimkan ke tujuan.
2. Pengiriman

   Pengiriman data dari sumber melalui media transmisi.
3. Media Transmisi

   Jalur transmisi yang menghubungkan antara sumber dengan tujuan.

4. Penerimaan

   Penerima sinyal yang dikirimkan melalui media transmisi untuk kemudian dikirimkan ke tujuan.

5. Tujuan

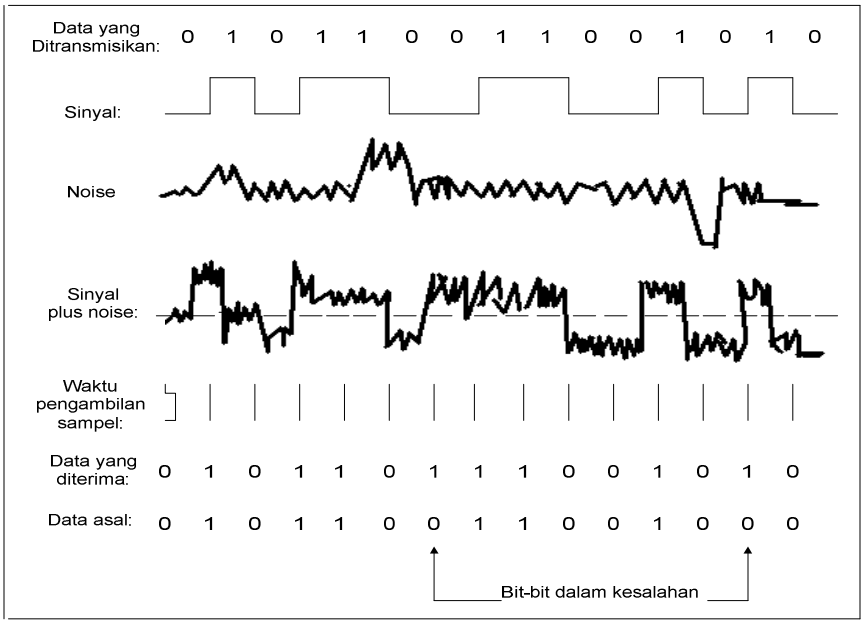
   Menampilkan hasil data yang dikirim oleh sumber.

Dalam komunikasi data jenis komunikasi ada 3 macam yakni :

a. *Simplex*, yaitu komunikasi data hanya dengan satu arah dari sumber ke tujuan.
b. *Half Duplex*, yaitu komunikasi data dengan dua arah tetapi tidak bisa melakukan komunikasi secara bersamaan.
c. *Full Duflex*, yaitu komunikasi data dengan dua arah dan bisa melakukan komunikasi secara bersamaan.

2.1.2. Kapasitas Kanal

Kapasitas kanal adalah rata-rata maksimum pada data yang dapat ditransmisikan melalui suatu jalur komunikasai tertentu, atau pada kondisi tertentu.



Gambar 2.2. Efek Noise dalam Sebuah Sinyal Digital

Disini terdapat empat konsep di mana kita berusaha saling menghubungi satu sama lain.

1. *Data rate*: Merupakan rata-rata dalam bit per detik (bps), pada data yang dapat dikomunikasikan.

2. *Bandwidth*: Merupakan lebar atau jalur dari sinyal yang ditransmisikan saat melalui transmitter dan sifat media transmisi, dinyatakan dalam siklus per detik, atau Hertz.
3. *Derau*: Merupakan level rata-rata sinyal yang tidak diinginkan sepanjang jalur komunikasi
4. *Error rate*: Merupakan rata-rata error yang terjadi, yaitu suatu error diterima sebesar 1 saat 0 ditransmisikan atau penerimaan sebesar 0 saat 1 ditransmisikan.

Permasalahan yang akan dibahas adalah fasilitas-fasilitas komunikasi yang mahal, dan umumnya bandwidth fasilitas yang lebih besar berarti biaya yang lebih besar pula. Selanjutnya, kanal transmisi dari kepentingan tertentu terbatas oleh bandwidth. Batasan-batasan tersebut muncul dari sifat-sifat fisik media transmisi atau dari pembatasan-pembatasan yang disengaja pada pengirim terhadap bandwidth untuk mencegah gangguan dari sumber-sumber lain. Karena itu, dalam membuat penggunaan bandwidth tertentu seefisien mungkin. Untuk data digital, hal ini berarti bahwa kita akan mendapatkan *data rate* setinggi mungkin pada batas *error rate* tertentu untuk bandwidth tertentu juga. Penghalang utama untuk mencapai tingkat efisiensi seperti itu adalah derau.

2.2. Teknik Komunikasi Data Digital

Salah satu tujuan komunikasi data adalah untuk mengirimkan data secara utuh dari sumber data hingga sampai ke penerima atau tujuan pengiriman. Data utuh diterima, berarti bahwa data tersebut lengkap tidak *corrupt* atau hilang pada saat pengiriman. Untuk menjaga dan meyakinkan bahwa data yang sedang dikirim akan tiba dengan selamat dan utuh ke tangan penerima itulah dilakukan pendeteksian kesalahan dan pembetulan kembali data jika ternyata ada yang salah.

Kesalahan dalam transmisi data, dapat terjadi karena gangguan-gangguan pada saluran, *system switching*, radiasi gelombang, *cross talk*, dan lain-lain. Ada beberapa kemungkinan terjadi kesalahan dalam pengiriman pada frame-frame data atau blok dari setiap paket pengiriman sebagai berikut:

a. Error bit tunggal yang disebut dengan *bit error rate*
b. Frame yang diterima tanpa adanya error
c. Frame diterima dengan error tetapi tidak terdeteksi
d. Frame diterima dengan error tetapi terdeteksi

Jika dalam suatu jaringan komunikasi data tidak dipasang suatu alat pendeteksian error, maka akan berlaku bahwa probabilitas probabilitas error yang terdeteksi = 0 atau P3 = 0. Demikian juga dengan probabilitas frame yang diterima tanpa error (P1) akan bertambah jika probabilitas dari error bit tunggal bertambah.

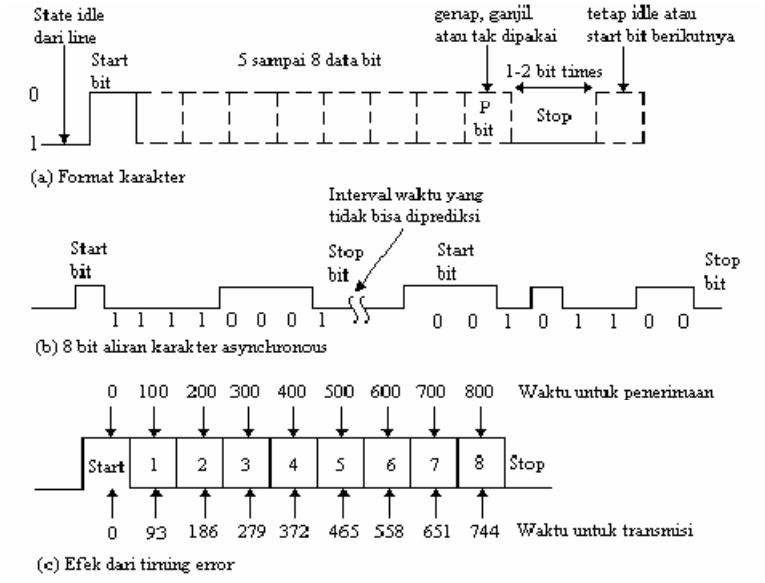
2.2.1. Sinkronisasi

Sinkronisasi adalah satu kunci kerja dari komunikasi data. Pengirim mengirimkan pesan 1 bit pada satu saat melalui medium ke penerima.Penerima harus menandai awal dan akhir blok dari bit, juga harus diketahui durasi untuk masing-masing bit sehingga dapat sample lajur dari timing untuk membaca masing-masing bit (merupakan tugas dari timming). Contoh : jika ada perbedaan misalkan 1 % (clock receiver 1% lebih lambat atau lebih cepat daripada clock transmitter), maka pada pensamplingan pertama akan meleset dari tengah bit dan setelah jumlah waktu tertentu, akan mengalami error.

Sinkroinisasi terbagi dua:

1. *Asynchronous*

   Untuk mencegah problem *timming* dengan tidak mengirim aliran bit panjang yang tidak putusputusnya. Bit-bit dikirim per-karakter pada setiap waktu dimana masing-masing karakter mempunyai panjang 5-8 bit. Timing atau synchronisasi harus dipertahankan antara tiap karakter; receiver mempunyai kesempatan untuk mensinkronkan awal dari tiap karakter baru.



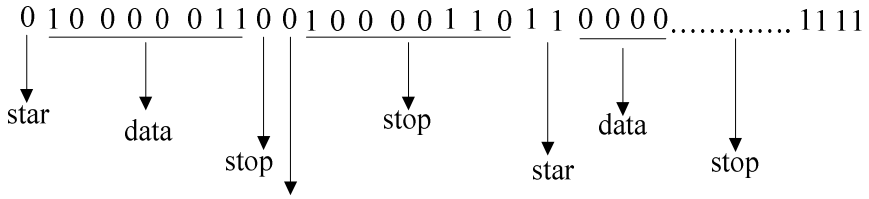
Gambar 2.3. Transmisi Asynchronous

Keterangan Gambar 2.3:

a. Idle (biasanya =’1’) jika tidak ada karakter yang ditransmisikan dan start bit = “0”, sedangkan jumlah karakter yang ditransmisikan antara 5-8 bit.
b. Bit paritas digunakan untuk mendeteksi error, diatur oleh pengirim agar jumlah total ‘1’ termasuk bit paritas adalah genap, dan stop bit = ‘1’, yang panjangnya 1; 1,5; 2 kali durasi bit pada umumnya.
c. Komunikasi asinkron adalah sederhana dan murah, tetapi memerlukan tambahan dari 2 ke 3 bit per karakter, prosentasi overhead dapat dikurangi dengan mengirimkan blok-blok bit besar antara bit start dan bit stop Contoh : akan dikirimkan data ASCII ABC dengan A = 41H, B = 42H dan C = 43H tanpa paritas, maka :

$A = 0100\ 0001_2$ invert kode ASCII 7 bit
$100\ 0001_2$
$B = 0100\ 0010_2$ invert kode ASCII 7 bit
$010\ 0001_2$
$C = 0100\ 0011_2$ invert kode ASCII 7 bit
$110\ 0001_2$

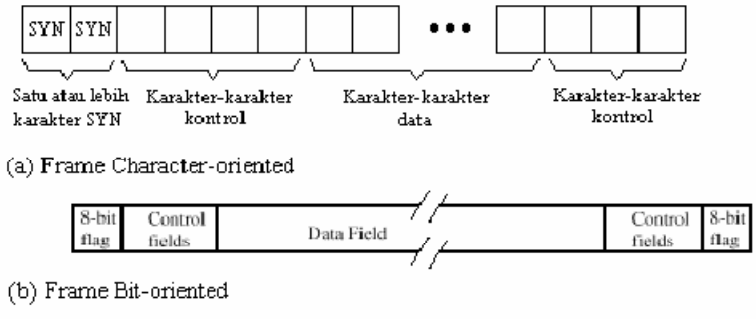
Datanya terdiri dari kode ASCII 7 bit dan bit stop, maka :



Gambar 2.4. Kode ASCII 7 bit dan bit stop

Kode 7 bit memakai panjang 1 bit untuk bit start dan 1 bit untuk bit stop, maka overheadnya 2 / 9 = 0,22.

2. Synchronous



Gambar 2.5. *Synchronous*

Keuntungan dari synchronous:

a. Efisien, karena blok-blok karakter / bit-bit ditransmisikan tanpa kode start dan stop, tetapi tiap blok blok dimulai dengan suatu pola *preamble* bit dan diakhiri dengan pola *postamble* bit. Pola-pola ini adalah control informasi.
b. Waktu kedatangan dan keberangkatan untuk masing-masing bit dapat diramalkan.
c. *Frame* adalah data plus kontrol informasi. Format framenya tergantung dari metode transmisi, yaitu:

- Transmisi orientasi karakter
  1. Blok-blok data dikerjakan sebagai barisan karakter (biasanya 8 bit karakter), frame dimulai dengan 1 atau lebih karakter sinkronisasi. Karakter sinkronisasi biasanya disebut dengan "SYN" yang merupakan bit pattern unik sinyal yang diterima penerima permulaan dari blok.
  2. Penerima kemudian merubah blok-blok data yang datang oleh karakter SYN dan menerima data sampai karakter *postamble* (informasi yang terletak pada bagian belakang blok data yang dikirimkan) terlihat dan begitu seterusnya

- Transmisi bit.
  1. Blok-blok data dikerjakan sebagai barisan bit-bit, tidak ada data maupun informasi kontrol diperlukan untuk menginterprestasikan dalam satuan karakter 8 bit
  2. Pada awal terdapat flag, begitu juga pada akhir yang panjangnya 8 bit yang berguna sebagai awal dan akhir untuk penerima

3. Perbandingan asinkron dan sinkron
   a. Untuk blok-blok data yang cukup besar, transmisi sinkronisasi jauh lebih efisien daripada asinkron. Transmisi asinkron memerlukan overhead 20 % atau lebih.
   b. Bila menggunakan transmisi sinkron biasanya lebih kecil dari 1000 bit, yang mengandung 48 bit kontrol informasi (termasuk flag), maka untuk pesan 1000 bit, overheadnya adalah 48 / 1048 X 100% = 4.6%
4. Urutan pengerjaan sinkronisasi yaitu
   a. Sinkronisasi bit
      Ditandai awal & akhir untuk masing-masing bit
   b. Sinkronisasi karakter / kata

Ditandai awal dan akhir untuk masing-masing karakter / satuan kecil lainnya dari data

c. Sinkronisasi blok / pesan

Ditandai awal dan akhir dari satuan besar data.Dan untuk pesan yang besar, dibagi-bagi menjadi beberapa blok kemudian baru dikirimkan pengurutan blok-blok yang telah dibagi tersebut adalah tugas dari timming. Sedangkan pengaturan level sinyal adalah tugas dari sintax dan untuk melihat arti dari pesan adalah tugas dari semantik.

Deteksi error dengan Redundansi, yaitu data tambahan yang tidak ada hubungannya dengan isi informasi yang dikirimkan, berupa bit pariti. Berfungsi menunjukkan ada tidaknya kesalahan data. Yaitu dengan mendeteksi dan mengoreksi kesalahan yang terjadi. Makin banyak redundansi makin baik deteksi errornya. Akibatnya makin rendah troughput dari data yang berguna. Troughput adalah perbandingan antara data yang berguna dengan data keseluruhan. Banyaknya tambahan pada redundansi sampai 100% dari jumlah bit data.

2.2.2. Teknik mendeteksi error

Teknik deteksi error menggunakan *errordetecting- code*, yaitu tambahan bit yang ditambah oleh transmitter. Dihitung sebagai suatu fungsi dari transmisi bit-bit lain. Pada receiver dilakukan perhitungan yang sama dan membandingkan kedua hasil tersebut, dan bila tidak cocok maka berarti terjadi deteksi error. Apabila sebuah frame ditransmisikan ada 3 kemungkinan klas yang dapat didefinisikan pada penerima, yaitu :

1. Klas 1 (P1) : Sebuah frame datang dengan tidak ada bit error (*jadi tidak berarti dalam mendeteksi error, karena tidak ada error!*)
2. Klas 2 (P2) : Sebuah frame datang dengan 1 atau lebih bit error yang tidak terdeteksi
3. Klas 3 (P3) : Sebuah frame datang dengan 1 atau lebih bit error yang terdeteksi dan tidak ada bit error yang tidak terdeteksi. (*tidak berarti juga, semua error sudah terdeteksi*)

Ada dua pendekatan untuk deteksi kesalahan:

1. Forward Error Control

Yaitu setiap karakter yang ditransmisikan atau frame berisi informasi tambahan (redundant) sehingga bila penerima tidak hanya dapat mendeteksi

dimana error terjadi, tetapi juga menjelaskan dimana aliran bit yang diterima error.

2. *Feedback (backward) Error Control*

   Yaitu setiap karakter atau frame memiliki informasi yang cukup untuk memperbolehkan penerima mendeteksi bila menemukan kesalahan tetapi tidak lokasinya. Sebuah transmisi control digunakan untuk meminta pengiriman ulang, menyalin informasi yang dikirimkan.

Feedback error control dibagi menjadi dua bagian yaitu:

1. Teknik yang digunakan untuk deteksi kesalahan.
2. Algoritma control yang telah disediakan untuk mengontrol transmisi ulang.

2.3. Metode Deteksi Kesalahan

1. *Echo*

   Metode sederhana dengan sistem interaktif operator memasukkan data melalui terminal dan mengirimkan ke komputer akan menampilkan kembali ke terminal, sehingga dapat memeriksa apakah data yang dikirimkan dengan benar.

2. *Error Otomatis / Parity Check*

   Penambahan parity bit untuk akhir masing-masing kata dalam *frame*. Tetapi problem dari parity bit adalah inpuls noise yang cukup panjang merusak lebih dari satu bit, pada data rate yang tinggi.

2.3.1. Jenis *Parity Check*

a. *Even parity* (paritas genap), digunakan transmisi asynchronous. Bit parity ditambahkan supaya banyaknya ‘0’ untuk tiap karakter adalah genap.

b. *Odd parity* (paritas ganjil), digunakan untuk transmisi synchronous. Bit parity ditambahkan supaya banyaknya ‘1’ untuk tiap karakter adalah ganjil.

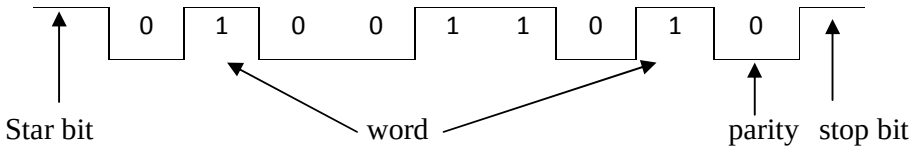
2.3.1.1.*Vertical Redundancy Check*

Pada metode ini, dalam satu byte terdapat satu bit parity. Yang diletakan setelah bit ke tujuh dan menjadi bit ke delapan. Seperti karakter 10101000 akan menjadi 110101000 atau 010101000.

Bit paritas yang digunakan supaya cacah 1 pada setiap karakter berjumlah ganjil atau bertambah genap, yang berjumlah ganjil disebut dengan nama paritas ganjil (*odd parity*), dan yang berjumlah genap disebut dengan nama paritas genap (*even parity*).

Nilai dari bit tergantung dari jumlah ganjil atau genapnya jumlah bit satu dalam tiap byte. Aturan yang berlaku pada odd parity adalah bahwa jumlah bit satu dalam setiap byte harus ganjil. Program yang di buat akan selalu melakukan pengecekan terhadap parity bit dari setiap karakter yang dikirim. Bila jumlah bit satu dari tujuh bit pertama adalah genap, maka bit paritas di ubah menjadi 1 dan sebaliknya bila jumlah bit satu dari tujuh bit pertama adalah ganjil,maka bit paritas diubah menjadi 0.

Sedangkan pada even bit parity berlaku, yaitu bahwa jumlah bit satu dalam setiap byte harus genap. Sebagai contoh, jika huruf “M” di susun dalam kode biner adalah “1001101” dimana 7 bit pertama jumlah bit satunya adalah genap.maka parity bit diubah menjadi 0 seperti pada contoh berikut:



Gambar 2.6. Contoh Karakter ’M' dengan even parity

Salah satu kelemahan dalam bit paritas Vertical Redundancy Check ini adalah sulitnya melakukan deteksi terhadap kesalahan jika jumlah bit error adalah genap.

Jika terjadi kesalahan dalam suatu pengiriman, maka bit yang diterima menjadi tidak sesuai dengan pengiriman. Misalnya diawal pengiriman berjumlah genap, tetapi tiba-tiba berjumlah ganjil. Berarti ada gangguan transmisi. Tetapi ada 2 atau 4 bit yang salah, menyulitkan pendeteksian error karena jumlah ganjil genapnya bit sama dengan jumlah ganjil genap bit sebelumnya.

2.3.1.2 .*Longitudinal Redundancy Checks*

LRC ini memperbaiki kekurangan yang terjadi pada VRC. Pengiriman data dilakukan per blok. Setiap blok terdiri dari 8 byte. Dan setiap blok memiliki *block check character* atau karakter pemeriksa blok yang diletakan pada akhir blok.

Block check character ini memuat 7 parity bit dari bit sebelumnya. Sedangkan untuk mengubah nilai ketujuh dari bit ini adalah dengan melihat jumlah bit 1 dari seluruh isi byte secara vertical

Tabel 2.1 Contoh Pembentukan Block Check Character

Nomor bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	0	1	0	0	0	0	0	1
	1	0	1	0	0	0	0	0
	0	1	0	0	0	1	1	1
	1	1	0	0	1	1	1	1
Karakter	1	1	0	0	1	1	1	1
	1	0	1	0	0	0	0	0
	0	1	0	0	0	0	1	0
	0	1	0	0	0	1	1	1
	0	1	0	0	0	1	1	1
	0	1	0	0	1	0	1	1
Block check karakter	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**

Pada tabel 2.1 terlihat bagaimana block check character dibentuk. Pada nomor bit ke tujuh, jumlah angka 1 pada karakter adalah 4 (genap) maka block check character untuk bit ke tujuh adalah 0. Demikian juga untuk nomor bit ke enam, jumlah angka 1 pada karakter adalah 8 (genap) maka block check character untuk bit ke enam adalah 0. Sedangkan pada nomor bit ke dua, jumlah angka 1 pada karakter adalah 5 (ganjil) maka block check character untuk bit ke dua adalah 1. Seperti halnya pada no bit ke satu, jumlah angka 1 pada karakter adalah 7 (ganjil) maka block check character untuk bit ke satu adalah 1, demikian seterusnya.

Pada sisi penerima, setiap kolom dan setiap baris diperiksa. Lokasi kesalahan tunggal dapat ditemukan dengan melakukan interseksi pada kolom dan baris yang mengandung kesalahan.

2.3.1.3.*Cyclic Redundancy Checks*

Cyclic Redundancy Checks (CRC) merupakan sistem dengan penambahan *control bit* untuk menjamin keamanan data dan digunakan untuk pengiriman yang berkecepatan tinggi. Control bit dibentuk oleh komputer pengirim berdasarkan perhitungan atas data yang dikirim. Kemudian pada pada prinsipnya, ketika data sampai dikomputer penerima dilakukan perhitungan seperti yang dilakukan oleh

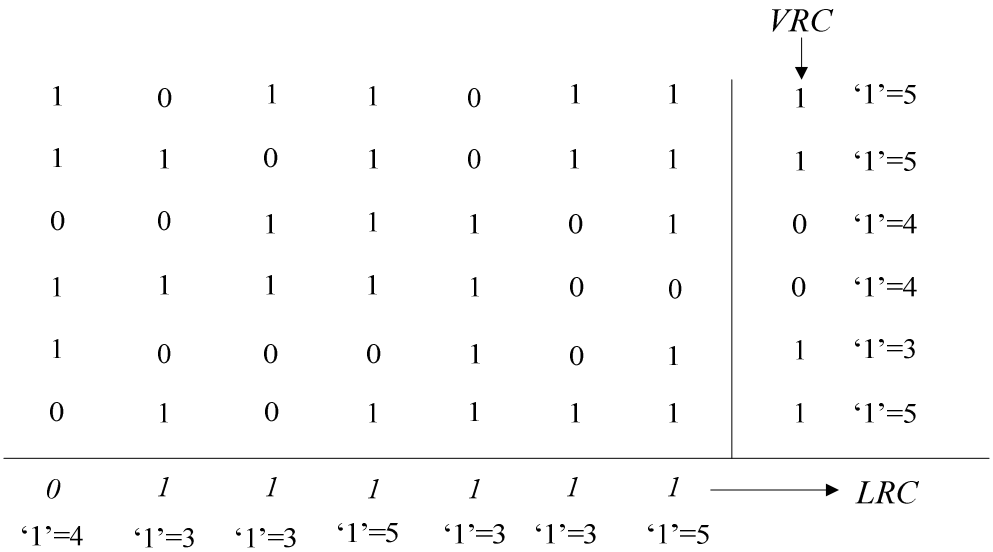
komputer pengirim. Jika hasil perhitungan sama, maka tidak ada kesalahan dalam pengiriman.

Pada CRC, data dikirim per frame dan setiap frame terdiri dari deretan yang panjang. Penggunaan CRC pada sambungan data sangat efisien untuk mendeteksi kesalahan. Bilangan biner yang digunakan sebagai pembagi disebut sebagi *generating polynomial* dan harus mempunyai satu bit lebih panjang dibandingkan dngan panjang CRC. Jika CRC memiliki panjang n = bit,maka bilangan biner data harus dikalikan dengan 2n yaitu bit 0 sebanyak n buah ditambah sesudah bit signifikan terkecil.

Untuk memahami lebih lanjut tentang konsep CRC, akan lebih baik jika memahami terlebih dahulu konsep pembagian modulo 2 serta konsep menjabarkan deretan bit menjadi polynomial aljabar.

2.3.1.4 .Menggunakan Paritas XOR

Pada paritas genap ini adalah perpaduan antara VRC dan LRC seperti pada gambar di bawah ini :



Gambar 2.7. Contoh penggunaan paritas XOR

2.3.1.5.Frame Check

Dipakai pada transmisi asinkron dengan adanya bit awal dan akhir. Data berada diantara bit awal dan bit akhir. Dengan memeriksa kedua bit ini dapat diketahui apakah data dapat diterima dengan baik atau tidak. Transmisi asinkron mempunyai bentuk bingkai sesuai dengan ketentuan yang dipergunakan.

Pendekatan yang umum dipakai adalah data link layer memecah aliran bit menjadi frameframe diskrit dan menghitung checksum setiap framenya. Ketika sebuah frame tiba di tujuan, checksum dihitung kembali. Bila hasil perhitungan ulang checksum tersebut berbeda dengan yang terdapat pada frame, maka data link layer akan mengetahui bahwa telah terjadi error dan segera akan mengambil langkah tertentu sehubungan dengan adanya error tersebut (misalnya, membuang frame yang buruk dan mengirimkan kembali laporan error).

Salah satu cara untuk melaksanakan pembuatan frame ini adalah dengan cara menyisipakn gap waktu di antara dua buah frame, sangat mirip seperti spasi antara dua buah katan dalam suatu teks. Akan tetapi, jaringan jarang memberikan jaminan tentang pewaktuan. Karena itu, mungkin saja gap ini dibuang, atau diisi oleh gap lainnya selama proses transmisi, karena sangat besar risikonya dalam menghitung pewaktuan untuk menandai awal dan akhir frame, telah dibuat metode lainnya, yaitu 4 buah metoda :

1. Karakter penghitung
2. Pemberian karakter awak dan akhir, denganpengisian karakter
3. Pemberian flag awal dan akhir, denganpengisian bit
4. Pelanggaran pengkodean physical layer.

Metoda framing pertama menggunakan sebuah field pada header untuk menspesifikasikan jumlah karakter di dalam frame. Ketika data link layer pada mesin yang dituju melihat karakter penghitung, maka data link layer akan mengetahui jumlah karakter yang mengikutinya, dan kemudian juga akan mengetahui posisi ujung frame-nya. Masalah yang dijumpai dalama algoritma ini adalah bahwa hitungan dapat dikacaukan oleh error transmisi. Misal bila hitungan karakter 5 frame menjadi 7, maka tempat yang dituju akan tidak sinkron dan tidak dapat mengetahui awal frame berikutnya.

2.4. Teori Dasar Gelombang

Gelombang adalah suatu gejala terjadinya penjalaran suatu gangguan melewati suatu medium, dimana setelah gangguan itu lewat keadaan medium akan kembali ke keadaan semula, seperti sebelum gangguan itu datang [Trisnobudi, 2000]. Medium merupakan sekumpulan benda yang saling berinteraksi dimana gangguan itu merambat. Berdasarkan medium perambatannya, gelombang dikelompokkan menjadi 2 yaitu:

1. Gelombang Mekanik

   Merupakan gelombang yang terjadi karena adanya gaya mekanik yang merambat dalam medium yang bersifat elastis, seperti gelombang bunyi, gelombang pada permukaan air dan gelombang pada tali dan ultrasonik

2. Gelombang Elektromagnetik

   Merupakan gelombang yang tidak memerlukan medium dalam perambatannya karena dapat merambat di ruang hampa dan perubahan yang diakibatkan bukanlah perubahan mekanik, seperti gelombang radio, sinar infra merah, dan sinar ultra violet.

2.5. Teknik Modulasi Digital

Teknik untuk pengkodean sinyal digital ke dalam sinyal analog disebut dengan Modulasi Digital. Beberapa teknik Modulasi Digital yang umum digunakan untuk data digital biner adalah:

a. Amplitudo Shift keying (ASK)

Pada cara ini, amplitudo gelombang pembawa diubah ubah sesuai dengan informasi yang ada.

Bentuk paling sederhana :

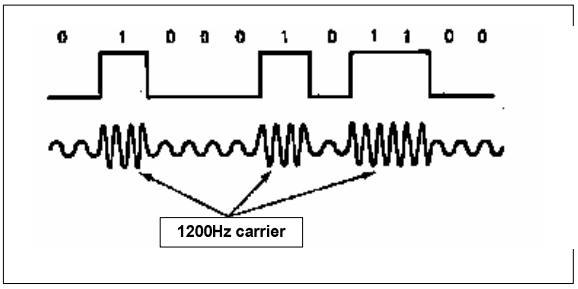
Bit ”0” : signal OFF

Bit “1” : signal ON

Karena informasi yang ada hanya terdiri dari dua jenis, maka akan terdapat dua buah tingkat besaran amplitudo yang berbeda. Tentu saja kedua amplitudo tersebut ada setelah proses pemodulasian.

Akan tetapi dapat juga menggunakan empat tingkat besaran amplitudo yang berbeda dengan menggunakan pasangan bit yaitu bit 00, bit 01, bit 10 dan bit 11. Dengan demikian dalam waktu yang sama dapat dikirimkan data dalam jumlah dua kali lipat. Selain itu dapat pula memperkecil lebar jalur yang dipergunakan.

Karena derau (noise) yang dialami oleh modulasi ASK cukup besar, maka sering kali data yang dikirimkan mengalami cacat. Sebab itulah teknik modulasi ASK jarang sekali diterapkan pada tranmisi data.



Gambar 2.8. Bentuk gelombang ASK

b. *Phase Shift keying (PSK)*

Teknik ini fase dari gelombang pembawa diubah sesuai dengan bit 1 dan bit 0.

bit "0" : 0^o

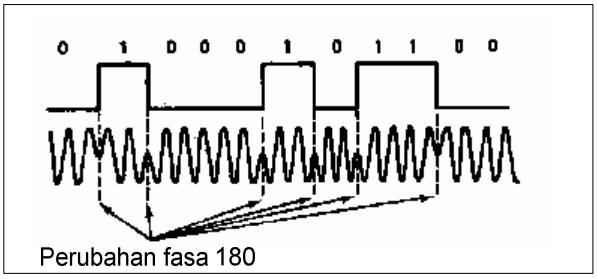
bit "1" : 180^o

Jika perubahan fasa sebesar 180^o, akan terdapat dua keadaan.

Jika perubahan fasa sebesar 90^o, akan terdapat empat keadaan.

Jika perubahan fasa sebesar 45^o, akan terdapat delapan keadaan.

Derau yang ditimbulkan PSK adalah terkecil dari pada ASK dan FSK. Biasanya PSK digunakan dalam transmisi yang mempunyai kecepatan sedang dan tinggi.



Gambar 2.9. Bentuk gelombang PSK

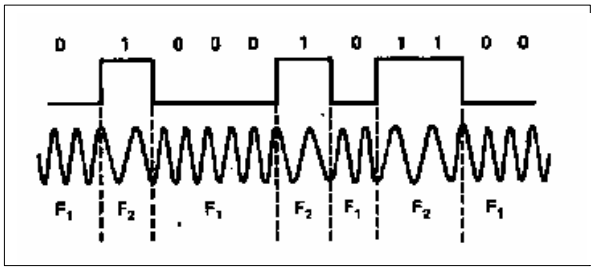
c. *Frekuqncy Shift Keying (FSK)*

Teknik ini mengubah frekuensi pembawa berdasarkan bit 1 dan bit 0

bit "0" : frekuensi fo

bit "1" : frekuensi f_1

Teknik modulasi ini banyak digunakan untuk tranmisi dengan kecepatan rendah. Derau yang didapat pada teknik FSK kelebihan kecil dibandingkan ASK.



Gambar 2.10. Bentuk gelombang FSK

2.5.1. Gangguan Transmisi

Dalam sistem komunikasi, sinyal yang diterima kemungkinan berbeda dengan sinyal yang ditransmisikan karena adanya gangguan transmisi. Untuk pengiriman sinyal analog terdapat gangguan yang dapat menurunkan kualitas sinyal, namun bagi pengiriman sinyal digital akan terdapat gangguan seperti *bit error*. Gangguan yang ada pada transmisi data yaitu :

1. Atenuasi dan Distorsi Atenuasi

   Kekuatan sinyal berkurang bila jaraknya terlalu jauh melalui media transmisi. Pada sinyal analog karena atenuasi berubah ubah sebagai fungsi frekuensi, sinyal diterima menjadi penyimpangan, sehingga mengurangi tingkat kejelasan.

2. Distorsi Tunda

   Distorsi tunda merupakan suatu kejadian khas pada *guided media*, kejadian ini disebabkan oleh sebuah sinyal yang melewati *guilded* berbeda frekuensi.

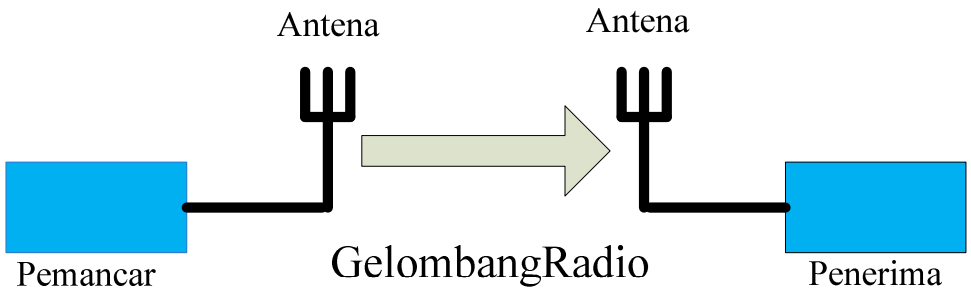
3. Noise

   Noise adalah sinyal-sinyal yang tidak diinginkan yang terselip atau terbangkitkan dari suatu tempat diantara transmisi dan penerimaan. Derau merupakan faktor utama yang membatasi kinerja sistem komunikasi.

2.5.2 Antena

Antena adalah salah satu elemen penting yang harus ada pada sebuah teleskop radio. Fungsinya adalah untuk mengubah sinyal listrik menjadi sinyal elektromagnetik, lalu meradiasikannya. Dan sebaliknya, antena juga dapat berfungsi untuk menerima sinyal elektromagnetik dan mengubahnya menjadi sinyal listrik. Pada radar atau sistem komunikasi satelit, sering dijumpai sebuah antena yang melakukan kedua fungsi (peradiasi dan penerima) sekaligus.Fungsi lainya juga sebagai radiator gelombang radio

(antena pemancar) dan penerima gelombang radio (antena penerima) dan antena juga mempunyai sifat *reciprocity*.

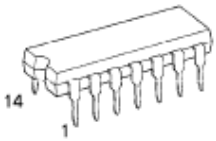


Gambar.2.11. Antena gelombang radio

2.6 Perangkat Keras Komunikasi Data yang di Pakai

2.6.1 IC Switch

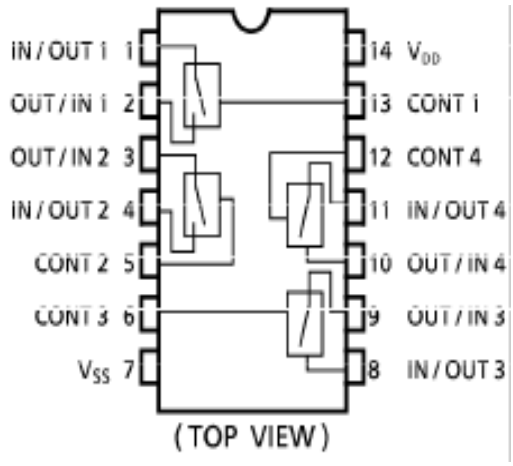
Penggunaan IC ini digunakan sebagai fungsi *switch,* dan IC ini merupakan IC yang berfungsi untuk *switch* transmisi atau mulltiplexing sinyal analog maupun sinyal digital. IC ini mempunyai kemampuan *switch* antara *crosstalk* sebesar 50 dB dan frekuensinya 0.9 Mhz resistansi dari IC ini memiliki keadaan tetap dalam keadaan pada saat signal-input range nya dalam keadaan penuh.



Gambar 2.12. IC 4066

IC 4066 memiliki 4 *switch* elektronik didalamnya, dan *switch* didalamnya memiliki fungsi yang sama, untuk setiap *switch* memiliki pin kontrol yang berbeda, untuk melewatkan sinyal pada *switch* A terdapat pin 13, *switch* B terdapat pada pin 5, *switch* C terdapat pada pin 6 dan *switch* D terdapat pada pin 6 konfigurasi pin lebih jelasnya terdapat pada gambar 2.13.

Pin-pin kontrol yang dijelaskan diatas merupakan pin pin yang akan mengaktifkan *switch* yang mana yang akan digunakan. *Switch* untuk memutus dan menyambungkan jalur. Cara melakukan *switch* adalah apabila pin kontrol diberi masukan low maka jalur in dan out tidak tersambung dan sebaliknya apabila diberi high maka jalur in dan out tersambung dan sebaliknya apabila diberi high maka jalur in dan out tersambung.



Gambar 2.13. konfigurasi pin IC 4066

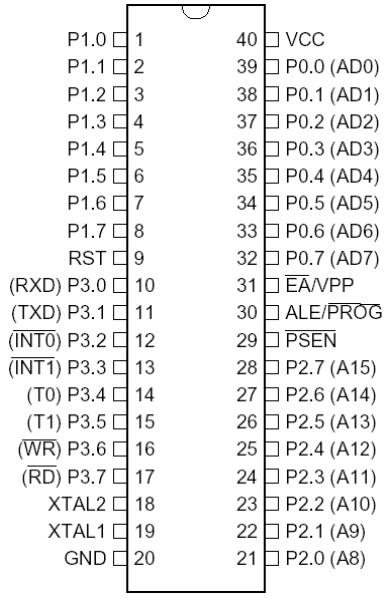
2.6.2. Mikrokontroler AT89C51

AT89C51 adalah mikrokontroler keluaran Atmel dengan 4 Kbyte Flash PEROM (*Programmable and Erasable Read Only Memory*). AT89C51 merupakan memori dengan teknologi *nonvolatile memory*, yaitu isi memori tersebut dapat diisi ulang ataupun dihapus berkali-kali.

Memori ini biasa digunakan untuk menyimpan instruksi berstandar MCS-51 sehingga memungkinkan mikrokontroler ini untuk bekerja dalam mode *single chip operation* yang tidak memerlukan *external memory* untuk menyimpan *source code* tersebut.

2.6.2.1.Deskripsi Pin

AT89C51 mempunyai 40 kaki, 32 kaki diantaranya adalah kaki untuk keperluan port paralel. Satu port paralel terdiri dari 8 kaki, dengan demikian 32 kaki tersebut membentuk 4 buah port paralel, yang masing-masing dikenal sebagai Port-0, Port-1, Port-2 dan Port-3. Nomor dari masing-masing kaki dari port paralel mulai dari 0 sampai 7. Jalur atau kaki pertama Port-0 disebut sebagai P0.0 dan jalur terakhir untuk port-0 adalah P0.7. Letak dari masing-masing port diperlihatkan pada gambar 2.14 dibawah ini.



Gambar 2.14. Konfigurasi pin ATMEL AT89C51

Adapun nama dan fungsi dari pin-pin pada mikrokontroler AT89C51 adalah sebagai berikut:

1. VCC (pin 40) : Power supply
2. GND (pin 20) : Ground
3. Port 0 (pin 32-39)

   Port 0 dapat berfungsi sebagai I/O biasa, *low order multiplex address/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. Pada fungsi sebagai *low order multiplex address/data* port ini akan mempunyai *internal pull up*. Pada saat *Flash Programming* diperlukan *external pull up* terutama pada saat verifikasi program.
4. Port 1 (pin 1-8)

   Port 1 berfungsi sebagai I/O biasa atau menerima *low order address bytes* pada saat *Flash Programming*. Port ini mempunyai *internal pull up* dan berfungsi sebagai input dengan memberikan logika 1. Sebagai output port ini dapat memberikan *output sink* keempat buah input TTL.
5. Port 2 (pin 21-28)

   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. Port ini mempunyai *internal pull up* dan berfungsi sebagai input dengan memberikan logika 1. Sebagai output, port ini dapat memberikan *output sink* keempat buah input TTL.

6. Port 3 (pin 10-17)

   Sebagai I/O biasa port 3 mempunyai sifat yang sama dengan port 1 maupun port 2. Port 3 menyediakan beberapa fungsi khusus sebagaimana diperlihatkan pada tabel 2.2.

Tabel 2.2 Fungsi-fungsi alternatif pada port 3

Kaki Port	Fungsi Alternatif
P3.0	RXD (port input serial)
P3.1	TXD (port output serial)
P3.2	INT0 (interupsi eksternal 0)
P3.3	INT1 (interupsi eksternal 1)
P3.4	T0 (input eksternal timer 0)
P3.5	T1 (input eksternal timer 1)
P3.6	WR (sinyal *write* pada data memori eksternal)
P3.7	RD (sinyal *read* pada data memori eksternal)

   Fungsi-fungsi alternatif pada tabel diatas hanya dapat diaktifkan jika bit-bit pengancing (*latch*) port yang bersangkutan berisi ‘1’.

7. RST (pin 9)

   Masukan reset kondisi 1 selama 2 siklus mesin selama osilator bekerja akan me-*reset* mikrokontroler yang bersangkutan ke alamat awal.

8. ALE/PROG (pin 30)

   Pin ini dapat berfungsi sebagai *Address Latch Enable* (ALE) yang me-*latch low byte address* pada saat mengakses memori eksternal. Sedangkan pada saat *Flash Programming* (PROG) berfungsi sebagai pulse input. Pada operasi normal ALE akan mengeluarkan sinyal *clock* sebesar 1/16 frekuensi oscillator kecuali pada saat mengakses memori eksternal. Sinyal *clock* pada pin ini dapat pula di-*disable* dengan men-set bit 0 dari *special function register* di alamat 8EH. ALE hanya akan aktif pada saat mengakses memori eksternal (MOVX & MOVC).

9. PSEN (pin 29)

   Pin ini berfungsi pada saat mengeksekusi program yang terletak pada memori eksternal. PSEN akan aktif dua kali setiap *cycle*.

10. EA/VPP (pin 31)

    Pada kondisi *low*, pin ini akan berfungsi sebagai *External Acces Enable* (EA), yaitu mikrokontroler akan menjalankan program yang ada pada memori eksternal setelah sistem di-*reset*. Jika berkondisi *high*, pin ini akan berfungsi untuk menjalankan program yang ada di memori internal.

11. XTAL1 (pin 19) : input oscillator.
12. XTAL2 (pin 18) : output oscillator.

2.6.2.2.Port Input/Output AT89C51

Seperti yang telah dijelaskan sebelumnya, mikrokontroler AT89C51 mempunyai empat buah port, yaitu Port 0, Port 1, Port 2 dan Port 3 yang terletak pada alamat 80H, 90H, A0H dan B0H. Namun, jika digunakan memori eksternal atau pun fungsi-fungsi khusus, seperti interupsi eksternal, serial atau pun timer eksternal, Por*t* 0, Port 2, dan Port 3 tidak dapat digunakan sebagai port dengan fungsi umum. Untuk itu disediakan port 1 yang dikhususkan untuk port dengan fungsi umum.

Port 0, Por*t* 2, dan Port 3 memiliki fungsi alternatif. Masing-masing pin dari port-port ini dapat digunakan sebagai jalur input / output digital secara umum atau alternatifnya dapat digunakan untuk fungsi keduanya. Fungsi kedua dari Port 0 dan Port 2 adalah untuk menghubungkan dengan memori eksternal. Ketika program eksternal atau memori data sedang diakses, Port 2 mengeluarkan byte tinggi dari alamat 16-bit. Port 0 awalnya mengeluarkan byte rendah dari alamat 16-bit, kemudian mengirim atau menerima byte data.

Semua port ini dapat diakses dengan pengalamatan secara bit sehingga dapat dilakukan perubahan output pada tiap-tiap pin dari port ini tanpa mempengaruhi pin-pin yang lainnya.

Ketika sebuah port digunakan sebagai port input, nilai FFH harus dituliskan pertama kali ke port, kemudian setiap input yang menggunakan tegangan rendah akan dianggap sebagai nilai 0, dan port tersebut dapat dibaca dari SFR yang sesuai. Lebih spesifik pembacaan SFR yang sesuai, membaca nilai pin port. Output latch menggerakkan pin port ke level logika ‘1’ jika tidak ada penurunan arus rangkaian eksternal pada pin tersebut.

Fungsi alternatif dari pin-pin *Port* 3 termasuk interrupt dan input timer, serial port input dan output, dan sinyal kontrol untuk menghubungkan dengan memori eksternal.

2.6.2.3 Struktur Memori

AT89C51 mempunyai struktur memori yang terdiri dari:

1. RAM Internal

   RAM internal memiliki memori sebesar 128 byte yang biasanya digunakan untuk menyimpan variabel atau data yang bersifat sementara, dialamati oleh *RAM Address Register* (Register Alamat RAM). RAM internal terdiri atas:

   a. *Register Banks*

      89C51 memiliki delapan buah register yang terdiri dari R0 sampai R7 yang tereletak pada alamat 00H hingga 07H pada setiap kali *reset.*

   b. *Bit Addressable* RAM

      RAM dengan alamat 20H hingga 2FH dapat diakses secara pengalamatan bit *(bit addressable)* sehingga hanya dengan sebuah instruksi saja setiap bit dalam area ini dapat di-set, *clear,* AND dan OR.

   c. RAM Keperluan Umum

      RAM keperluan umum dimulai dari alamat 30H hingga 7FH dan dapat diakses dengan pengalamatan langsung maupun tak langsung. Pengalamatan langsung dilakukan ketika salah satu *operand* merupakan bilangan yang dialamati. Sedangkan pengalamatan tak langsung pada lokasi dari RAM Internal ini adalah akses data dari memori ketika alamat memori tersebut tersimpan dalam suatu register R0 atau R1 yang dapat digunakan sebagai *pointer* dari lokasi memori pada RAM Internal.

2. Special *Function Register* (Register Fungsi Khusus)

   Memori yang berisi register-register yang memiliki fungsi khusus yang tersediakan oleh mikrokontroler, seperti *timer*, serial dan lain-lain. 89C51 memiliki 21 *Special Function Register* yang terletak pada alamat 80H hingga FFH dengan rincian pada tabel 2.2. Salah satu contoh dari *Special Function Register* adalah *Accumulator*, register ini terletak pada alamat E0H. Semua operasi aritmatika dan operasi logika dan proses pengambilan dan pengiriman data ke memori selalu menggunakan register ini, seperti dapat di lihat pada tabel 2.3.

Tabel 2.3 Alamat register fungsi khusus

Register	Mnemonic	Alamat
P0	Port 0 Latch	80H
SP	Stack Pointer	81H
DPTR	Data Pointer	82H-83H
DPL	Data Pointer Low Byte	82H
DPH	Data Pointer High Byte	83H
PCON	Power Control	87H
TCON	Timer/Counter Control	88H
TMOD	Timer/Counter Mode Control	89H
TL0	Timer/Counter 0 Low Byte	8AH
TL1	Timer/Counter 1 Low Byte	8BH
TH0	Timer/Counter 0 High Byte	8CH
TH1	Timer/Counter 1 High Byte	8DH
P0	Port 0 Latch	80H
SP	Stack Pointer	81H
DPTR	Data Pointer	82H-83H
DPL	Data Pointer Low Byte	82H
DPH	Data Pointer High Byte	83H
PCON	Power Control	87H
TCON	Timer/Counter Control	88H

Tabel 2.3 lanjutan

TMOD	Timer/Counter Mode Control	89H
TL0	Timer/Counter 0 Low Byte	8AH
TL1	Timer/Counter 1 Low Byte	8BH
TH0	Timer/Counter 0 High Byte	8CH
TH1	Timer/Counter 1 High Byte	8DH
P1	Port 1 Latch	90H
SCON	Serial Port Control	98H
SBUF	Serial Data Port	99H
P2	Port 2 Latch	A0H
IE	Interrupt Enable	A8H
P3	Port 3 Latch	B0H
IP	Interrupt Priority Control	B8H
PSW	Program Status Word	D0H
ACC	Accumulator	E0H
B	Register B	F0H

3. *Flash PEROM*

   Memori yang digunakan untuk menyimpan instruksi-instruksi MCS-51 dialamati oleh *Program Address Register* (Register Alamat Program). AT89C51 memiliki 4 Kb *Flash* PEROM yang menggunakan *Atmel's High-Density Non Volatile Technology*.

   Program yang ada pada *Flash* PEROM akan dijalankan jika pada saat sistem di-*reset*, pin EA/VP berlogika satu maka mikrokontroler aktif berdasarkan program yang ada pada *Flash* PEROM-nya. Namun jika pin EA/VP berlogika nol, mikrokontroler aktif berdasarkan program yang ada pada memori eksternal.

2.6.2.4.Mode Operasi Port Serial

Dalam port serial 89C51 mempunyai 4 buah mode operasi yang diatur oleh bit ke 7 dan bit ke 5 dari register SCON (serial control).

SCON

	SCON.7	SCON.6	SCON.5	SCON.4	SCON.3	SCON.2	SCON.1	SCON.0
911	SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0 : serial port mode 0, bit pengatur mode serial

SM1 : serial port mode 1, bit pengatur mode serial

SM2 : serial port mode 2, bit untuk mengaktifkan komunikasi multiprocessor pada kondisi set.

REN : *receive enable*, bit ini untuk mengaktifkan penerimaan data dari port serial pada kodisi set. Bit ini diset dan clear oleh perangkat lunak.

TB8 : *transmit* bit 8, bit ke 9 yang akan dikirimkan pada mode 2 tau 3. bit ini diset dan clear oleh perangkat lunak.

RB8 : *Receive* bit 8, bit ke 9 yang akan diterima pada mode 2 tau 3. pada mode 1 bit ini berfungsi sebagai stop bit.

TI : *Transmit interrupt flag* , bit yang akan diset pada akhir pengiriman karakter. Bit ini diset oleh perangkat keras dan di *clear* oleh perangkat lunak.

RI : *receive interrupt flag*, bit yang akan di set pada akhir penerimaan karakter. Bit ini diset oleh perangkat keras dan di *clear* oleh perangkat lunak

Tabel 2.4 Mode operasi port serial

SM0	SM1	Mode	Deskripsi
0	0	0	*Shift register 8 bit*
0	1	1	*UART 8 bit dengan baud rate yang dapat diatur*
1	0	2	*UART 9 bit dengan baud rate permanen*
1	1	3	*UART 9 bit dengan baud rate yang dapat diatur*

2.7. Baudrate

Baudrate adalah jumlah bit data yang terkirim tiap detik.Untuk mengakses port serial, ada beberapa hal yang harus diatur terlebih dahulu dengan mengisi beberapa register tertentu. Proses penentuan mode serial dilakukan dengan mengisi SCON. Baud Rate Serial Baudrate dari Port Serial 89C51 dapat diatur pada Mode 1 dan Mode 3, namun pada Mode 0 dan Mode 2, baudrate tersebut mempunyai kecepatan yang permanen yaitu untuk Mode 0 adalah 1/12 frekwensi osilator dan Mode 2 adalah 1/64 frekwensi osilator.

Dengan mengubah bit SMOD yang terletak pada Register PCON menjadi set (kondisi awal pada saat sistem reset adalah clear) maka baudrate pada Mode 1, 2 dan 3 akan berubah menjadi dua kali lipat.Pada Mode 1 dan 3 baud rate dapat diatur dengan

menggunakan Timer1. Cara yang biasa digunakan adalah Timer Mode 2 (8 bit auto reload) yang hanya menggunakan register TH1 saja. Pengiriman setiap bit data terjadi setiap Timer 1 verflow sebanyak 32 kali sehingga dapat disimpulkan bahwa:
Lama pengiriman setiap bit data = Timer 1 Overflow X 32 .....

Baudrate (jumlah bit data yang terkirim tiap detik) = $\frac{1}{\text{Timer 1 Overflow X 32}}$

Timer 1 dapat digunakan sebagai pewaktu untuk mengatur *baud rate* pada komunikasi serial. Lama pengiriman tiap bit data = timer 1 overflow x 32

Rumus frekuensi osilator terhadap *baud rate* :

$$\frac{12x(FFH - TH1)}{Fosc} = \frac{1}{Baudratex32} \quad \text{..... (2.3)}$$

Rumus nilai register TH1 :

$$TH1 = 256 - \left[\frac{Fosc}{12xbaudratex32}\right] \quad \text{..... (2.4)}$$

Jika dikehendaki baudrate 1200 bps, timer 1 harus diatur agar oferflow setiap $\frac{1}{1200x32}$ detik. Timer 1 overflow tiap TH1 mencapai nilai FFh dengan nilai frekuensi sebesar Fosc/12, maka formulasi untuk kasus ini adalah :

$$TH1 = 256 - \frac{11.059.200}{(12x1200x32)}$$

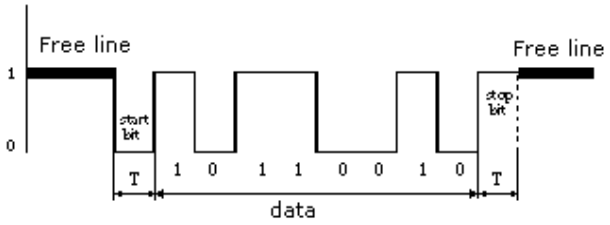
Rumus nilai register TH1 dengan baudrate 1200 bps dengan frekuensi 11,0592 MHz, maka TH1 adalah 232 atau E8H.

2.8. Komunikasi Data Serial

Dikenal ada dua cara komunikasi data serial, yaitu komunikasi data serial secara Sinkron dan komunikasi data serial secara Asinkron. Pada komunikasi data serial sinkron, clock dikirimkan bersama-sama dengan data serial, sedangkan komunikasi data serial asinkron, clock tidak dikirimkan bersama data serial, tetapi dibangkitkan secara sendiri-sendiri baik pada sisi pengirim (*transmitter)* maupun pada sisi penerima *(reciver)*. Pada PC kompatibel port serialnya termasuk jenis asinkron. Komunikasi data serial ini dikerjakan oleh UART (Universal Asyncronous Receiver Transmitter), pada UART kecepatan pengiriman data *(baudrate)* dan fase clock pada bagian pemancar dan

bagian penerima harus sinkron, untuk itu diperlukan sinkronisasi antara dua bagian tersebut.

Salah satu caranya adalah dengan mengirimkan bit ‘start’ dan bit ‘stop’. Untuk bit ‘start’ adalah data biner 0 dan untuk bit ‘stop’ adalah data biner 1. Setelah pengiriman bit ‘start’ maka akan diikuti oleh data yang akan dikirim, selanjutnya diakhiri dengan bit ’stop’. Berikut adalah contoh pengiriman karakter B2 heksa atau 10110010 biner tanpa bit paritas. Dapat terlihat pengiriman data diawali dengan bit ‘ start’ lalu data B2 heksa dan diakhiri dengan bit ’stop’ sebagai akhir dari pengiriman.



Gambar 2.15 Pengiriman data serial

Kecepatan pengiriman data (*baud rate*) bervariasi, mulai dari 110, 135, 150, 300, 600, 1200, 2400, 4800 dan 9600 (bit/detik). Pada komunikasi data serial *baud rate* dari kedua bagian harus diatur pada kecepatan yang sama. Setelah itu harus ditentukan panjang datanya, apakah 6, 7 atau 8 bit.

2.8.1. Port Serial/RS-232

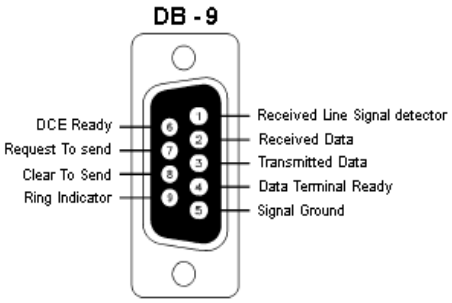
Port serial lebih sulit ditangani daripada *port* paralel karena peralatan yang dihubungkan ke *port* serial harus komunikasi dengan menggunakan transmisi serial, sedangkan data di komputer diolah secara paralel dan serial. Sehingga, data dari *port* serial harus dikonversikan kebentuk paralel atau sebaliknya untuk bisa digunakan secara hardware hal ini bisa digunakan oleh UART (*Universal Asyncronus Receiver Transmitter*).

Adapun keunggulan menggunakan port serial dari pada port paralel sebagai transfer data yaitu :

1. Kabel port serial bisa lebih panjang dibandingkan kabel port paralel. Hal ini karena port serial mengirimkan logika 1 sebagai –3 Volt hingga –25 Volt dan logika 0 sebagai +3 Volt hingga +25 Volt, sedangkan port paralel menggunakan TTL, yakni hanya 0 Volt untuk logika 0 dan +5

Volt untuk logika 1. ini berarti port serial mamiliki rentang kerja 50 Volt sehingga kehilangan daya karena panjang kabel bukan merupakan masalah serius jika dibandingkan dengan port paralel.

2. Transmisi serial memerlukan lebih sedikit jumlah kabel dibandingkan dengan transmisi paralel, bisa hanya menggunakan tiga kabel yaitu saluran *transmit* Data, *Receive* Data dan Ground, dapat di lihat pada gambar 2.10



Gambar 2.16 Susunan Port Serial 9 pin

EIA (*Electronic Industry Association*) mengeluarkan spesifikasi listrik untuk standar RS-232 yaitu :

1. *Space* (logika 0) antara +3 sampai +15 Volt.
2. *Mark* (logika 1) antara –3 sampai –15 Volt.
3. Daerah antara +3 Volt dan –3 Volt tidak ditetapkan.
4. Tegangan rangkaian terbuka tidak boleh melebihi 25 Volt (terhadap *Ground*).
5. Arus pada rangkaian tertutup *(Short Circuit)* atau hubung singkat tidak boleh melebihi 500mA

Tabel 2.5 Konfigurasi pin dan nama sinyal konektor serial DB9

Nama Pin	**Nama Sinyal**	**Direction**	**Keterangan**
1	DCD	In	Data Carrier Detect/Received Line Signal Detect
2	DSR	In	Received Data
3	RxD	Out	Transmite Data
4	TxD	Out	Data Terminal Ready
5	DTR	-	Ground
6	GND	In	Data Set Ready
7	DSR	Out	Request to Send
8	CTS	In	Clear to Send
9	RI	In	Ring Indicator

Keterangan mengenai saluran RS-232 pada konektor DB9 adalah sebagai berikut :

1. *Received Line Signal Detect*, dengan saluran ini DCE memberitahukan ke DTE bahwa terminal masukan ada data masukan.
2. *Received Data,* digunakan DTE menerima data dari DCE.
3. *Transmite Data,* digunakan DTE mengirimkan data ke DCE.
4. *Data Terminal Ready*, pada saluran ini DTE memberitahukan kesiapan sinyalnya.
5. *Signal Ground*, saluran Ground.
6. *Ring Indicator*, pada saluran ini DCE memberitahukan ke DTE bahwa stasiun menghendaki hubungan dengannya.
7. *Clear to Send*, dengan saluran ini DCE memberitahukan ke DTE boleh mengirimkan data.
8. *Request to Send*, dengan saluran ini DCE diminta mengirimkan data oleh DTE. *DCE Ready,* sinyal aktif pada saluran ini menunjukan bahwa DCE sudah siap.

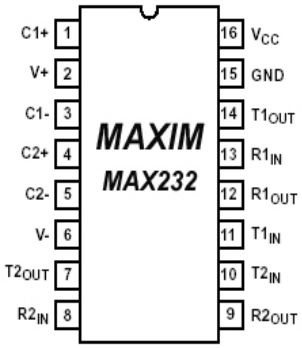
2.8.2. Perantara Mikrokontroler dan Port Serial

Untuk dapat berhubungan dengan PC, mikrokontroler harus membutuhkan komponen tambahan baik komunikasi paralel maupun serial. Pada pembuatan tugas akhir ini yang digunakan adalah komunikasi serial. Pada mikrokontroler sendiri terdapat penyesuai tegangan yang dapat digunakan sebagai pendukung proses komunikasi tersebut. Pada saat ini banyak komponen yang dapat digunakan untuk pendukung proses komunikasi tersebut, salah satu contohnya adalah maxim232.

Maxim232 berfungsi sebagai perantara antara mikrokontroler dengan port serial, karena mikrokontroler tidak dapat mengirim data begitu saja maka diperlukan maxim232. Di dalam IC terdapat charge pump yang akan membangkitkan +10 Volt dan -10 Volt dari sumber +5 Volt tunggal dalam IC DIP (*Dual in-line Package*) 16 pin (8 pin x 2baris) ini terdapat 2 buah transmiter dan dua buah receiver. Jadi IC ini berfungsi sebagai perantara karena maxim232 hanya menerima data dari mikrokontroler untuk kemudian dikirim ke pc melalui DB9.

2.8.3. Deskripsi Pin Maxim232

Maxim232 mempunyai 16 kaki yang terdiri untuk keperluan port serial, komunikasi mikrokontroler dengan maxim. Letak dari masing-masing port diperlihatkan pada gambar 2.11 dibawah ini.



Gambar 2.17 Konfigurasi pin MAXIM232

Adapun nama dan fungsi dari kaki-kaki pin pada Maxim232 adalah sebagai berikut:

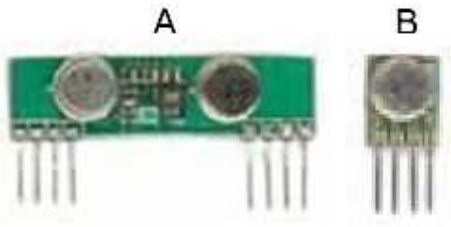
1. VCC (pin 16) : Power supply
2. GND (pin 15) : Ground
3. T_{1IN} dan R_{1OUT} (pin 11 dan 12) : Pin ini terhubung dengan pin 11 mikrokontroler AT89C51.
4. R_{1IN} dan T_{1OUT} (pin 13 dan 14) : Pin ini terhubung dengan pin 2 dan 3 DB9.
5. C_{1+} dan C_{1-} :
6. C_{2+} dan C_{2-} :
7. V_+ dan V_- : Tegangan referensi dari Maxim232.

2.9. Modul RF

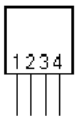
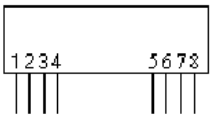
Modul RF yang digunakan adalah TLP-434A (Pemancar) dan RLP-434A (Penerima). Digunakannya TLP dan RLP 434A sebagai modul RF (Radio Frekuensi) pada perancangan Tugas akhir ini, selain kemampuannya di dalam pengiriman dan penerimaan data yang cukup baik.

Modul RF buatan LAIPAC ini sering sekali digunakan sebagai alat untuk komunikasi data. Input masukan data adalah serial dengan level TTL (Transistor – Transistor Logic). Jarak pancar maksimum dari modul RF ini adalah 100 meter tanpa halangan dan 30 meter di dalam gedung. Ukuran ini dapat dipengaruhi oleh faktor

antena,kebisingan, dan tegangan kerja dari pemancar. Panjang antena yang digunakan adalah 17 cm, dan terbuat dari besi alumunium.



Gambar 2.18 Bentuk fisik TLP-434 dan RLP-434

 modul tampak dari depan	Pin 1: GND Pin 2: Data In Pin 3: VCC Pin 4: Antenna (RF Output)	 modul tampak dari depan	Pin 1: GND Pin 2: Digital Data Output Pin 3: Linear Output / Test Pin 4: VCC Pin 5: VCC Pin 6: GND Pin 7: GND Pin 8: Antenna (RF Output)

Gambar 2.19 Konfigurasi kaki TLP-434 dan RLP-434

2.10. Bahasa *Assembly*

Assembler adalah program komputer yang men-*translitrasi* program dari bahasa *assembly* ke bahasa mesin. Sedangkan bahasa *assembly* adalah ekuivalensi bahasa mesin dalam bentuk alpanumerik. *Mnemonics* alpanumerik digunakan sebagai alat bantu bagi programer untuk memprogram mesin komputer daripada menggunakan serangkaian 0 dan 1 (bahasa mesin) yang panjang dan rumit.

2.10.1. Konstruksi Program *Assembly*

Program sumber *assembly* terdiri dari kumpulan baris-baris perintah dan biasanya disimpan dengan *extension.* ASM dengan 1 baris untuk satu perintah, setiap baris perintah tersebut bisa terdiri atas beberapa bagian, yakni bagian label, bagian *mnemonic*, bagian *operand*. Program sumber (*source code*) dibuat dengan program editor seperti *notepad* atau Editor DOS, selanjutnya program sumber diterjemahkan ke bahasa mesin dengan menggunakan program *assembler*. Hasil kerja program *assembler* adalah "*program objek*" dan juga "*assembly listing*". Ketentuan penulisan *source code* adalah sebagai berikut:

1. Masing-masing bagian dipisahkan dengan spasi atau TAB, khusus untuk *operand* yang lebih dari satu masing-masing *operand* dipisahkan dengan koma.
2. Bagian-bagian tersebut tidak harus semuanya ada dalam sebuah baris, jika ada satu bagian yang tidak ada maka spasi atau TAB sebagai pemisah bagian tetap harus ditulis.
3. Bagian label ditulis mulai huruf pertama dari baris, jika baris bersangkutan tidak mengandung label maka label tersebut digantikan dengan spasi atau TAB, yakni sebagai tanda pemisah antara bagian label dan bagian *mnemonic*.

2.10.2. Instruksi MCS-51 yang Digunakan

Beberapa instruksi yang digunakan dalam penyusunan program sensor kendaraan adalah sebagai berikut :

1. ORG

   Digunakan untuk menunjukkan lokasi memori tempat instruksi atau perintah yang ada di bawahnya disimpan.
2. CLR

   Memberikan nilai “nol” pada bit tertentu.
3. SETB

   Memberikan nilai “satu” pada bit tertentu.
4. CALL

   Instruksi melakukan lompatan dengan area sebesar 2 Kbyte.
5. MOV

   Instruksi ini melakukan pemindahan data dari variabel pada kode operasi kedua dan disimpan di variabel pada kode operasi pertama.
6. JMP

   Melakukan lompatan dan menjalankan program yang berada di alamat yang ditentukan oleh label tertentu.
7. SJMP

   Melakukan lompatan untuk jarak yang pendek (*Short JMP*).
8. LJMP

   Melakukan lompatan untuk jarak yang jauh (*Long JMP*).
9. JNB

Kebalikan dari JB, JNB merupakan instruksi untuk beralih ke alamat tertentu jika Bit tertentu bernilai “nol”.

10. ANL

    Logika AND untuk variabel tertentu. Perintah ANL bekerja dengan melakukan operasi AND antara variabel yang ditentukan dengan nilai tertentu untuk kemudian hasilnya disimpan pada alamat yang ditunjuk oleh variabel yang ditentukan.

11. ORL

    Logika OR untuk variabel tertentu. Perintah ORL bekerja dengan melakukan operasi OR antara variabel yang ditentukan dengan nilai tertentu untuk kemudian hasilnya disimpan pada alamat yang ditunjuk oleh variabel yang ditentukan.

12. INC

    Menambahkan nilai variabel yang ditunjuk dengan 1 dan hasilnya disimpan di variabel tersebut.

13. SUBB

    Mengurangkan nilai variabel yang ditunjuk dengan nilai lain dan hasilnya disimpan di variabel tersebut.

14. ADD

    Menjumlahkan nilai variabel yang ditunjuk dengan nilai lain dan hasilnya disimpan di variabel tersebut.

15. END

    END biasanya diletakkan di akhir baris dari file program sumber *assembler* sebagai tanda akhir pernyataan (*statement*) bagi program *assembler* dalam melakukan proses *assembly*.

2.11. Program Visual Basic 6.0

Visual Basic adalah program pembuat aplikasi *Microsoft windows*. Kata “Visual” menunjukan cara yang digunakan untuk membuat *graphical user interface* (*GUI*). Dengan cara ini *programmer* tidak lagi menuliskan intruksi pemprograman dalam kode-kode baris, tetapi secara mudah dilakukan *drag* dan *drop* objek-objek yang akan digunakan.

Kata “Basic” menunjukan basic program ini merupakan bagian bahasa Basic *(beginners All-Purpose Symbolic Intruction Code),* yaitu sebuah bahasa pemrograman

yang dalam sejarahnya sudah banyak digunakan untuk menyusun sebuah aplikasi. *Visual Basic* dikembangkan dari bahasa BASIC dan sekarang berisi banyak *statement*, fungsi, dan *keyword* yang beberapa diantaranya terhubung ke *windows GUI*.

Visual Basic dilengkapi dengan komponen tambahan untuk koneksi dengan perangkat eksternal melalui port, yaitu dengan menambahkan control **Microsoft Comm Control 6.0** pada *form* aplikasi dengan cara yang sama seperti menambahkan *command button* pada *form* aplikasi. Untuk menggunakan control ini hanya perlu menambahkan beberapa baris kode tambahan, diawali dengan membuka koneksi *port*, pengiriman data, kemudian menutup kembali koneksi port. Kode tersebut dituliskan seperti dibawah ini :

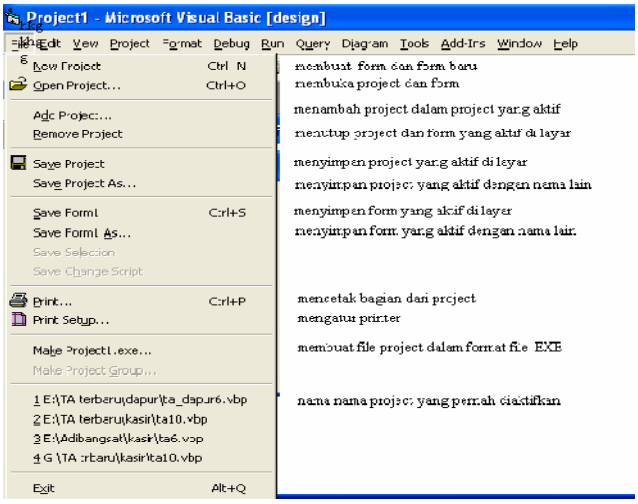
```
MSComm1.PortOpen = True          'membuka koneksi port'
MSComm1.Output = Text1.text      'mengirimkan data berupa text'
MSComm1.PortOpen = False         'menutup koneksi port'
```

2.11.1. Pemakaian Komponen

Komponen adalah suatu kesatuan pembentuk aplikasi yang digunakan untuk merancang user interface program aplikasi. Komponen terbagi menjadi dua, yaitu vsiual dan non-visual. Komponen visual misalnya tombol sedangkan contoh yang non-visual adalah print kotak dialog. Masing-masing komponen memiliki atribut tersendiri dalam melakukan aksi kendalinya.

2.11.2. Menu

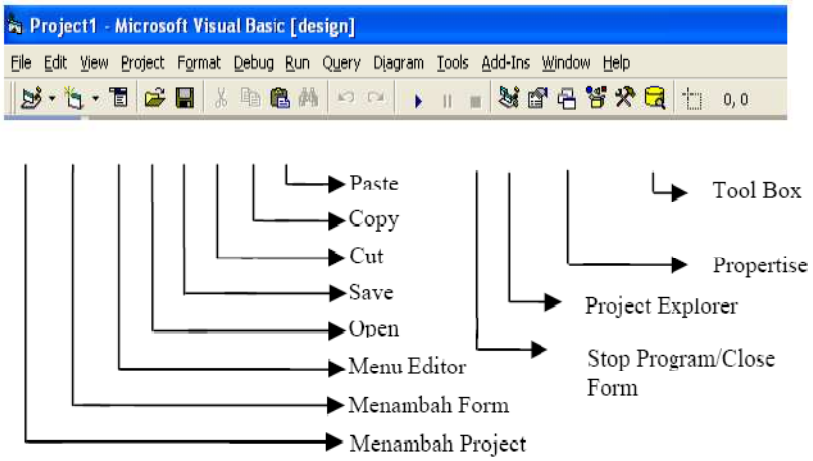
Menu merupakan sekumpulan perintah yang dipakai dalam MS Visual Basic. Berikut merupakan contoh tampilan salah satu menu pada MS Visual Basic.



Gambar 2.20 Menu di dalam Visual Basic 6.0

2.11.3. Toolbar Standar

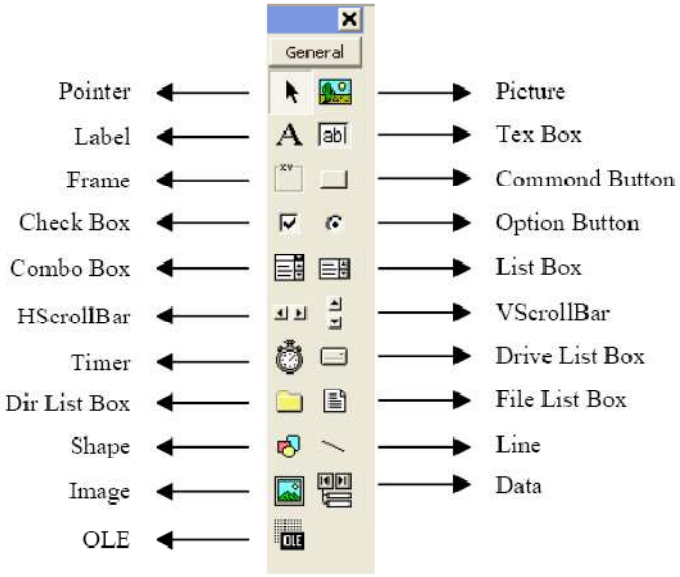
Gambar 2.8 yang mewakili suatu perintah.



Gambar 2.21. Komponen standar pada Visual Basic 6.0

2.11.4. Tool Box

Komponen Visual Basic yang akan ditempatkan pada form.



Gambar 2.22. Tool Box didalam Visual Basic 6.0

Berikut ialah keterangan dari masing masing Tool Box yang ditampilkan pertama kali atau tampilan default dari Visual Basic.

a. Pointer : Mengklik suatu obyek.

b. Picture : Menyisipkan gambar pada form

c. Label : Sebagai keterangan tanpa proses input.
d. TextBox : Dipakai untuk input atau output data.
e. Frame : Untuk mengelompokkan Control
f. Comman on : Tombol perintah
g. Check B : Untuk memilih pilihan yang lebih
h. Option B : Untuk memilih salah satu pilihan
i. Combo B : Membuat daftar pilihan secara dropdown
j. List Box : Membuat daftar pilihan tampak lan
k. Hscroll Bar : Horisontal scroll bar
l. Vscroll Bar : Vertikal scroll bar
m. Timer : Mengatur waktu
n. Drive List Box : Membuat daftar drive
o. Dir List B : Membuat daftar direktori
p. File List Box : Membuat daftar file
q. Shape : Membuat bentuk lingkaran, kotak dll
r. Line : Membuat garis
s. Image : Menyisipkan gambar
t. Data : Menyisipkan data base
u. OLE : Prosses Embeding File

Apabila tool-tool yang lain ingin ditambahkan maka step yang dilakukan ialah dengan menekan tombol Crtl + T. Atau pilihan menu Project-Component.

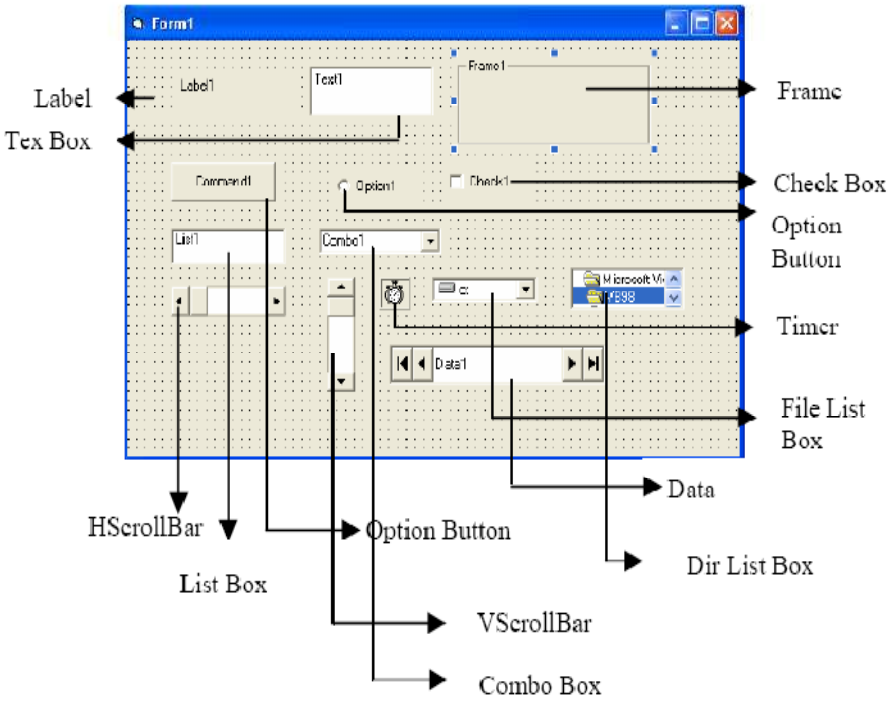
2.11.5. Form

Tempat untuk meletakkan kontrol dan sebagai layar secara visual

Cara menempatkan Control (Tool Box) ada 2 cara

1. Click salah satu Control dan drag pada form dengan posisi dan ukuran tertentu
2. Double Click salah satu Control maka akan tampil pada form. Kemudian atur posisi dan ukurannya.

Contoh pemakaian Control pada form yaitu :

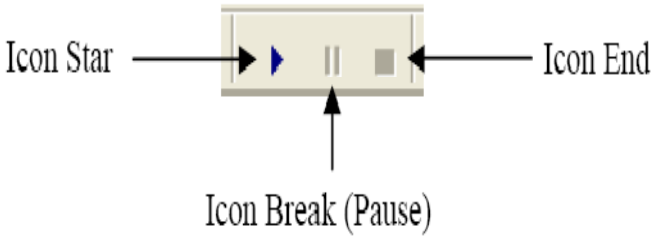


Gambar 2.23. Tampilan Tool Box didalam Form Visual Basic 6 .0

Ada tiga langkah dalam membuat program aplikasi dengan Visual Basic. Yaitu :

1. Merancang *Form* dengan meletakkan dan menggambar kontrol-kontrol sesuai dengan kebutuhan pemograman.
2. Inisialisasi nilai-nilai properti untuk masing-maasing *form* dan *control*.
3. Menulis *code* program untuk menjalankan perintah-perintah yang diharapkan.

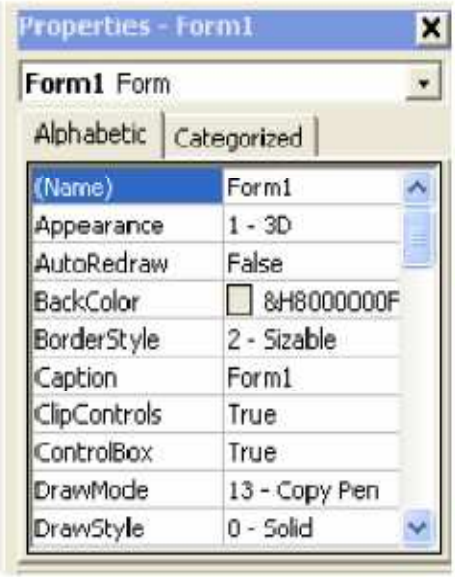
Untuk menjalankan program aplikasi anda dapat melakukannya telah menyimpan program aplikasi yang telah dibuat, jalankan dengan memilih menu Run lalu Start atau tekan fungsi tombol F5. Seperti pada gambar 2.11.



Gambar 2.24. Tombol Eksekusi Program

2.11.6. Properties

Properties digunakan untuk menentukan setting suatu objek. Suatu objek biasanya mempunyai beberapa properti yang dapat diatur langsung dari jendela Properties atau lewat kode program. Setting properti akan menentukan cara kerja dari objek yang bersangkutan saat program aplikasi dijalankan, misalnya menentukan warna objek, bingkai objek, pengambilan data dan lain-lain.



Gambar 2.25. Tampilan Jendela Properties

2.12. Pengaksesan Port Serial Pada Visual Basic

Untuk pengaksesan port serial dapat mengaksesnya secara langsung malalui register UART (Universal Asyncronous Reciver/Transmitter) atau menggunakan control MSComm yang telah disediakan Visual Basic.

2.12.1. Pengaksesan Secara Langsung Melalui Register UART

Saluran yang digunakan UART untuk komunikasi baik untuk pengiriman maupun penerimaan data adalah saluran RxD dan saluran TxD serta saluran-saluran untuk control, yaitu saluran DCD, DSR, RTS, CTS, DTR, dan RI. Saluran-saluran ini ada sebagai output dan ada yang sebagai input. Kecuali saluran RxD, saluran-saluran ini dapat diakses secara langsung melalui register UART. Berikut adalah tabel alamat dan lokasi bit saluran tersebut pada register UART.

Tabel 2.6 Alamat dan lokasi bit pada register UART

Nama pin	Nomor pin pada DB-9	COM1	COM2	Bit	Arah
TxD	3	3FBh	2FBh	6	Output
DTR	4	3FCh	2FCh	0	Output
RTS	7	3FCh	2FCh	1	Output
CTS	8	3FEh	2FEh	4	Input
DSR	6	3FEh	2FEh	5	Input
RI	9	3FEh	2FEh	6	Input
DCD	1	3FEh	2FEh	7	Input

Untuk dapat mengaksesnya dapat menggunakan fungsi *Port_Out* dan *Port_In*, namun pada VB 6.0 tidak disediakan secara langsung, harus ada program tambahan tersendiri untuk pengaksesan UART ini.

2.12.2. Pengaksesan Dengan Menggunakan Kontrol MSComm

Kontrol MSComm menyediakan fasilitas komunikasi antara program aplikasi dengan *port* serial untuk mengirim dan menerima data melalui *port* serial. Setiap MSComm hanya menangani satu *port* serial sehingga jika ingin menggunakan lebih dari satu *por*t serial, maka harus menggunakan MSComm sebanyak *port* serial yang dipakai.

Jumlah property pada MSComm sangat banyak sehingga tidak akn dibahas secara keseluruhan. Namun hanya membahas beberapa properti yang cukup sesuai dengan kebutuhan saja properti-properti yang sering dipakai adalah sebagai berikut :

a. CommPort : Digunakan untuk menentukan nomor port serial yang akan dipakai.
b. Setting : Digunakan untuk menset nilai baudrate, parity, jumlah bit data, dan jumlah bit stop.
c. PortOpen : Digunakan untuk membuka ataupun menutup port serial yang dihubungkan dengan MSComm ini.
d. Input : Digunakan untuk mengambil data string yang ada pada buffer penerima.
e. Output : Digunakan untuk menulis data string pada buffer kirim.

Berikut adalah contoh penggunaan properti untuk mengirim satu karakter dengan MSComm pada *port* Comm1 :

```
Private Sub Form_Load ()
MSComm1.CommPort = 1
MSComm1.Settings = “9600,N,8,1”
MSComm1.PortOpen = True
MSComm1.Output = “A”
MSComm1.PortOpen = False
End Sub
```

Kode-kode program pada prosedur di atas akan melakukan aksi sebagai berikut :

a. *Port* serial yang digunakan adalah Comm1.

b. Setting MSComm dengan baudrate 9600, tanpa bit paritas, jumlah data 8 bit dan jumlah stop bit adalah 1.

c. Membuka *port* serial Comm1.

d. Mengirim satu karakter (“A”).

e. Menutup kembali comm. Serial yang dipakai.