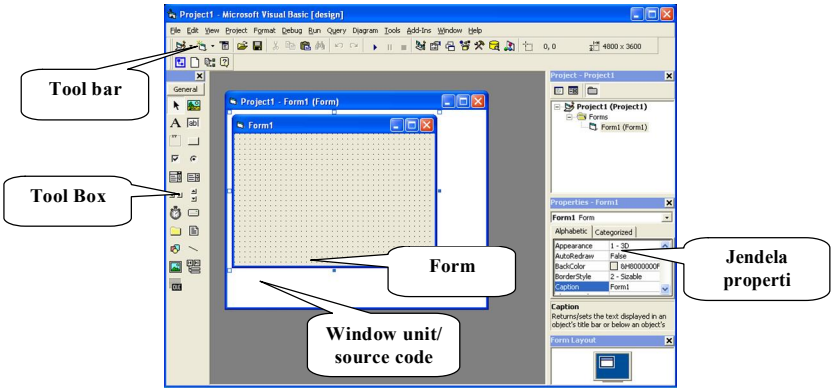
- **Commport** : digunakan untuk menentukan nomor port serial yang akn dipakai
- **Setting** : digunakan untuk menset nilai *baudrate*, *parity*, jumlah bit data, dan jumlah *bit stop*.
- **Port open** : digunakan untuk membuka ataupun menutup port serial yang dihubungkan dengan MSComm ini.
- **Input** : digunakan untuk mengambil data string yang ada dalam *buffer* penerima.
- **Output** : digunakan untuk menulis data string pada *buffer* kirim

MSComm hanya mempunyai satu event saja yaitu event OnComm. Event OnComm dibangkitkan jika nilai property dari commevent berubah yang mengindikasikan telah terjadi even pada port serial baik even komunikasi maupun even error berikut ini merupakan properti CommEvent, nilai-nilai properti ini tidak tersedia pada saat design time, tetapi hanya dapat dibaca pada saat *run time*.

2.12 Pemrograman Visual Basic 6.0

Visual basic (atau sering disebut VB) adalah perangkat lunak untuk meyusun program aplikasi yang bekerja dalam lingkungan sistem operasi windows. VB 6.0 dapat dilihat Gambar 2.15



Gambar 2.15 Lingkungan kerja visual basic 6.0

Keterangan :

- **Toolbar** : layar yang berisikan tool-tool yang dibutuhkan dalam menjalankan program yang dibuat.
- **Toolbox** : layar yang berisikan komponen-komponen yang dipakai dalam program yang akan dibuat.
- **Jendela Form** : digunakan sebagai layar/ window yang digunakan sebagai lembar kerja kita. Di *form-lah* semua komponen seperti tombol dan komponen lainya disimpan.
- **Jendela Code** : window/ layar yang berisi perintah-perintah yang akan dieksekusi oleh komputer. Dilayar inilah kita mengisikan program-program
- **Project Explorer**
- **Jendela Properties** : suatu window yang berguna untuk mengatur suatu oject baik *properti*, *even*t dan *method*

2.11.1 Pengaksesan Port Serial Dengan Visual Basic 6.0

Dalam pemrograman VB 6.0 untuk pengaksesan port serial dapat menggunakan control MSComm. Control MSComm ini menyediakan fasilitas komunikasi antara program aplikasi yang kita buat dengan port serial. Beberapa properti yang sering dipakai adalah sebagai berikut :

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.

22. LOOP

    Untuk melakukan suatu proses yang berulang (Looping) dengan CX sebagai counternya. Perintah LOOP akan mengurangkan CX dengan 1 kemudian dilihat apakah CX telah mencapai nol. Proses looping akan selesai jika register CX telah mencapai nol. Oleh karena inilah maka jika kita menjadikan CX=0 pada saat pertama kali, kita akan mendapatkan suatu pengulangan yang terbanyak karena pengurangan 0-1 =-1(FFFF).

23. INC

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

24. SUBB

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

25. ADD

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

26. ADDC

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

27. NOP

    No Operation, yang berarti tidak melakukan apa – apa. Tujuan dari instruksi ini adalah hanya untuk menambahkan siklus pulsa pewaktu sebesari 1 siklus.

28. END

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

13. JNB

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

14. JNC

    Fungsinya identik dengan JNB.

15. JC

    Melakukan suatu loncatan menuju “Tujuan” bila CF=1. “Tujuan” dapat berupa nama label atau alamat memori. Pada lompatan bersyarat ini, besaarnya lpmpatan tidak dapat melebihi -128 dan +127 byte.

16. DJNZ

    Melakukan pengurangan pada Rn (R0…R7) dengan 1 dan lompat ke alamat yang ditentukan apabila hasilnya bukan 00. Apabila hasilnya telah mencapai 00, maka program akan terus menjalankan instruksi dibawahnya.

17. CJNE

    Instruksi ini melakukan perbandingan antara data tujuan dan data sumber serta melakukan lompatan ke alamat yang ditentukan jika hasil perbandingan tidak sama.

18. RET

    Instruksi ini melakukan lompatan ke alamat yang disimpan dalam SP dan SP-1. Instruksi ini biasa digunakan pada saat kembali dari subrutin yang dipanggil dengan instruksi ACALL atau LCALL.

19. SWAP

    Baca 4 bit (nibble) data pertama LSB dari akumulator. Perintah SWAP memungkinkan untuk menukar 4 bit rendah menjadi 4 bit tinggi dari akumulator (berlaku sebaliknya).

20. 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.

21. ORL

2.11.2 Instruksi MCS-51 yang Digunakan

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

1. EQU

   Konstanta Data yang dideklarasikan dengan nilai pada variabel tertentu.

2. BIT

   Konstanta Bit yang dideklarasikan dengan 1 bit data dari variabel tertentu.

3. ORG

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

4. CLR

   Memberikan nilai “nol” pada bit tertentu.

5. SETB

   Memberikan nilai “satu” pada bit tertentu.

6. CALL

   Instruksi melakukan lompatan dengan area sebesar 2 KByte.

7. LCALL

   Sama seperti instruksi CALL, hanya instruksi ini digunakan jika label yang dipanggil letaknya lebih dauh dari 2 Kbyte

8. MOV

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

9. MOVC

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

10. JMP

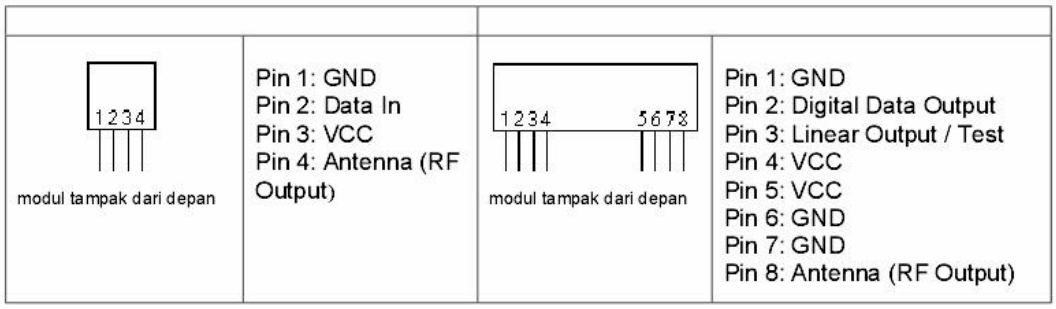
    Melakukan lompatan dan menjalankan program yang berada di alamat yang ditentukan oleh label tertentu.

11. SJMP

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

12. LJMP

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



Gambar 2.14. *Konfigurasi kaki TLP-434 dan RLP-434*

2.11 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

2.11.1 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*.

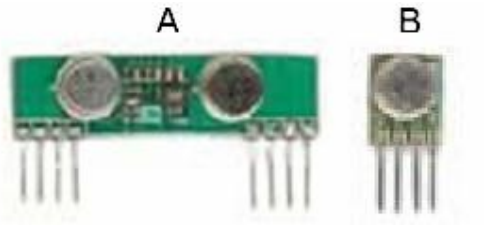
Adapun nama dan fungsi dari kaki-kaki pin pada Maxim232 adalah sebagai berikut:

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.

2.10 TLP-434A dan RLP-434A

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 2.13 *Bentuk fisik TLP-434 dan RLP-434*

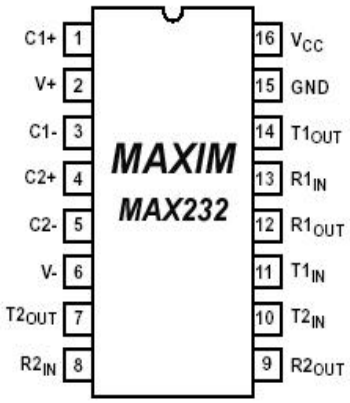
2.9.2 Maxim232

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 *buffer* 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.

2.9.3 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 2.4 dibawah ini.



Gambar 2.12 Konfigurasi pin MAXIM232

Tabel 2.5 Konfigurasi pin dan nama sinyal konektor serial DB9

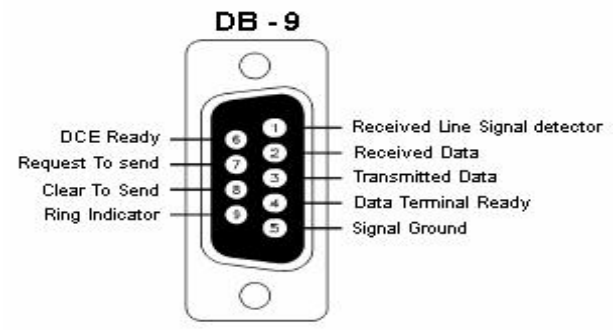
Nama Pin	**Nama Sinyal**	**Direction**	**Keterangan**
1	DCD	In	Data Carrier Detect/Received Line Signal Detect
2	DSR	In	Received Data
3	RxD	Out	Transmite Data
4	TxD	Out	Data Terminal Ready
5	DTR	-	Ground
6	GND	In	Data Set Ready
7	DSR	Out	Request to Send
8	CTS	In	Clear to Send
9	RI	In	Ring Indicator

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.

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.



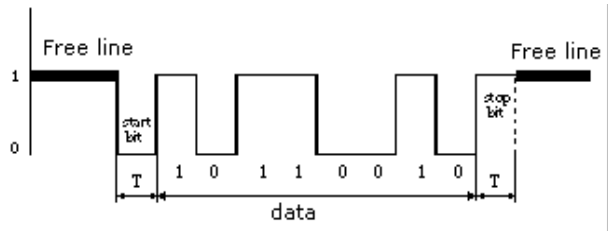
Gambar 2.11 *Susunan Port Serial 9 pin*

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 samapai –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.

(reciver). Pada PC kompatibel port serialnya termasuk jenis asinkron. Komunikasi data serial ini di kerjakan oleh oleh UART (Universal Asyncronous Reciver/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 2.10 *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., juga apakah data disertai dengan paritas genap,paritas ganjil atau tidak menggunakan paritas.

2.9.1 Port Serial/RS-232

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

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 vs *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 1200bps dengan frekuensi 11,0592 MHz, maka TH1 adalah 232 atau E8H.

2.9 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 di kirimkan bersama – sama dengan data serial, sedangkan komunikasi data serial asinkron, clock tidak di kirimkan bersama data serial, tetapi di bangkitkan secra sendiri-sendiri baik pada sisi pengirim (*transmitter)* mupun pada sisi penerima

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*

2.8.5 Baud Rate

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 Baud rate 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.

Tabel 2.3 lanjutan

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.

2.8.4 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

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.(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

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.

2.8.3 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

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.*

2.8.2 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.

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 dibawah.

Tabel 2.2 Fungsi-fungsi alternatif pada port 3

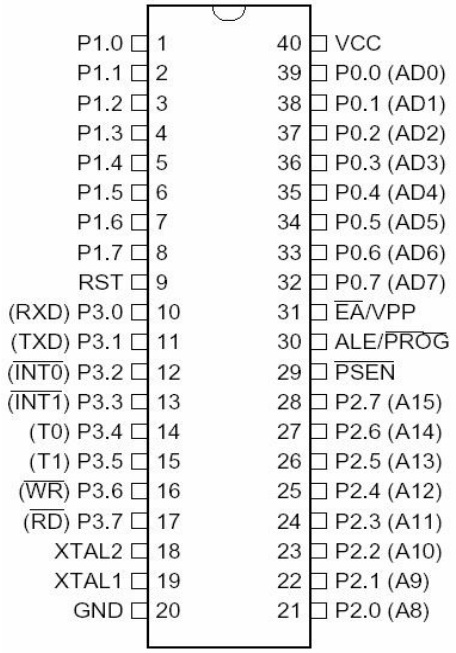
Kaki Port	Fungsi Alternatif
P3.0	RXD (port input serial)
P3.1	TXD (port output serial)
P3.2	INT0 (interupsi eksternal 0)
P3.3	INT1 (interupsi eksternal 1)
P3.4	T0 (input eksternal timer 0)
P3.5	T1 (input eksternal timer 1)
P3.6	WR (sinyal *write* pada data memori eksternal)
P3.7	RD (sinyal *read* pada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



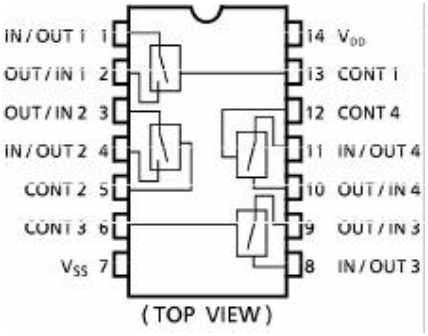
Gambar 2.9 Konfigurasi pin ATMEL AT89C51

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.



Gambar 2.8 konfigurasi pin IC 4066

2.8 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.

2.8.1 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 2.3 dibawah ini.

2.6.4 Bilangan Heksadesimal

Bilangan hexadesimal merupakan bilangan yang berbasis 16, dimana 1 angka pada hexadesimal mempunyai nilai 4 Bit, berikut angka decimal nya:

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

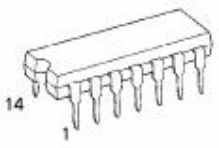
Contoh :

Bilangan hexadesimal dikonversikan ke bilangan biner

Bilangan (A) = 1010_{10}

2.7 IC 4066

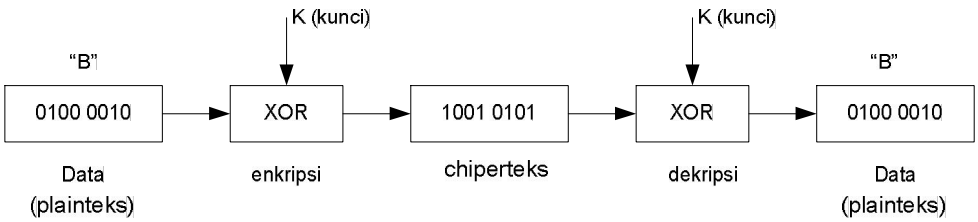
Penggunaan IC ini digunakan sebagai fungsi *switch,* dan IC ini merupakan IC yang berfungsi untuk *switch* transmisi atau mulltiplexing sinyal analog maupun sinyal digital. IC ini mempunyai kemampuan *switch* antara *crosstalk* sebesar 50 dB dan frekuensinya 0.9 Mhz resistansi dari IC ini memiliki keadaan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



Gambar 2.7 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 2.8.

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 2.6 Proses enkripsi dan dekripsi

2.6 Jenis Bilangan

Dalam suatu pemrograman ada beberapa jenis bilangan yang digunakan, yaitu:

a Bilangan Biner

b Bilangan Oktal

c Bilangan Desimal

d Bilangan Heksadecimal

2.6.1 Bilangan Biner

Bilangan biner adalah bilangan yang hanya terdiri atas 2 kemungkinan (berbasis dua), yaitu 0 dan 1. karena berbasis dua maka pengkonversian kedalam bentuk decimal adalah dengan mengalikan suku ke_N dengan 2^N.

Contoh:

Bilangan biner: $0111_2 = (0X2^3)+(1X2^2)+(1X2^1)+(1X2^0)$

$= 7_{10}$

2.6.2 Bilangan Oktal

Bilangan oktal adalah dengan basis 8 artinya angka yang dipakai antara 0-7, suatu bilangan oktal dapat dikonversi dalam bentuk decimal dengan mengalikan suku ke_N dengan 8^N.

Contoh:

Bilangan $128 = (1X8^1) + (2X8^0)$

$= 10_{10}$

2.6.3 Bilangan Desimal

Bilangan decimal adalah bilangan dengan berbasis 10, artinya yang dipakai 0-9.

Contoh :

Bilangan $24 = (1X10^1) + (4X10^0)$

$= 24_{10}$

2.5 Operator Logika

Operator logika (*logical operator*) adalah sebuah operator yang digunakan untuk membandingkan logika dalam persamaan aljabar boolean. Operator logika membandingkan dua buah nilai logika, yaitu benar dan salah (*True or False*). Operator logika yang digunakan untuk pemecahan masalah ini diantaranya operator XOR.

2.5.1 Operator XOR

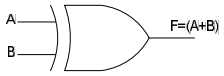
Operator XOR akan menghasilkan "1" atau "true" hanya jika salah satu inputnya bernilai "1" , atau "true"selain itu bernilai "0"atau "false"

Berikut tabel kebenaran

Tabel 2.1 tabel kebenaran operator XOR

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

Dibawah ini adalah simbol operator logika XOR



Gambar 2.5 simbol operator logika XOR

Dalam algoritma enkripsi dan dekripsi operator logika XOR merupakan inti dari proses penyandian. Sebagai contoh misal "B" adalah sebuah data lalu dienkripsi dengan kunci "K=1101 0111", maka dihasilkan "t" sebagai data enkripsi. Contoh berikut menggambarkan proses enkripsi dengan *stream cipher*.

kriptografer menggunakan algoritma yang sederhana, yang memungkinkan cipherteks dapat dipecahkan dengan mudah (melalui pengguaan ststistik, terkaan, intuisi dan sebagainya). Algoritma kriptografi modern dibuat sedemikian kompleks sehingga kriptanalis sangat sulit memecahan ciperteks tanpa mengetahui kunci.

Algoritma kriptografi modern umumnya beroperasi dalam mode bit ketimbang mode karakter (seperti yang dilakukan pada cipher subtitusi atau cipher transposisi dari algoritma kriptografi klasik). Operasi dalam mode bit berarti semua data dan informasi (baik kunci, palinteks maupun ciperteks) dinyatakan dalam rangkaian (*string)* bit biner, 0 dan 1, algoritma enkripsi dan dekripsi memproses semua data dan informasi dalam bentuk rangkaian bit yang menyatakan palainteks dienkripsi menjadi ciperteks dalam bentuk rangkaian bit, demikian sebaliknya, perkembangan kriptografi modern berbasis bit didorong oleh penggunaan komputer digital yang mempresentasikan data dalam bentuk biner.

2.4.4 Algoritma Enkripsi Dengan XOR

Algoritma enkripsi sederhana yang menggunakan XOR adalah dengan meng-xor-kan plainteks (*P*) dengan kunci (*K*) menghasilkan cipherteks:

$C = P \oplus K$

Karena men-XOR-kan nilai yang sama dua kali berturut-turut menghasilkan nilai semula, maka dekripsi menggunakan persamaan :

Contoh :

Plainteks	01100101	(Karakter 'e')
Kunci	00110101 ⊕	(Karakter '5')
Chipherteks	01010000	(Karakter 'P')
Kunci	00110101 ⊕	(Karakter '5')
Plainteks	01100101	(Karakter 'e')

Algoritma enkripsi XOR pada prinsipnya sama seperti Vigenere cipher dengan pengguanaan kunci yang berulang secara periodik. Setiap bit plainteks di-XOR-kan dengan setiap bit kunci.

Pertama kali kriptografi digunakan oleh militer untuk komunikasi rahasia yaitu sebelum perang dunia ke I. Saat itu orang-orang sudah mengetahui bahwa militer menggunakan pengkodean rahasia dalam berkomunikasi, Tapi hanya sedikit yang tahu tentang ilmu kriptografi.

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti keabsahan, integritas data, serta autentikasi data. Kriptografi tidak berarti hanya memberikan keamanan informasi saja, namun lebih ke arah teknik-tekniknya.

Dalam hal ini kriptografi dapat diartikan sebagai ilmu yang mempelajari pengacakan data elektronik dengan menggunakan algoritma tertentu sehingga tidak ada seorang pun yang dapat mengartikannya.

2.4.1 Enkripsi dan Dekripsi

. **Enkripsi** adalah suatu proses mengubah data menjadi sandi yang merupakan salah satu proses dari kriptografi. Data yang disandikan berupa file sebagai input dan dengan menggunakan suatu kunci, file tersebut di ubah menjadi file enkripsi yang tidak dapat dibaca. Adapun tujuan dari enkripsi ini adalah menyembunyikan data atau informasi dari orang yang tidak berhak.

Dekripsi adalah proses sebaliknya dari enkripsi yaitu mengembalikan sandi-sandi atau informasi yang telah diacak ke bentuk file aslinya dengan menggunakan kunci pula.

2.4.2 Kunci Asimetris

Dalam proses enkripsi dengan kunci asimetris terdapat dua buah kunci yang berbeda, yaitu *private key* dan *public* key. Keduanya dikenal sebagai *key pairs* (sepasang kunci). Private key adalah kunci yang dimiliki oleh seseorang dan tidak pernah diperlihatkan kepada public, *public key* adala kunci yang boleh diketahui orang lain karena kunci bersifat publik. Kedua kunci tersebut dapat digunakan untuk mengenkrisi data, tetapi untuk mendekripsi hanya bisa menggunaka kunci lawanya

2.4.3 Algoritma kriftografi modern

Kriftografi modern menggunakan gagasan dasar yang sama seperti kriftografi klasik (permutasi dan transposisi) tetapi penekanannya berbeda, pada kriftografi klasik,

2.3 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 terlalau jauh melalaui 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 Derau

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.

2.4 Kriptografi

Kriptografi (*cryptography*) adalah cabang ilmu matematika tentang persandian, ilmu ini membuat sebuah pesan menjadi aman.

Kategori aman ini meliputi :

a. *Confidentiality* (pesan tidak diketahui oleh pihak yang tidak berkepentingan)
b. *Authenticity* (penerima pesan bisa mengetahui secara pasti, siapa yang mengirim pesan itu).
c. *Integrity* (penerima yakin bahwa pesan yang diterimanya tidak dirubah saat dikirim).
d. *Non-repudiation* (pengirim pesan, tidak bisa membatah bahwa dia pernah mengirimkan pesan tersebut

b. *Phase Shift keying (PSK)*

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

bit "0" : 0^o

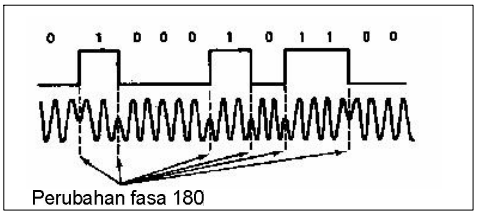
bit "1" : 180^o

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 2.2 Bentuk gelombang PSK

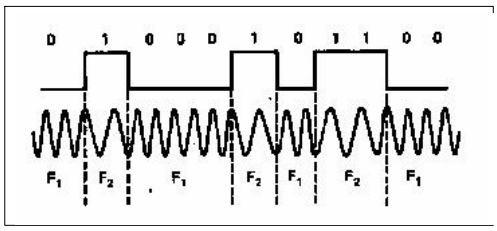
c. *Frekuqncy Shift Keying (FSK)*

Teknik ini mengubah frekuensi pembawa berdasarkan bit 1 dan bit 0

bit "0" : frekuensi f0

bit "1" : frekuensi f1

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



Gambar 2.3 Bentuk gelombang FSK

2.2.1 Teknik Modulasi Sinyal Analog

Teknik Modulasi Sinyal Analog :

1. *Amplitudo Modulation (AM)*
2. *Frekuancy Modualtion (FM)*
3. *Phase Modulation(PM)*

2.2.2 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 :

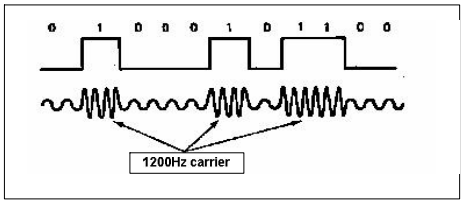
Bit ”0” : signal OFF

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 2.1 Bentuk gelombang ASK

BAB II
LANDASAN TEORI

2.1 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 (Amoranto 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, 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.

2.2 Sistem Komunikasi Radio untuk Transmisi Digital

Pada konsep ruang bebas dalam hambatan gelombang elektromagnetik (EM) berawal dari asumsi bahwa suatu *link* frekuensi radio (RF) propagasinya bebas dari segala gangguan. Dalam sistem komunikasi radio gelombang pembawa dipropagasikan dari pemancar dengan menggunakan antena pengirim. Di bagian antena pemancar atau sebaliknya mengkonversi gelombang elektromagnetik (EM) menjadi sinyal di bagian penerima.

Sinyal analog yang mengandung informasi asli disebut dengan *baseband signal*. Bila sinyal *baseband* ini memiliki frekuensi yang lebih rendah, maka sinyal ini harus digeser ke frekuensi yang lebih tinggi untuk memperoleh transmisi yang efisien. Hal ini disebut modulasi, pada sistem modulasi terdapat dua macam yaitu modulasi analog dan modulasi digital. **(**Msch schwartz, 1986**)**