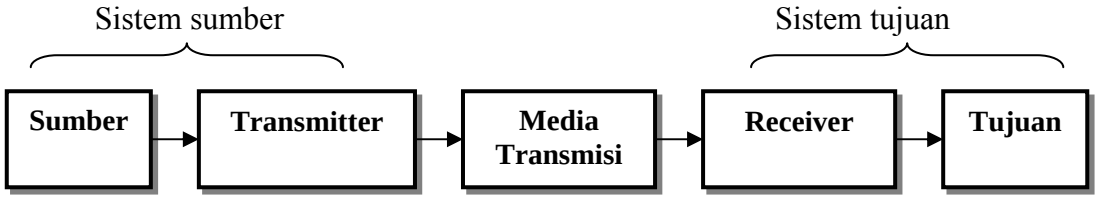
BAB II
LANDASAN TEORI

2.1 Komunikasi data

Prinsip dasar dari sistem komunikasi data adalah suatu cara untuk sebuah pertukaran data dari kedua pihak. Pada Gambar 2.1 dijelaskan sebuah contoh sistem komunikasi data sederhana.



Gambar 2.1 Blok diagram model komunikasi sederhana

Pada diagram model komunikasi data sederhana dapat dijelaskan :

1. **Sumber *(Source)*** : Alat ini membangkitkan data sehingga dapat ditransmisikan.
2. **Pengirim *(Transmitter)*** : Pada bagian ini data yang dibangkitkan dari sistem sumber tidak ditransmisikan secara langsung dalam bentuk aslinya namun pada sebuah *transmitter* cukup memindahkan informasi dengan menghasilkan sinyal elektromagnetik yang dapat ditransmisikan dengan beberapa sistem transmisi berurutan.
3. **Media Transmisi *(Transmission media)*** : Merupakan jalur transmisi tunggal yang menghubungkan antara sumber dan tujuan.
4. **Penerima *(Receiver)*** : Pada bagian ini sinyal dari pengirim diterima dari sistem transmisi dan memindahkan bentuk sinyal elekromagnetik menjadi digital yang dapat ditangkap oleh tujuan.
5. **Tujuan *(Destination)*** : Alat ini menerima data yang dihasilkan oleh penerima.

Dalam sebuah transmisi data dapat berupa *simplex* yaitu sinyal ditransmisikan hanya pada satu arah, *half duplex* yaitu kedua stasiun dapat mentransmisikan, namun hanya satu pada saat yang sama, *full duplex* yaitu kedua stasiun bisa mentransmisikan secara bersamaan.

Transmisi data terjadi antara *transmitter* dan *receiver* melalui beberapa media transmisi. Media transmisi dapat digolongkan sebagai transmisi dengan panduan *(guided media)* atau transmisi tanpa panduan *(unguided media).* Pada kedua hal tersebut

komunikasi berada dalam bentuk gelombang elektromagnetik . Dengan *guided media*, gelombang dikendalikan melalui jalur fisik, sedangkan pada *unguided media* menyediakan alat untuk mentransmisikan gelombang elektromagnetik namun tidak mengendalikannya.

2.2 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. 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. Seperti intermodulasi, efek dari intermodulasi akan menghasilkan sinyal-sinyal pada suatu frekuensi sehingga akan menghalangi sistem transmisi.

2.3 Sistem komunikasi radio untuk transmisi digital

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

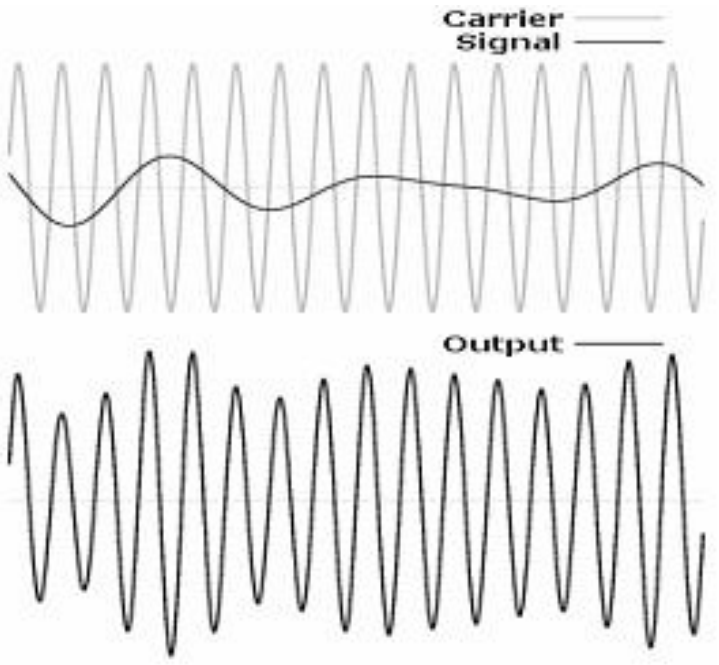
Sinyal analog yang mengandung informasi asli disebut dengan *baseband signal*. Bila sinyal *baseband* ini memiliki frekuensi yang lebih rendah, maka sinyal ini harus digeser ke frekuensi yang lebih tinggi untuk memperoleh transmisi yang efisien. Hal ini

dilakukan dengan mengubah-ubah amplitudo, frekuensi atau fasa dari suatu sinyal pembawa yang berfrekuensi lebih tinggi yang disebut sinyal pembawa (*carrier*). Proses ini disebut modulasi, modulasi didefinisikan sebagai proses yang mana beberapa karakteristik dari pembawa diubah-ubah berdasarkan gelombang pemodulasinya. Pada sistem modulasi terdapat dua macam yaitu modulasi analog dan modulasi digital.

Teknik modulasi sinyal analog :

- *Amplitude Modulation (AM)*
- *Frequency Modualtion (FM)*

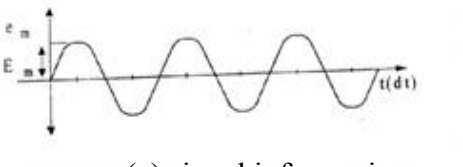
Amplitude Modulation (AM) merupakan proses modulasi yang mengubah amplitudo sinyal pembawa sesuai dengan sinyal pemodulasin atau sinyal informasinya. Sehingga dalam modulasi AM, frekuensi dan fasa yang dimiliki sinyal pembawa tetap, tetapi amplitudo sinyal pembawa berubah sesuai dengan informasi.



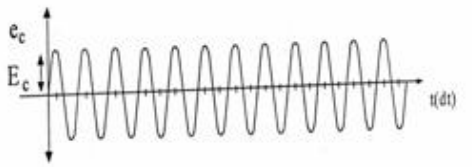
Gambar 2.2 Bentuk gelombang AM

Jika sinyal frekuensi rendah mengendalikan amplitudo dari sinyal frekuensi tinggi maka kita dapatkan modulasi amplitudo.

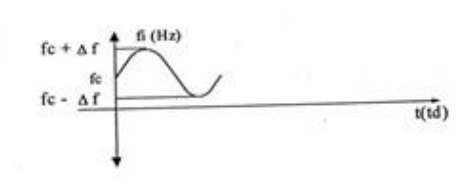
Frequency Modulation (FM) proses modulasi yaitu sinyal informasi ditumpangkan ke sinyal *carrier* atau sinyal pembawa, Modulasi Frekuensi merupakan suatu proses modulasi dengan cara mengubah frekuensi gelombang pembawa sinusoidal, yaitu dengan cara menyelipkan sinyal informasi pada gelombang pembawa tersebut.



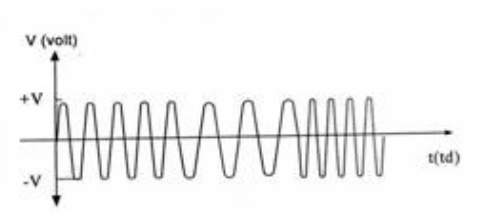
(a) sinyal informasi



(b)sinyal pembawa



(c) Frekuensi yang diubah



(d)gelombang pembawa yang termodulasi

Gambar 2.3 Bentuk gelombang FM

Sinyal informasi pada gambar 2.3(a) ditumpangkan pada sinyal pembawa gambar 2.3.(b) dengan cara mengubah lengkungan frekuensi sesaat fungsi waktu seperti, Gambar 2.3.(c) sehingga menghasilkan gelombang pembawa yang termodulasi, seperti pada gambar 2.3.(d)

Jika sinyal modulasi mengendalikan frekuensi pembawa maka kita dapatkan modulasi frekuensi. Jalur komunikasi radio biasanya dirancang untuk transmisi data digital, maka data digital tersebut harus terlebih dahulu dinyatakan kedalam sinyal analog sebagai *baseband signal.* Teknik untuk pengkodean sinyal digital ke dalam sinyal analog disebut dengan modulai digital. Beberapa teknik modulasi digital yang umum digunakan untuk data digital biner adalah:

- *Amplitudo Shift keying (ASK)*
- *Phase Shift keying (PSK)*
- *Frekquency Shift Keying (FSK)*

Pada ASK, modulasi ini menyatakan sinyal digital 1 sebagai suatu nilai tegangan tertentu (misalnya 1Volt) dan sinyal digital 0 sebagai sinyal digital dengan tegangan 0 Volt.

$$s(t) = \begin{cases} ACos(2\pi fct) \\ 0 \end{cases} \begin{matrix} biner1 \\ biner0 \end{matrix} \quad \text{.....(2.1)}$$

ASK tidak diterapkan secara luas untuk mengkonversi data biner pada PSTN, karena sinyalnya mudah terpengaruh oleh redaman, derau dan distorsi. Tetapi pada beberapa hal ASK masih digunakan terutama pada pada modulasi *hybrid* (misalnya ASK digabung dengan PSK). ASK umumnya digunakan untuk mentransmisikan sinyal digital pada serat optik, adanya cahaya menandakan adanya 1 elemen sinyal atau biner 1 dan bila tidak ada cahaya berarti biner 0. *Transmitter* laser pada umumnya memiliki arus bias yang tetap yang menyebabkan memancarkan cahaya dengan level yang rendah, oleh karena itu level yang rendah dapat dipakai untuk mewakili elemen sinyal 1 sedangkan untuk level yang tinggi mewakili elemen sinyal 0.

Pada PSK sinyal digital 1 sebagai suatu nilai tegangan tertentu dengan beda fasa tertentu pula (misalnya tegangan 1Volt dengan beda fasa 0o) dan sinyal digital 0 sebagai nilai tegangan yang sama (misalnya 1Volt dengan beda fasa yang berbeda, misalnya 180^o).Beda pasa yang diinginkan tergantung pada level modulasi yang digunakan, sebagai contoh level modulasi 4 pada QPSK adalah sebagai berikut:

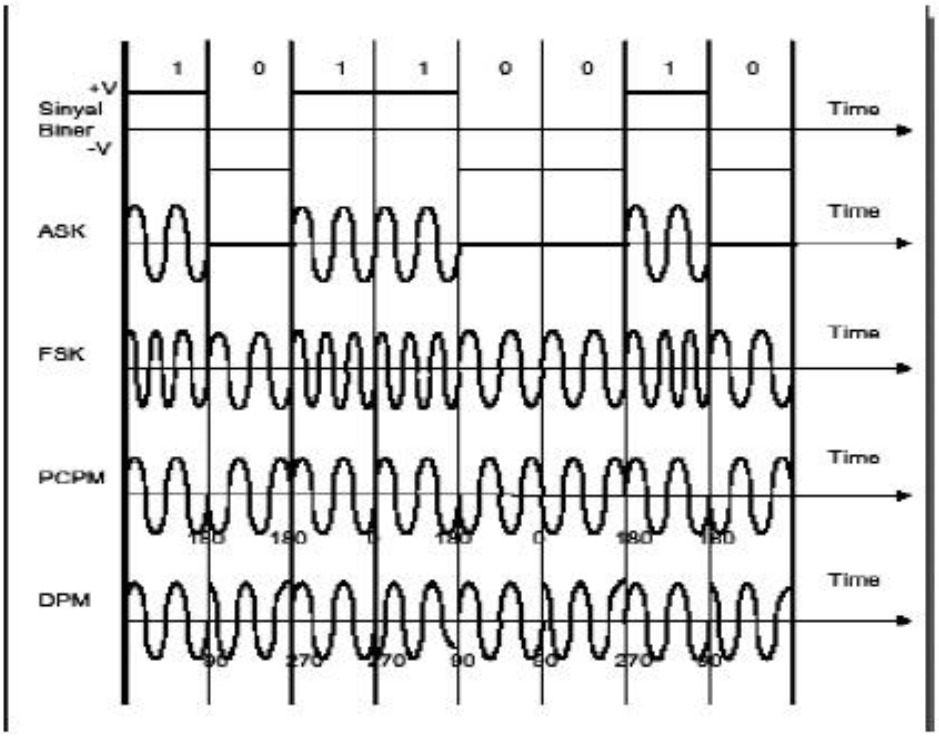
$$S(t) = \begin{Bmatrix} A\cos(2\pi fct + 45^0) \\ A\cos(2\pi fct + 135^0) \\ A\cos(2\pi fct + 225^0) \\ A\cos(2\pi fct + 315^0 \end{Bmatrix} \begin{matrix} biner\,01 \\ biner\,10 \\ biner\,00 \\ biner\,01 \end{matrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.2)$$

Modulasi *phase* memiliki dua jenis yaitu *Phase-coherent PM* atau yang umum disebut dengan *phase-shift-keying* (PSK), dipergunakan untuk memodulasi data biner 1 dan 0 dengan perbedaan fasa sebesar 180^o pada setiap perubahan data biner. Sedangkan yang lain disebut *differential-PM* yang dipergunakan untuk memodulasi dengan perbedaan fasa sebesar 90^o untuk kode biner 0 dan perbedaan fasa sebesar 270^o untuk biner 1. Secara alamiah pada PSK memiliki kelebihan untuk manambah efisiensi dalam *bandwidth* yang terbatas, maka setiap perubahan fasa dapat digunakan untuk mewakili lebih dari satu elemen sinyal, karena pergeseran fasa yang lebih kecil dari 180^o dapat dilakukan.

Pada FSK*(Frequency Shift Keying*) atau keying pergeseran frekuensi yaitu pengiriman sinyal melalui pergeseran frekuensi. Pada FSK sinyal digital 1 sebagai suatu nilai tegangan dengan frekuensi tertentu (misalnya f1=1200Hz) sementara sinyal digital 0 dinyatakan sebagai suatu nilai tegangan dengan frekuensi tertentu yang berbeda (misalnya f2=2200Hz).

$$S(t) = \begin{Bmatrix} A\cos f_1 t \\ A\cos f_2 t \end{Bmatrix} \begin{matrix} biner 0 \\ biner 1 \end{matrix} \quad \text{.....(2.3)}$$

Pada modulasi frekuensi yang rumit dapat dilakukan pada beberapa frekuensi sekaligus, dengan cara ini pengiriman data menjadi lebih efisien. Untuk melihat perbedaan bentuk sinyal dari ketiga teknik modulasi seperti diuraikan di atas, pada Gambar 2.4. diberikan beberapa contoh sinyal *carrier* yang dimodulasi oleh sinyal biner menggunakan ASK, FSK, *phase-coheren*t-PM dan *differential*-PM.



Gambar 2.4 Jenis modulasi digital

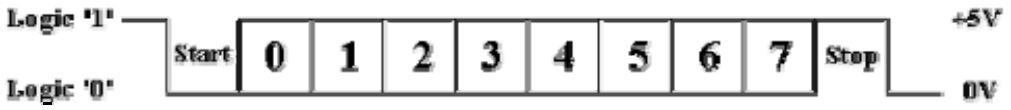
2.4 Modulator-Demodulator FSK

Alat untuk melakukan modulasi dan demodulasi disebut modem (*modulator-demodulator*). Modem memungkinkan dua buah sistem elektronik digital untuk berkomunikasi menggunakan saluran transmisi. Alat yang menggunakan *port* serial untuk berkomunikasi dibagi menjadi 2 kategori, yaitu DTE *(Data Terminal Equipment)* dan *(Data Communication Equipment),* modem adalah perangkat DCE, perangkat yang berhubungan langsung dengan medium transmisi, sedangkan perangkat DTE contohnya adalah terminal atau komputer.

Modem FSK umumnya memiliki kecepatan 300 bps sampai 1200 bps dan sering digunakan untuk komunikasi data antar komputer dan pada PSTN yang memiliki rangkaian *switching* yang sederhana dan memiliki *bandwidth* yang rendah.

2.5 Pengiriman data tak sinkron

Pengiriman data tak sinkron, setiap karakter dikirimkan sebagai suatu kesatuan bebas, yang berarti bahwa waktu antara pengiriman sebagai bit terakhir dari sebuah karakter dan bit pertama dari karakter berikutnya tidak tetap. Pengiriman data tak sinkron lebih sederhana dibandingkan pengiriman sinkron, karena hanya di dalam penerima dan tetap dijaga agar sesuai dengan detak pengiriman yang menggunakan bit awal (*start bit*) dan bit akhir (*stop bit*) yang dikirim dengan setiap karakter.



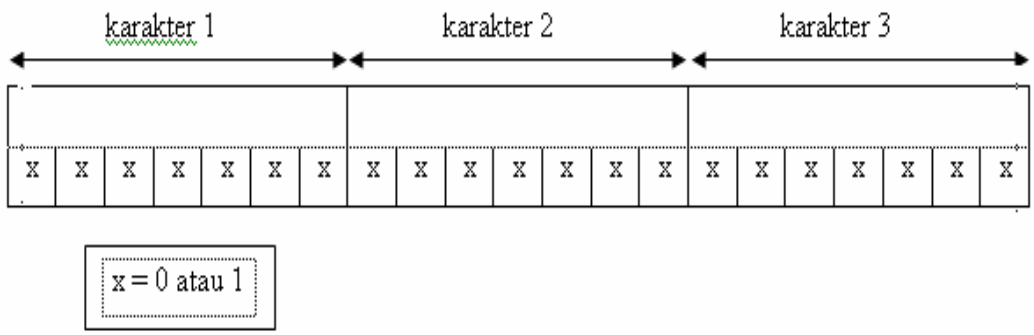
Gambar 2.5 Aliran data tak sinkron

Pengiriman data tak sinkron banyak dipakai karena sederhana dan murah. Tetapi hanya cocok untuk rangkaian data berkecepatan rendah karena dua alasan :

1. Efisiensi pengiriman menjadi berkurang dengan bertambahnya panjang kabel.
2. Detak penerima yang bekerja bebas hanya akan memenuhi persyaratan sinkronisasi apabila bekerja pada kecepatan rendah.

2.6 Pengiriman data sinkron

Pada pengiriman data sinkron sejumlah blok data dikirimkan secara kontinu tanpa bit awal atau bit akhir. Detak pada penerima dioperasikan secara berulang-ulang dan dikunci agar sesuai dengan detak pada pengirim.



Gambar 2. 6 Aliran data sinkron

Untuk mendapatkan keadaan yang sesuai, informasi pendetakan harus dikirimkan lewat jalur bersama-sama dengan data memanfaatkan metode penyandian tertentu sehingga informasi pendetakan dapat diikutsertakan, atau dengan menggunakan modem yang menyandikan informasi pendetakan selama proses modulasi.

Penerima harus memulai pencacah pada tengah-tengah bit pertama dari karakter pertama, jika akan timbul kesalahan pada isyarat yang diterima. Setelah penyesuaian bit, penerima harus tahu pada kelompok mana bit tersebut akan membentuk karakter (penyesuaian karakter). Penerima harus mamantau data yang diterima setiap bit sampai mengenali pola karakter sinkronisasi. Dengan cara ini penerima dapat mengetahui himpunan bit mana yang membentuk karakter pertama dikirimkan. Maka karakter berikut dengan mudah dapat dikenali.

2.7 Perbedaan pengiriman sinkron dan tak sinkron

Umumnya pengiriman tak sinkron tidak mahal. Setiap *byte* yang diterima dibedakan dengan bit awal dan bit akhir, karena detak penerima selalu dimulai kembali setelah satu karakter diterima atau dengan kata lain detak panerima hanya akan berjalan pada saat ada isyarat data yang akan diterima dan hanya perlu pada keadaan sinkron untuk selang waktu 8 bit, maka penyesuaian bit juga bukan merupakan persoalan besar.

Pengiriman sinkron lebih mahal dibandingkan pengiriman tak sinkron, tetapi dapat bekerja pada laju yang lebih tinggi. Karena data biasanya dikirim tanpa pembatas, diperlukan adanya *buffering* baik pada pengirim maupun penerima. Laju pengiriman dapat diubah dengan mengubah detak pengiriman dan kecepatan data pada waktu yang sama.

2.8 Mikrokontroler AT89S51

Mikrokontroler AT89S51 termasuk dalam *MCS-51*^{TM} dari Intel. Sebuah mikrokontroler tidak dapat bekerja bila tidak diberi program kepadanya. Program tersebut memberitahukan mikrokontroler apa yang harus dilakukan. Salah satu keunggulan dari AT89S51 adalah dapat diisi ulang dengan program lain sebanyak 1000 kali pengisian. Instruktur-instruktur perangkat lunak berbeda untuk masing-masing jenis mikrokomtroler. Instruksi-instruksi hanya dapat dipahami oleh jenis mikrokontroler yang bersangkutan. Instruksi-instruksi dikenal sebagai bahasa pemrogrman system mikrokontroler.

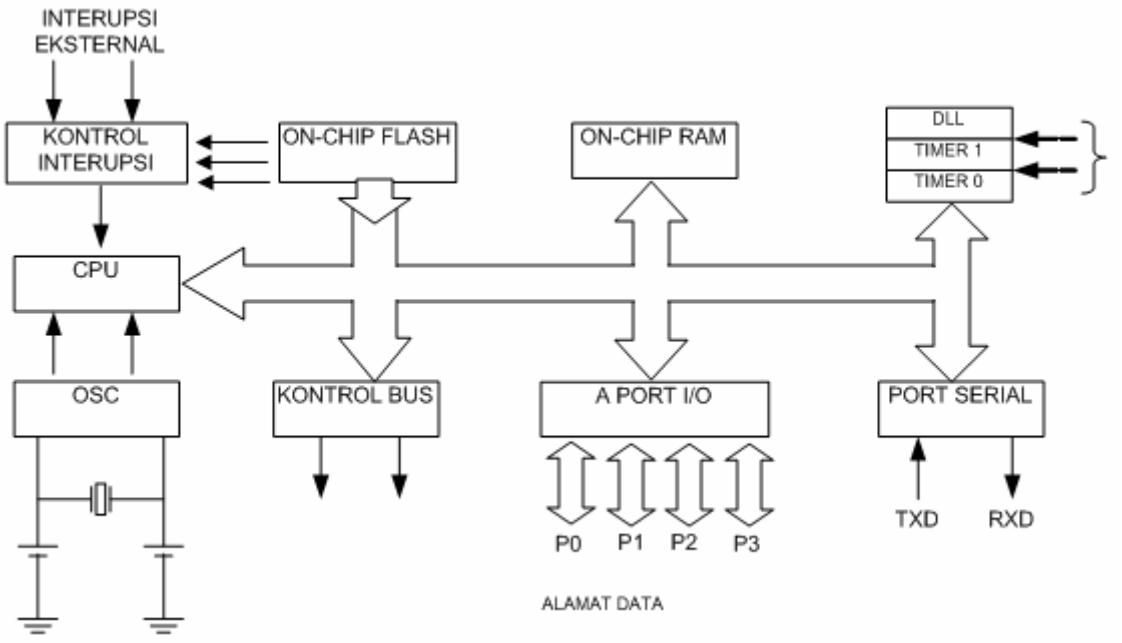
Sebuah mikrokontroler tidak dapat memahami instruksi-instruksi yang berlaku pada mikrokontroler lain. Sebagai contoh, mikrokontroler buatan Intel dengan mikrokontroler buatan Motorolla memiliki perangkat instruksi yang berbeda.

2.8.1 Karakteristik Mikrokontroler AT89S51

Mikrokontroler AT89S51 memiliki beberapa fasilitas, diantaranya sebagai berikut:

1. Sebuah CPU (*Central Prossesing Unit*) 8 bit yang termasuk keluarga Osilator *internal* dan rangkaian *timer*.
2. Flash memori 4 Kbyte.
3. RAM internal 128 byte.
4. Empat buah *programmable* port I/O, masing-masing terdiri atas 8 buah jalur I/O.
5. Lima buah jalur interupsi (2 buah interupsi eksternal dan 3 buah internal).
6. Sebuah port serial dengan kontrol *serial full duplex* UART.
7. Kemampuan melaksanakan operasi perkalian, pembagian dan operasi boolean (bit).
8. Kecepatan pelaksanaan interuksi per siklus 1 mikrodetik pada frekuensi clock 1 MHz.

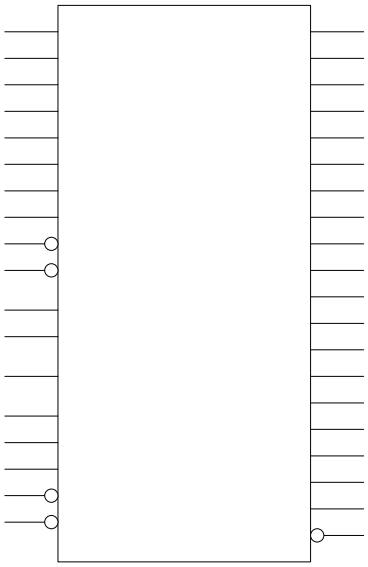
Dengan fasilitas seperti diatas, pembuatan alat menggunakan AT89S51 menjadi lebih sederhana dan tidak memerlukan IC pendukung yang banyak. Agar lebih jelasnya dapat dilihat pada gambar 2.7 dibawah ini.



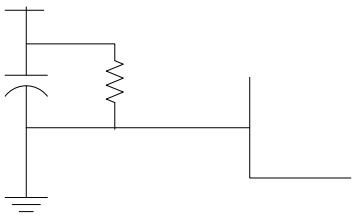
Gambar 2.7 Diagram Blok Mikrokontroler AT89S51

2.8.2 Deskripsi Pin-Pin AT89S51

Susunan pin-pin mikrokontroler AT89S51 memperlihatkan pada gambar 2.8 dibawah ini. Penjelasan dari masing-masing pin adalah sebagai berikut :



Gambar 2.8 Diagram Pin Mikrokontroler AT89S51



Gambar 2.9 Rangkaian Reset

1. Pin 1 sampai 8 (Port 1) merupakan port pararel 8 bit dua arah (*output-input*) yang dapat digunakan untuk berbagai keperluan (*general purpose*).
2. Pin 9 (Reset) adalah input reset (aktif tinggi). Pulsa transisi dari rendah ke tinggi akan me-reset AT89S51. pin ini dihubungkan dengan rangkaian power on reset seperti ditunjukan pada gambar 2.3 diatas.
3. Pin 10 sampai 17 (Port 3) adalah port pararel 8 bit dua arah (output-input) yang memiliki fungsi pengganti. Fungsi pengganti meliputi TxD (*Transmision Data*), RxD (*Received Data*), Int 0 (*Interupsi 0*), Int 1 (*Interupsi 1*), T0 (*Timer 0*), T1 (*Timer 1*), WR (*Write*) dan RD (*Read*). Bila fungsi penggnati tidak dipakai, pin-pin ini dapat digunakan sebagai port pararel 8 bit serba guna.
4. Pin 18 dan 19 (XTAL 1 dan XTAL 2) adalah pin input kristal, yang merupakan input clock bagi rangkaian osilator internal.

5. Pin 20 (*Ground*) dihubungkan ke Vss atau Ground.
6. Pin 21 sampai 28 (Port 2) adalah port pararel 2 selebar 8 bit dua arah. Port 2 ini mengirimkan byte alamat bila dilakukan pengaksesan memori eksternal.
7. Pin 29 adalah pin PSEN (*Program Strobe Enable*) yang merupakan sinyal pengontrol yang membolehkan program memori eksternal masuk ke dalam bus selama proses pemberian/pengambilan instruksi.
8. Pin 30 adalah pin output ALE (*Address Latch Enable*) yang digunakan untuk menahan alamat memori eksternal selama pelaksanaan instruksi.
9. Pin 31 (EA). Bila pin ini diberi logika tinggi, maka mikrokontroler akan melaksanakan instruksi dari ROM ketika isi program counter kurang dari 4096. Bila diberi logika rendah, maka mikrokontroler akan melaksanakan seluruh instruksi dari memori program diluar.
10. Pin 32 sampai 39 (*Port 0*) merupakan port pararel 8 bit open drain dua arah. Bila diberi logika rendah, mikrokontroler akan melaksanakan seluruh instruksi dari memori program luar.
11. Pin 40 (Vcc) dihubungkan ke Vcc (+5 Volt).

2.8.3 Organisasi Memori

Semua mikrokontroler dalam keluarga MCS-51 memiliki pembagian ruang alamat (*address space*) untuk program dan data. Pemisahan memori program dan memori data membolehkan memori data untuk diakses oleh alamat 8 bit. Meskipun demikain, alamat data memori 16 bit dapat dihasilkan melalui register DPTR (Data Pointer Register).

Memori program hanya dapat dibaca tidak bisa ditulis, karena disimpan dalam *Flash Memori*. Memori program sebesar 64 Kbyte dapat dimasukkan dalam EPROM eksternal. Seperti tampak pada gambar 2.10

Sinyal yang membolehkan pembacaan dari memori program eksternal adalah pin PSEN. Pada gambar 2.11 memperlihatkan memori data yang terletak pada ruang alamat terpisah dari memori program. RAM ekternal 64 Kbyte dapat dialamati dalam ruang memori data eksternal. CPU menghasilkan sinyal *read* dan *write* selama menghubungi memor data eksternal.

Mikrokontroler AT89S51 memiliki 5 buah ruang alamat, yaitu :

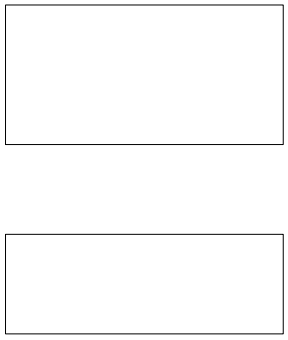
(a) Ruang alamat kode (*Code Address Space*) sebanyak 64 Kbyte, yang seluruhnya merupakan ruang alamat kode eksternal.

(b) Ruang alamat memori data internal yang dapat dialamati secara langsung, yang terdiri atas :
   1. RAM sebanyak 128 byte
   2. Hardware register sebanyak 128 byte

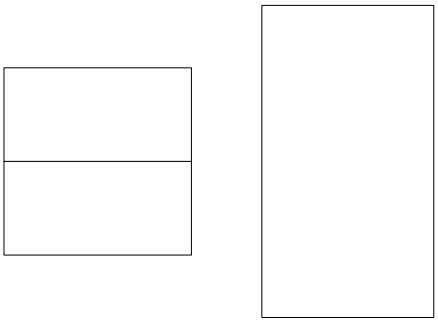
(c) Ruang alamat memori data internal yang dialamati secara tidak langsung sebanyak 128 byte, seluruhnya diakses dengan pengalamatan tidak langsung.

(d) Ruang alamat memori data eksternal sebanyak 64 Kbyte yang dapat ditambahkan oleh pemakai.

(e) Ruang alamat bit. Dapat diakses dengan pengalamatan langsung.



Gambar 2.10 Struktur Program Memori AT89S51



Gambar 2.11 Struktur Data Memori AT89S51

2.8.4 Special Function Register (SFR)

SFR berisi register-register dengan fungsi tertentu yang disediakan oleh mikrokontroler seperti timer dan lain-lainnya. AT89S51 memiliki 21 SFR yang terletak pada memori 80H-FFH. Masing-masing ditunjukkan pada tabel 2.1 yang meliputi simbol, nama dan alamatnya.

Tabel 2.1 *Spesial Function Register*

Simbol	Nama	Alamat
ACC	Akumulator	E0H
B	*B register*	F0H
PSW	*Program Status Word*	D0H
SP	*Stack Pointer*	81H
DPTR	*Data Pointer* 16 bit *DPL byte* rendah *DPH byte* tinggi	 82H 83H
P0	*Port 0*	80H
P1	*Port 1*	90H
P2	*Port 2*	A0H
P3	*Port 3*	B0H
IP	Interupt Priority Control	B8H
IE	Interupt Enable Control	A8H
TMOD	Timer/Counter Mode Control	89H
TCON	Timer/Counter Control	88H
TH 0	*Timer/Counter High Low byte*	8CH
TL 0	*Timer/Counter Low byte*	8AH
TH 1	*Timer/Counter High byte*	8DH
TH 1	*Timer/Counter Low byte*	8BH
SCON	*Serial Control*	98H
SBUF	*Serial Data Buffer*	99H
PCON	*Power Control*	87H

2.8.5 Timer AT89S51

Mikrokontroler AT89S51 mempunyai dua buah timer, yaitu Timer 0 dan Timer 1, setiap timer terdiri dari 16 bit timer yang tersimpan dalam dua buah register yaitu THx untuk *Timer High Byte* dan TLx untuk *Timer Low Byte* yang keduanya dapat berfungsi sebagai *counter* maupun sebagai timer. Secara fisik timer juga merupakan rangkaian T flip-flop yang dapat diaktifkan dan dinonaktifkan setiap saat/. Perbedaan keduanya terletak pada sumber *clock* dan aplikasinya.

Timer mempunyai sumber *clock* dengan frekuensi tertentu yang sudah pasti sedangkan *counter* mendapat sumber *clock* dari pulsa yang hendak dihitung jumlahnya.

Aplikasi dari timer atau pewaktu biasa digunakan untuk aplikasi menghitung lamanya suatu kejadian yang terjadi sedangkan *counter* atau penghitung biasa digunakan untuk aplikasi menghitung jumlah kejadian yang terjadi dalam periode tertentu. Perilaku dari register THx dan TLx diatur oleh register TMOD dan TCON. Timer dapat diaktifkan melalui perangkat keras maupun perangkat lunak.

Periode waktu timer/counter dapat dihitung menggunakan rumus 2.1 dan 2.2 sebagai berikut.

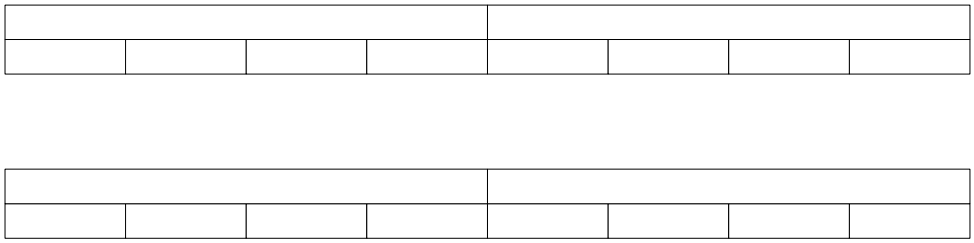
Sebagai timer/counter 8 bit

$$T = (255 - TLx) * \frac{12}{frekuensiXTAL}\ \mu s \quad \textbf{.....(2.1)}$$

Sebagai timer/counter 16 bit

$$T = (65535 - THxTLx) * \frac{12}{frekuensiXTAL}\ \mu s \quad \textbf{.....(2.2)}$$

Di mana : THx = isi register TH0 atau TH1 dan TLx = isi register TL0 atau TL1.



Gambar 2.12 Register TCON dan TMOD

Pengontrolan kerja *timer/counter* diatur oleh register TCON. Register ini bersifat *bit addresable* sehingga bit TF1 dapat disebut TCON.7 dan seterusnya hingga bit IT0 sebagai TCON.0. Register ini hanya mempunyai 4 bit saja yang berhubungan dengan timer seperti diperlihatkan gambar 2.12 dan dijelaskan pada tabel 2.2

Tabel 2.2 Fungsi bit register TCON yang berhubungan dengan *timer*

Nama Bit	Fungsi
TF1	*Timer* 1 *overflow flag* yang akan diset jika *timer overflow*
TR1	Membuat timer 1 aktif (*set*) dan nonaktif (*clear*)
TF0	*Timer* 0 *overflow flag* yang akan diset jika *timer overflow*
TR0	Membuat timer 0 aktif (*set)* dan nonaktif (*clear*)

Register TMOD berfungsi untuk pemilihan mode operasi *timer/counter* dengan fungsi setiap bitnya adalah sebagai berikut :

Gate : Pada saat TRx = 1, *timer* akan berjalan tanpa memperlihatkan nilai pada Gate (timer dikontrol software).

C/T : Pemilihan fungsi timer (0) atau counter (1).

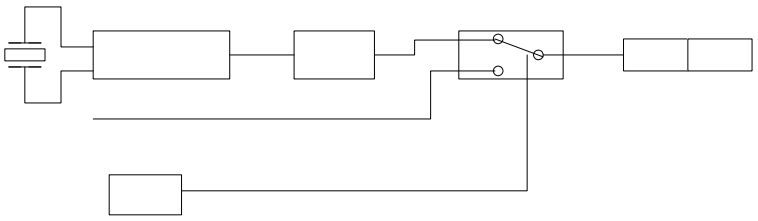
M1 & M0 : Untuk memilih mode timer dengan variasi seperti pada tabel 2.3.

Tabel 2.3 *Mode Timer*

M1	M0	Mode	Operasi
0	0	0	Timer 13 bit
0	1	1	Timer/Counter 16 bit
1	0	2	Timer 8 bit di mana nilai timer tersimpan pada TLx. Register THx berisi nilai isi ulang yang akan dikirim ke TLx setiap *overflow*.
1	1	3	Pada mode ini, AT89S51 bagaikan memiliki 3 buah timer. Timer 0 terpisah menjadi 2 buah timer 8 bit (TL0-TF0 dan TH0-TF1) dan timer tetap 16 bit.

Prinsip Kerja Timer

Pada gambar 2.13 *Timer* mempunyai dua buah sumber *clock* untuk beroperasi, yaitu sumber *clock* internal dan sumber *clock* eksternal. Jika *timer* menggunakan sumber *clock* eksternal, maka bit C/T harus di-set atau berkondisi *high*, saklar akan menghubungkan sumber *clock timer* ke pin Tx (T0 untuk timer 0, T1 untuk timer 1). Apabila sumber *clock* internal digunakan, *input clock* berasal dari osilator yang telah dibagi 12. Maka bit C/T harus di-clear atau berkondisi *low* sehingga saklar akan menghubungkan sumber *clock timer* ke osilator yang telah dibagi 12.



Gambar 2.13 Operasi *Timer*

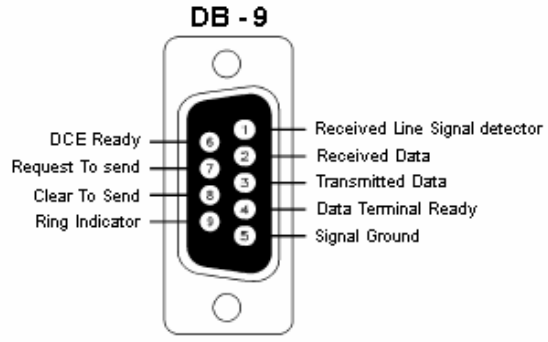
2.9 Port serial/RS-232

Port serial lebih sulit ditangani daripada *port* paralel karena peralatan yang dihubungkan ke *port* serial harus berkomunikasi 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 Asynchronus 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 memiliki 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 kabel dibandingkan dengan transmisi paralel.
3. *Port* serial memungkinkan untuk berkomunikasi dengan menggunakan *Infra Red.*



Gambar 2.14 Konfigurasi *port* serial *male*

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 2.4 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	RxD	In	Received Data
3	TxD	Out	Transmite Data
4	DTR	Out	Data Terminal Ready
5	GND	-	Ground
6	DSR	In	Data Set Ready
7	RTS	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.
9. *DCE Ready,* sinyal aktif pada saluran ini menunjukan bahwa DCE sudah siap.

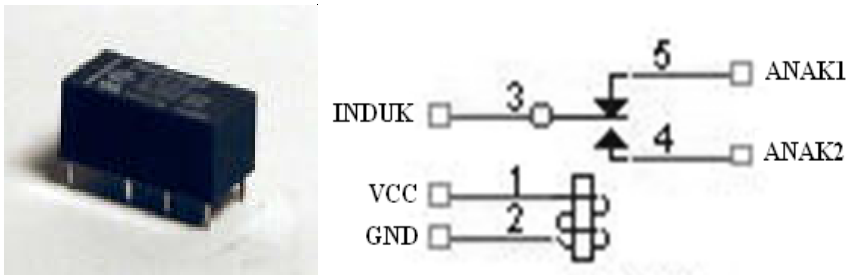
2.10 Relay

Relay adalah komonen listrik yang berfungsi untuk membuka dan menutup kontak listrik yang berdasarkan prinsip elektromagnetik yang mempengaruhi komponen lain dalam rangkaian yang sama atau rangkaian yang lain. Dengan bantuan relay kita dapat menutup dan memutuskan kontak dari jarak jauh dengan arus dari relay coil. Relay terdiri dari sebuah kumparan dan inti, sebuah saklar *Normally Open* (NO) dan sebuah saklar *Normally Closed* (NC). Pada saat output dari transistor yang tadinya terbuka

menjadi tertutup sehingga komponen yang dihubungkan menjadi berubah keadaan. Jadi saklar ini bekerja berdasarkan arus yang mengalir pada kumparan.

Sifat – sifat relay yang diperlukan, jika hendak memilih relay yang baik :

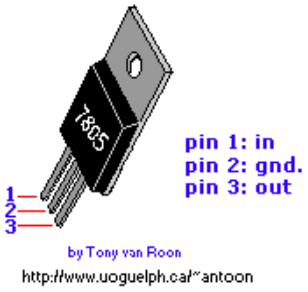
1. Besar hambatan tergantung dari tebal kawat dan banyaknya lilitan yang digunakan. Besarnya harga hambatan ini antara 1 sampai 50KΩ,
2. Daya yang diperlukan oleh relay sama dengan perkalian antara besar tegangan dan arus yang digunakan oleh relay.
3. Besarnya arus yang diperlukan oleh relay biasanya ditentukan oleh pabrik. Relay dengan hambatan kecil memerlukan arus yang besar.
4. Tegangan yang diperlukan oleh relay mengikuti hokum Ohm, V=I.R. Besar tegangan sama dengan perkalian antara besar arus dengan hambatan relay.



Gambar 2.15 Relay

2.11 LM7805

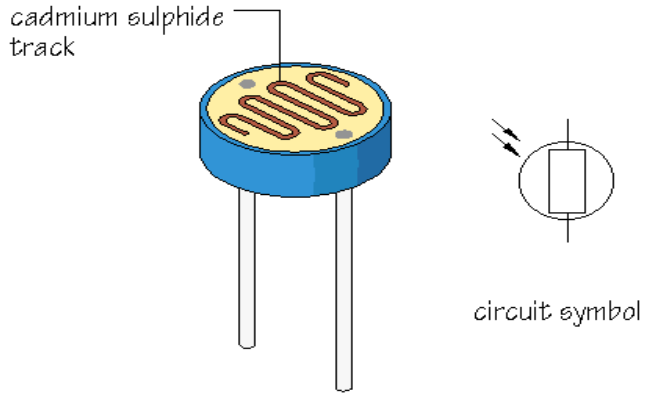
IC ini mempunyai tiga kaki yang digunakan sebagai komponen pendukung dari Vcc untuk menghasilkan tegangan 5V. IC *regulator* ini berfungsi untuk menstabilkan tegangan 5V dan dapat bekerja dengan baik jika tegangan *input* (Vin) lebih besar dari pada tegangan *output* (Vout). Biasanya perbedaan tegangan *input* dengan *output* yang direkomendasikan tertera pada *datasheet* komponen tersebut. Contoh LM7805 diperlihatkan pada gambar 2.16.



Gambar 2.16 IC LM7805

2.12 Light Dependent Resistor (LDR)

LDR atau light Dependent Resistor adalah salah satu jenis resistor yang nilai hambatannya dipengaruhi oleh cahaya yang diterima olehnya. LDR dibuat dari Cadmium Sulfida yang peka terhadap cahaya. Seperti yang telah diketahui bahwa cahaya memiliki dua sifat yang berbeda yaitu sebagai gelombang elektromagnetik dan foton/partikel energi (dualisme cahaya). Saat cahaya menerangi LDR, foton akan menabrak ikatan Cadmium Sulfida dan melepaskan elektron. Semakin besar intensitas cahaya yang datang, semakin banyak elektron yang terlepas dari ikatan. Sehingga hambatan LDR akan turun saat cahaya meneranginya.



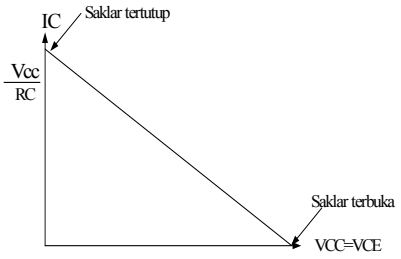
(a) Bentuk LDR (b) Simbol LDR

Gambar 2.17. Light Dependent Resistor (LDR)

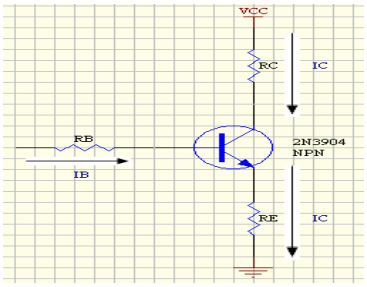
LDR akan mempunyai hambatan yang sangat besar saat tidak ada cahaya yang mengenainya (gelap). Dalam kondisi ini hambatan LDR mampu mencapai 1MΩ. Akan tetapi saat terkena sinar, hambatan LDR akan turun secara drastis hingga nilai beberapa puluh ohm saja. Dalam aplikasi, dianjurkan untuk mengukur nilai R_{max} dan R_{min} dari LDR. Pengukuran R_{max}dilakukan saat gelap (”agak gelap”) dan pengukuran R_{min} dilakukan saat terang.

2.13 Transistor sebagai saklar

Sebuah transistor akan bekerja sebagai saklar dengan mengoperasikan transistor pada salah satu dari saturasi atau titik sumbat. Jika transistor berada dalam keadaan saturasi, transistor tersebut seperti sebuah saklar yang tertutup dari kolektor ke emitter.Jika transistor tersumbat (*cut off*) transistor seperti sebuah saklar terbuka.



(a)



(b)

Gambar 2.18. (a) Garis beban dc (b) rangkaian Transistor Garis beban dc

Pada titik sumbat arus basis nol dan arus kolektor sangat kecil sehingga dapat diabaikan (hanya ada arus bocor I_{ceo}). Digunakan suatu pendekatan bahwa tegangan kolektor emitter sama dengan V_{CC}.

$V_{CC} = V_{CE}$ .....(2.4)

Pada titik saturasi arus basis sama dengan $I_{B(sat)}$ dan arus kolektor adalah maksismum. Digunakan suatu pendekatan arus kolektor pada saturasi adalah :

$$I_{C(sat)} = \frac{V_{CC}}{R_C} \quad \text{.....(2.5)}$$

$$I_{B(sat)} = \frac{I_{C(sat)}}{\beta} \quad \text{.....(2.6)}$$

Keterangan : β = 200 ($V_{CE(sat)}$ dari datasheet transistor jenis NPN tipe 2N3904)

Rangkaian ini menggunakan transistor NPN tipe 2N3904 yang akan merubah logika 1 menjadi logika 0 sekaligus memperkuat arusnya. Jika input (Basis) diberi logika 1 maka arus positif (Emiter / V_{CC}) akan mengalir ke output (kolektor) menyebabkan output berlogika 1. Jika input diberi logika 0 maka arus positif tidakdialirkan.

2.14 Pengaturan Baudrate

Untuk transmisi data antara pengiriman dan penerimaan harus menggunakan laju kecepatan (*Baudrate*) yang sama agar terjadi kesinkronan antara data yang dikirim dan data yang diterima.

Timer1 dapat digunakan sebagai pewaktu untuk mengatur *baudrate* pada komunikasi serial. Lama pengiriman tiap bit data = timer1 *overflow* x 32.

Rumus frekuensi osilator dengan *baudrate*:

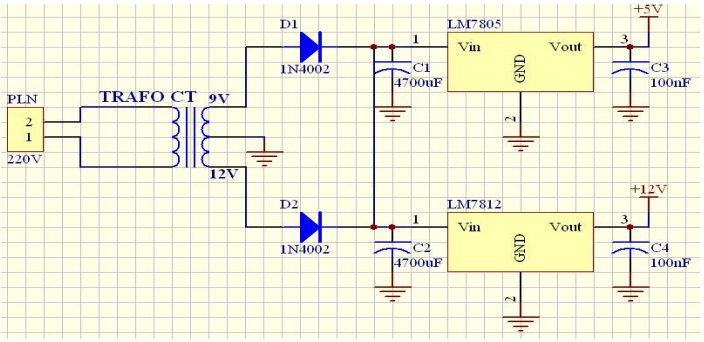
$$\frac{12x(FFh - TH1)}{F_{OSC}} = \frac{1}{Baudratex32} \quad \text{.....(2.7)}$$

Rumus nilai register TH1 :

$$TH1 = 256 - \frac{F_{osc}}{(12 x baudrate x 32)} \quad \text{.....(2.8)}$$

2.15 Catu Daya

Perangkat elektronik mestinya dicatu oleh suplai arus searah DC yang stabil agar dapat bekerja dengan baik. Baterai atau accu adalah sumber catu daya DC yang paling baik. Namun untuk aplikasi yang membutuhkan catu daya lebih besar, sumber dari diatas (baterai dan accu) tidak cukup. Butuh sumber daya yang lebih besar yaitu sumber bolak-balik AC dari pembangkit listrik. Untuk itu diperlukan catu daya yang digunakan untuk mengubah arus AC menjadi DC. Transformator diperlukan untuk menurunkan tegangan AC dari jala-jala listrik pada kumparan primer menjadi tegangan AC yang lebih kecil pada kumparan sekunder. Untuk mendapatkan arus yang searah diperlukan dioda, dioda berperan hanya untuk meneruskan tegangan positif ke regulator. Regulator berfungsi sebagai komponen aktif yang dapat meregulasi tegangan keluar agar stabil.



Gambar 2.19 Catu daya

2.16 Software Visual Basic 6.0

Perancangan software dititik beratkan pada pembangunan sebuah program interface yang *user friendly* dan yang terpenting adalah *software* harus mampu berkomunikasi dengan *hardware* sehingga dapat menyampaikan informasi yang sesuai. Pada sistem ini software yang digunakan adalah *Visual Basic* 6.0.

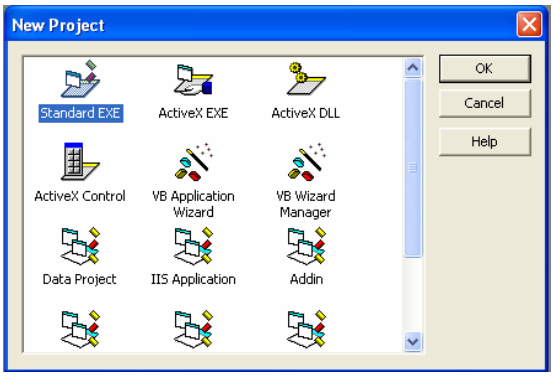
Bahasa pemrograman adalah bahasa yang dimengerti oelh *object* untuk melakukan tugas-tugas tertentu, salah satu contoh bahasa *Visual Basic*. Bahasa pemrograman *Visual Basic* yang dikembangkan oleh *Microsoft* sejak tahun 1991 merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (*Baginners All-purpose Symbolic Instruction Code*) yang dikembangkan pada era 1950-an. *Visual Basic* merupakan salah satu *Development Tool* yaitu alat bantu untuk membuat berbagai

macam program *object*, khususnya yang menggunakan sistem operasi *Windows*, juga salah satu bahasa pemrograman *object* yang mendukung *object* (*Object Oriented Programming* = OOP). Dalam pemrograman berbasis obyek (OOP) kita perlu mengenal istilah *object*, *property*, *method* dan *event.* Berikut adalah keterangan mengenai hal tersebut diatas:

1. *Object* adalah komponen didalam sebuah program
2. *Property* adalah karakteristik yang dimiliki oleh *object*.
3. *Method* adalah aksi yang dapat dilakukan oleh *object*.
4. *Event* adalah kejadian yang dapat dialami oelh *object*.

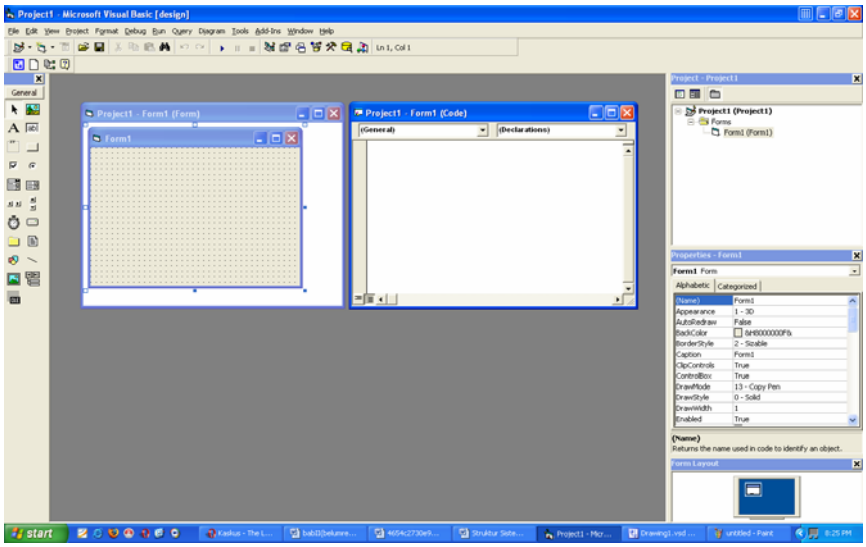
Seperti program berbasis *Windows* lainnya, *Visual Basic* terdiri dari banyak jendela (*windows*) ketika kita akan melalui *Visual Basic* sekumpulan *windows* yang saling berkaitan inilah yang disebut dengan *Integrated Development Environment* (IDE). Program yang berbasis *windows* bersifat *Event-Driven*, artinya program bekerja berdasarkan *event* yang terjadi pada *object* di dalam program tersebut, misalnya jika seorang *user* mengklik sebuah tombol maka program akan memberikan "reaksi" terhadap *event* klik tersebut. Program akan memberikan reaksi sesuia dengan kode-kode program yang dibuat untuk suatu *event* pada *object* tertentu. Pada waktu memulai *Visual Basic* beberapa *windows* kecil berada di dalam sebuah *windows* besar (*windows* induk), bentuk inilah yang dikenal dengan format *Multiple Document Interface* (MDI).

Pada gambar 2.20 memperlihatkan tampilan awal saat akan membuat sebuah **New Project** pada Visual Basic 6.0.



Gambar 2.20 Tampilan awal Visual Basic

Pada gambar 2.21 memperlihatkan contoh tampilan *Integrated Development Environment* (IDE) pada sebuah *project Visual Basic* dengan sebuah *form* dan sebuah *Command Button*.



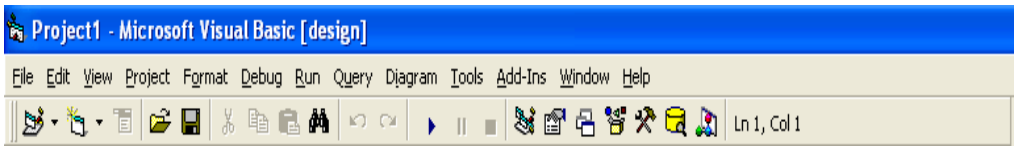
Gambar 2.21 Tampilan IDE Visual Basic

Menu pilihan pada *Visual Basic*

1. Menu Bar/Toolbar

   Menu Bar *Visual Basic* berisi semua perintah *Visual Basic* yang dapat dipilih untuk melakukan tugas tertentu, isi dari menu ini sebagaian hampir sama dengan program-program *windows* pada umumnya.

   Toolbar adalah tombol-tombol (*shortcut*) yang mewakili suatu perintah tertentu pada *Toolbar*. Ini dapat dilihat pada gambar 2.22.



Gambar 2.22 Menu Bar/*Toolbar*

2. Toolbox

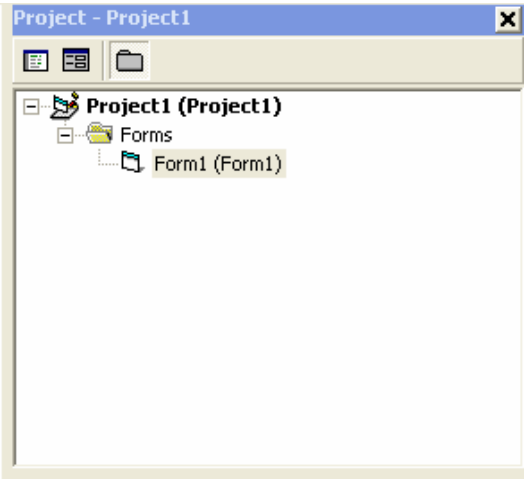
   Toolbox adalah sebuah "kotak piranti" yang mengandung semua objek atau "*control*" yang dibutuhkan untuk membentuk suatu program aplikasi. Kontrol adalah suatu objek yang akan menjadi penghubung antara program aplikasi dan *user*nya yang kesemuanya harus diletakkan di dalam jendela *form*. *Toolbox* dapat disembunyikan untuk memberikan ruangan bagi element pada *Intergrated Development Environment* (IDE) lainnya. Sehingga lebih mempermudah desain maupun penulisan program. Ini dapat dilihat pada gambar 2.23.



Gambar 2.23 *Toolbox*

3. Project Window

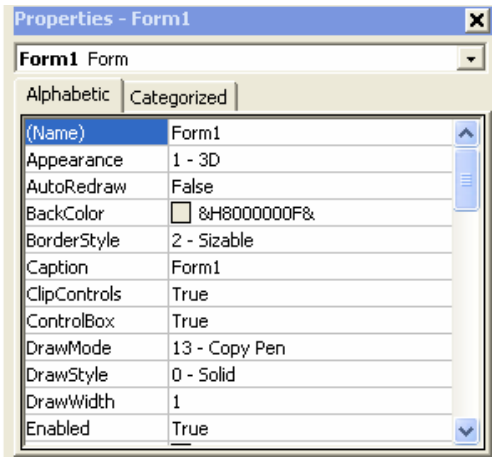
   Window ini menampilkan seluruh *form*, *class*, *class module* dan komponen lainnya yang ada pada sebuah *project*. Ini dapat dilihat pada gambar 2.24.



Gambar 2.24 *Project Window*

4. Property Window

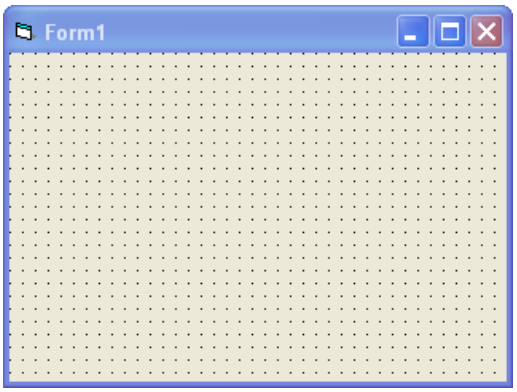
   Window ini berisi seluruh *property* dari masing-masing *object* pada sebuah *project* yang meliputi *property form* dan kontrol-kontrol yang ada pada *form* tersebut. Beberapa *property* dapat diisikan pada tahap desain dan adapula *property* yang harus diisikan dengan menuliskan kode selama program dijalankan (*runtime*). Ini dapat dilihat pada gambar 2.25.



Gambar 2.25 *Property Window*

5. Form

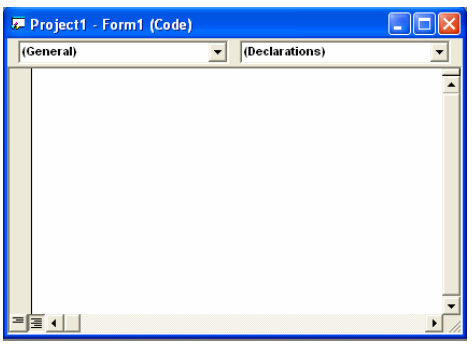
   Form adalah sebuah atau beberapa *window* untuk pembuatan program aplikasi. *Form* ini dapat memuat berbagai macam *control* (tombol-tombol maupun teks) yang diperlukan dalam desain program yang sesuai dengan kebutuhan program. Ini dapat dilihat pada gambar 2.26.



Gambar 2.26 *Form*

6. Code Window

   Pada *window* inilah semua kode/perintah tentang program dituliskan dengan memperhatikan *event* apa saja yang diperlukan. Ini dapat dilihat pada gambar 2.27.



Gambar 2.27 *Code Window*

2.16.1 Pengaksesan port serial dengan Visual Basic 6.0

Pada *port* serial komputer dengan VB 6.0 dapat diakses dengan menggunakan MSComm. *Library* untuk pengaksesan *port* serial melalui kontrol MSComm yaitu Mscomm32.ocx . Kontrol MSComm pada jendela *toolbox* didapat dari komponen kontrol Microsoft Comm Control 6.0. kontrol MSComm menyediakan fasilitas komunikasi 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 *port* serial, maka harus menggunakan MSComm sebanyak *port* serial yang dipakai.

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

- *CommPort* : digunakan untuk menentukan nomor *port* serial yang akan dipakai.
- *Setting* : digunakan untuk menset nilai *baud rate*, *pariti*, jumlah bit data, dan jumlah *bit stop*.
- *PortOpen* : digunakan untuk membuka ataupun menutup *port* serial yang dihubungkan dengan MSComm ini.
- *Input* : digunakan untuk mengambil data string yang ada pada *buffer* penerima.
- *Output* : Digunakan untuk menulis data *string* pada *buffer* kirim.

Berikut merupakan properti untuk mengirim satu karakter dengan MSComm pada Commport 1 :

```
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:

- *Port* serial yang digunakan adalah Comm 1
- *Setting* MSComm dengan *baud rate* 9600. tanpa bit paritas, jumlah data 8 bit dan jumlah *stop bit* adalah1
- Membuka *port* serial Comm 1

- Mengirim satu karakter (“A”)
- Menutup kembali com serial yang dipakai

Even pada MSComm hanya mempunyai satu even saja yaitu even OnComm saja. Even OnComm dibangkitkan jika nilai properti 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 di baca pada saat *run time.*

- comEventFrame : *Hardware* mendeteksi adanya kesalahan framing.
- comEventRxParity : *Hardware* mendeteksi adanya kesalahan parity.
- comEventRxOver : *Buffer* penerima mengalami over flow, tidak ada ruang kosong lagi pada *buffer* penerima.
- comEventTxFull : *Buffer* pengiriman telah penuh.
- comEventOverrun : *Port* mengalami *overrun*
- comEventBreak : Sinyal *Break* dikirim
- comEventDCB : Mendapatkan kembali *Device Control Clock* (DCB) dari *port* serial.

Berikut adalah nilai–nilai properti dari even komunikasi pada CommEvent yaitu:

- commEvSend : Jumlah karakter pada *Buffer* kirim lebih sedikit dari pada nilai properti Sthreshold. Even ini akan dibangkitkan jika nilai properti sthreshold tidak diisi dengan “0”.
- comEvReceive : Telah diterima karakter sebanyak nilai properti Rthreshold. Even ini akan dibangkitkan terus menerus sampai data diambil dari *Buffer* penerima menggunakan perintah *Input*. Even ini akan dibangkitkan jika nilai pada properti Rthreshold tidak diisi “0”.
- commEvCTS : Terjadi perubahan pada saluran *Clear to Send.*
- commEvDSR : Terjadi perubahan pada saluran *Data Set Ready*.
- commEvRing : Terjadi perubahan pada saluran *Carrier Detect.*
- comEvRing : Terdeteksi adanya sinyal *Ring.*
- comEvEOF : Karakter *End of File* diterima.

Berikut adalah contoh program pada penggunaan Even OnComm untuk berkomunikasi. Namun yang hanya dibaca hanya even comEvReceive saja, yang lainya diabaikan

```
Private Static Sub MsComm1_OnCom ()
```

```
Dim Buffer as variant
Select Case MSComm1.CommEvent
        Case comEvReceive
                If  MSComm1,InbufferCount = 1 then
                Buffer = CStr(MSComm1.Input)
                Text1.Text  = asc(Midbuffer, 1,1))
                End If
        End If
End Select
End Sub
```

Kode-kode program pada prosedur di atas akan melakukan aksi dengan mendeteksi even comEvReceive, kemudian menentukan apakah sudah diterima 1 karakter pada *buffer* penerima.

2.16.2 Pengaksesan secara langsung melalui register UART

Saluran yang digunakan UART untuk berkomunikasi serial yaitu TXD dan RXD serta saluran-saluran untuk kontrol, yaitu DCD, DSR, RTS, CTS, DTR, dan RI. Saluran ini ada yang berfungsi sebagai *output* dan data yang sebagai *input.* Terkecuali saluran RXD, saluran-saluran ini dapat diakses melalui register UART. Berikut adalah tabel dan lokasi bit saluran tersebut pada UART.

Tabel 2.5 Alamat dan Lokal bit pada register UART

Nama Pin	Nomor pin DB9	Com1	Com2	Bit	Arah
RXD	2	3FBH	2FBH	2	Input
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.

2.16.3 Pengaksesan port serial dengan kontrol MSComm

Kontrol MSComm menyediakan fasilitas komunikasi 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 *port* serial, maka harus menggunakan MSComm sebanyak *port* serial yang dipakai.

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

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

Berikut merupakan properti untuk mengirim satu karakter dengan MSComm pada port Comm 1 :

```
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:

- *Port* serial yang digunakan adalah Comm 1
- *Setting* MSComm dengan *baud rate* 9600. tanpa bit paritas, jumlah data 8 bit dan jumlah *stop bit* adalah1
- Membuka *port* serial Comm 1
- Mengirim satu karakter (“A”)
- Menutup kembali com serial yang dipakai