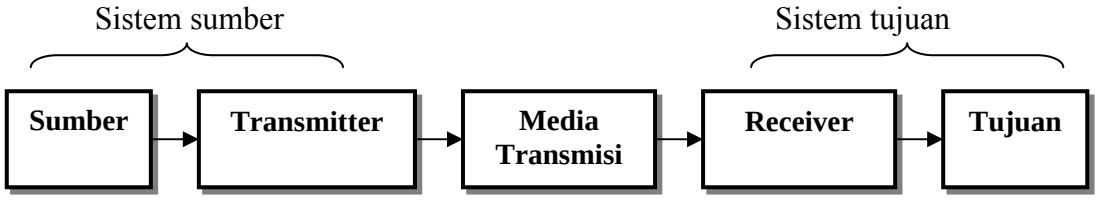
BAB II
LANDASAN TEORI

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.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).
Berdasarkan medium perambatannya, gelombang dikelompokkan menjadi 2 yaitu:

1. Gelombang Mekanik

   Merupakan gelombang yang merambat dalam medium yang bersifat elastis, seperti gelombang bunyi, gelombang pada permukaan air dan gelombang pada tali.
2. Gelombang Elektromagnetik

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

Gelombang akustik adalah gelombang yang terjadi karena adanya getaran mekanik perantara (padat, cair, dan gas). Berdasarkan frekuensi kerjanya, gelombang akustik dapat dikelompokan menjadi 3 yaitu:

1 Gelombang Infrasonik

   Merupakan gelombang akustik yang mempunyai frekuensi lebih kecil dari 20 Hz. Frekuensi gelombang infrasonik ini sangat rendah sehingga tidak terdengar oleh manusia.

2 Gelombang Sonik

   Merupakan gelombang akustik yang mempunyai daerah frekuensi antara 20 Hz sampai 20 KHz. Gelombang sonik dapat terdengar oleh manusia.

3. Gelombang Ultrasonik

   Merupakan gelombang akustik yang mempunyai frekuensi diatas 20 KHz. Frekuensi gelombang ultrasonik ini sangat tinggi sehingga tidak terdengar oleh manusia.

Dalam pengukuran untuk menentukan kualitas dan kuantitas suatu bahan dengan menggunakan gelombang ultrasonik, dikenal tiga macam metode pengukuran yaitu:

1. Metode Resonansi

   Prinsip dasar dari metode ini adalah berdasarkan terjadinya frekuensi resonansi pada suatu bahan, dimana jika sistem dalam keadaan resonansi maka akan terjadi gelombang berdiri pada bahan tersebut. Biasanya metode ini digunakan untuk mengukur ketebalan suatu bahan.

2. Metode Intensitas

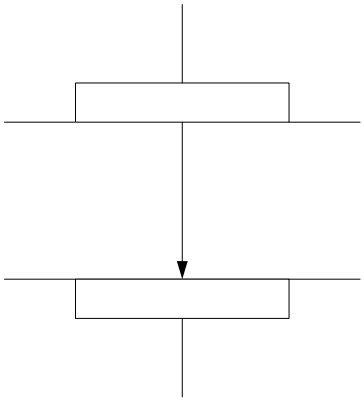
   Prinsip utama dari metode intensitas adalah mengukur besarnya intensitas gelombang ultrasonik yang diterima setelah melewati bahan. Sehingga berdasarkan pengurangan ini dapat diketahui terjadinya cacat didalam bahan.

3. Metode Waktu Tempuh

   Prinsip utama dari metode waktu tempuh adalah dengan cara mengukur waktu tempuh gelombang ultrasonik didalam bahan. Ditinjau berdasarkan cara perambatan gelombang ultrasonik yang melalui suatu bahan, dapat diklasifikasikan menjadi dua cara yaitu:

   a. Cara Langsung

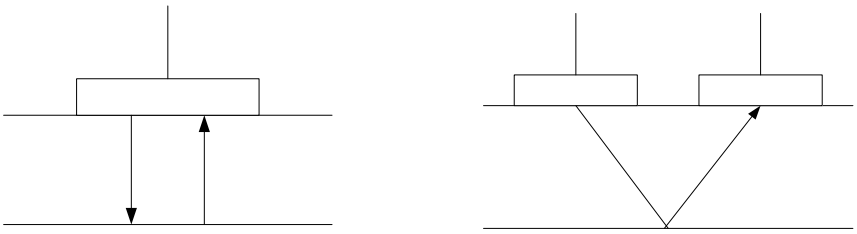
      Cara langsung merupakan cara dimana penempatan transduser pengirim dan penerima saling berhadapan seperti yang diperlihatkan dalam gambar 2.2. Cara ini lebih umum digunakan untuk menentukan kualitas dan homogenitas bahan.



Gambar 2.2. Cara transmisi langsung

b. Cara Pantulan

Merupakan cara yang memanfaatkan peristiwa pantulan gelombang pada permukaan sampel bahan yang akan diuji. Cara ini memungkinkan penggunaan satu buah transduser saja yang berfungsi sebagai pemancar dan penerima sekaligus.



Gambar 2.3. Cara transmisi tidak langsung

Metode waktu tempuh biasanya digunakan untuk mengukur suatu jarak atau kedalaman, yaitu dengan menggunakan rumus :

$$\text{Jarak} = \frac{\text{Kec. suara} \times \text{Waktu pantul}}{2} \quad \text{.....(1)}$$

2.3.Sensor Ultrasonik

Sensor ultrasonik yaitu suatu alat yang berfungsi mengukur besaran jarak dan kecepatan dan sensor ini tidak langsung dapat masuk ke mikrokontroller karena perlu penyesuaian besaran tegangan dan lain-lainnya maka dikondisikan dulu sinyalnya dibagian pengkondisi sinyal (*signal conditioner*), sehingga levelnya sesuai atau dapat dimengerti oleh bagian input *mikrokontrolle*r atau *prosessor* lainnya.

2.3.1.Karakteristik fisik gelombang Ultrasonik

Gelombang ultrasonik adalah gelombang yang timbul akibat getaran mekanik dengan frekuensi diatas batas ambang pendengaran manusia yakni diatas 20 Khz. Jika gelombang bolak-balik terjadi terus menerus secara periodik maka akan menghasilkan deretan gelombang periodik dimana pada setiap gerak periodik, partikel-partikel yang

berada pada titik-titik yang sama pada gelombang tersebut akan berada dalam fase yang sama.

Jarak antara dua nilai puncak gelombang yang berurutan (gelombang *transversal*) atau jarak dari dua bagian pemampatan gelombang yang berurutan (gelombang *longitudinal*) disebut panjang gelombang (λ). Waktu yang dibutuhkan untuk menempuh satu gelombang penuh atau waktu yang ditempuh sepanjang gelombang tersebut disebut periode (T). hubungan antara panjang gelombang dengan periode ini adalah :

$$\lambda = c \cdot T \quad \text{.....(2)}$$

freukuensi gelombang (f) adalah banyaknya getaran yang terjadi persatuan waktu periode sehingga persamaan 2 dapat juga ditulis dalam bentuk frekuensi terhadap laju gelombang bunyi dalam medium (c) yakni :

$$\lambda = c / f \quad \text{.....(3)}$$

Besarnya panjang gelombang ini sangat penting untuk menentukan batas resolusi pencitraan sistem. Dua bentuk struktur yang berdekatan panjang gelombangnya tidak bisa diidentifikasikan secara terpisah pada pencitraan ultrasonik.

Gelombang ultrasonik berasal dari suatu tranduser ultrasonik. Tranduser ultrasonik ini biasanya terbuat dari *piezoelektrik* yang akan mengubah energi listrik menjadi energi akustik. Energi akustik ini kemudian diradiasikan ke dalam medium didepannya. Pola radiasi dari berkas gelombangnya tergantung pada ukuran tranduser dan panjang gelombangnya. Bahan *piezoelektrik* yang digunakan untuk membuat tranduser ultrasonik dan pola radiasi yang dihasilkan masing-masing akan dijelaskan di bawah ini:

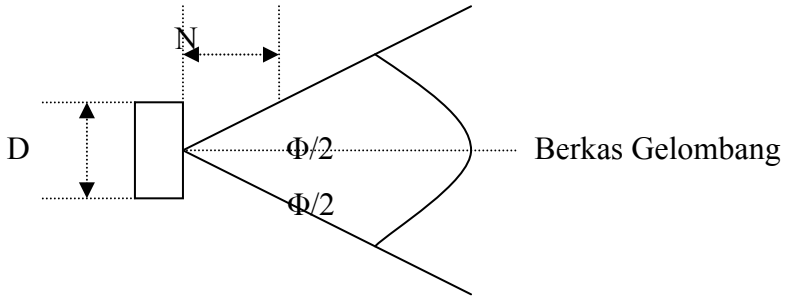
2.3.2.Bahan Piezoelektrik

Bahan *Piezoelektrik* adalah suatu bahan yang bila mendapat tekanan (*piezo*), maka pada kedua permukaannya akan timbul muatan listrik (*elektrik*). Oleh karena bahan tersebut juga merupakan kapasitor dengan konstanta dielektrik tertentu, maka pada kedua permukaanya akan timbul perbedaan tegangan listrik. Peristiwa ini disebut

efek *piezoelektrik* langsung. Sebaliknya bila kedua permukaan bahan *piezoelektrik* diberi tegangan listrik, maka akan timbul tekanan pada kedua permukaannya. Oleh karena bahan *piezoelektrik* juga merupakan bahan elastik dengan modus elastisitas tertentu, maka tebal bahan ini akan berubah. Peristiwa ini disebut efek *piezoelektrik* balik. Bila tegangan listrik yang diberikan merupakan tegangan *sinusoidal* dengan frekuensi tertentu, maka kedua permukaannya akan bergetar dengan frekuensi yang sama. Kedua efek diatas menyebabkan tranduser ultrasonik dapat dipakai baik sebagai pemancar (*transmitter*) maupun sebagai penerima (*reciver*). Bahan-bahan alam yang bersifat *piezoelektrik* adalah *kuarsa*, *Garam Rochelle*, dan *Tourmaline*. Sedangkan bahan-bahan buatan manusia adalah *Barium Titanate*, *Lead Circonate-titanate*, dan *Lead Metaniobate*.

2.3.3.Pola Radiasi Gelombang Ultrasonik

Pola radiasi yang dipancarkan oleh suatu tranduser ultrasonik ke medium yang berada didepannya tergantung pada diameter tranduser dan panjang gelombangnya, sehingga tranduser yang sama dapat mempunyai pola radiasi yang berbeda bila memancarkan gelombang ultrasonik ke medium yang berlainan. Pola radiasi suatu tranduser ultrasonik merupakan gabungan antara gelombang bidang datar (bergerak hanya ke satu arah) dan gelombang bola (bergerak ke segala arah) seperti ditunjukkan pada gambar 2.4.



Gambar 2.4. Pola Radiasi Gelombang Ultrasonik

Terlihat disini bahwa dari permukaan tranduser sampai jarak tertentu, yang disebut medan dekat (*near zone, Fresnel zone*), gelombang ultrasonik yang dipancar merupakan gelombang bidang datar. Oleh karena luas berkas gelombangnya tidak berubah, maka sepanjang medan dekat intensitasnya tidak berubah. Sifat di dalam medan dekat ini, yang bergerak lurus dengan intensitasnya tidak konstan, dapat

digunakan untuk menentukan letak cacat di dalam bahan. Panjang medan dekat ini dapat dihitung dari :

$$N = \frac{D^2}{4\lambda} \qquad \text{.....(4)}$$

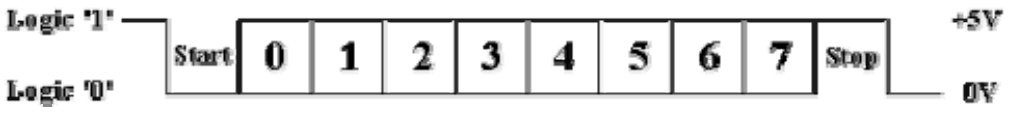
Diluar medan dekat yang disebut medan jauh (*far zone, Fraunhofer zone*), gelombang ultrasonik akan menyebar seperti gelombang bola sehingga intensitasnya berkurang, yaitu berbanding terbalik dengan kuadrat jarak. Sifat di dalam medan jauh ini, menyebar dengan penurunan intensitas yang teratur, dapat dimanfaatkan untuk menentukan ukuran cacat di dalam bahan. Besarnya sudut penyebarannya dapat dihitung dari :

$$Sin = \frac{\phi}{2} = 1{,}22\frac{\lambda}{D} \qquad \text{.....(5)}$$

Bila diameternya besar, maka medan dekatnya panjang dan sudut penyebarannya kecil. Demikian juga jika panjang gelombangnya pendek atau frekuensinya tinggi akan menghasilkan berkas gelombang yang lurus dan panjang. Berkas gelombang seperti ini berguna sekali untuk mendeteksi cacat yang terletak jauh di dalam bahan. Sebaliknya jika diameternya kecil dan frekuensinya rendah, maka berkasgelombangnya menyebar dengan medan dekat yang pendek. Berkas gelombang seperti ini berguna sekali untuk menentukan ukuran cacat yang terletak di dekat permukaan. Pemilihan besarnya frekuensi masih ditentukan oleh absorbsi bahan.

2.4.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.



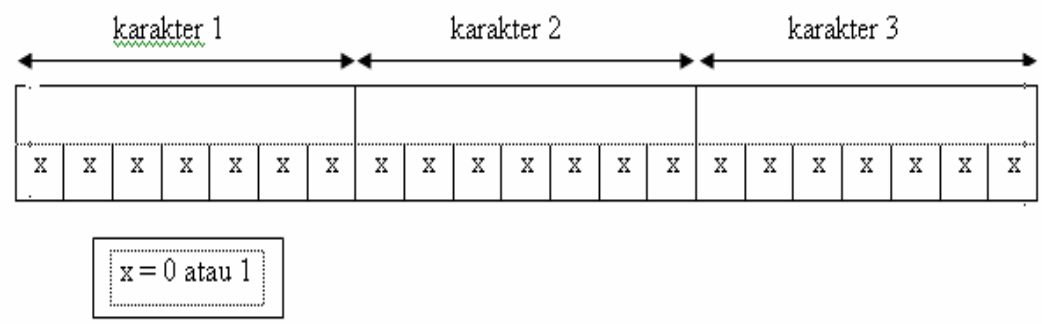
Gambar 2.5. *Aliran data tak sinkron*

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.

2.5.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.

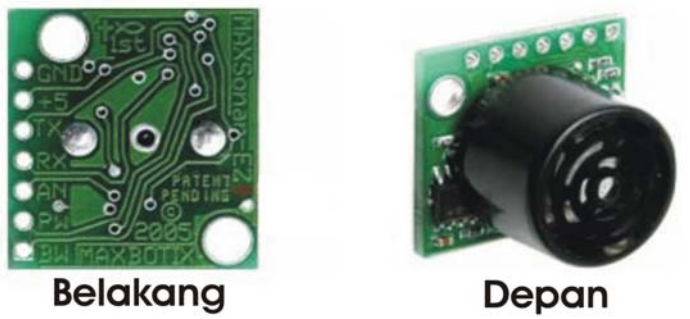
2.6.Perbedaan pengiriman sinkron dan tak sinkron

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 dapat bekerja pada laju yang lebih tinggi dibandingkan pengiriman tak sinkron. 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.

2.7.LV-MaxSonar EZ-1

LV-MaxSonar EZ-1 merupakan produk *Maxbotix* yang dapat mendeteksi obyek dalam jarak 0 inci – 254 inci (6,45 meter diudara). *LV-MaxSonar EZ-1* ini dapat mengukur jarak obyek tersebut dengan ketelitian jarak minimum deteksi 1 inci untuk jarak 6 inci – 254 inci, artinya jarak minimum yang dideteksi oleh *LV-MaxSonar EZ-1* adalah 6 inci, sehingga obyek yang berjarak 0 – 6 inci akan dianggap berjarak 6 inci. Kelebihan sensor ini adalah tersedia berbagai jenis keluaran, antara lain : *pulse width,* tegangan analog, dan UART (*Universal Asynchronous Receiver-Transmitter*). Berikut adalah Gambar 2.7. untuk sensor ultrasonik *LV-MaxSonar EZ-1*.



Gambar 2.7. *Sensor Ultrasonik LV-MaxSonar EZ-1*

Konfigurasi pin LV-MaxSonar EZ-1 adalah sebagai berikut :

1. GND : Saluran ground.
2. +5 : Vcc.
3. TX : Merupakan saluran output dengan keluaran berupa serial (UART) dengan format ASCII. Output diawali dengan kapital 'R', diikuti 3 buah karakter ASCII yang menyertakan jarak dalam inci dan diakhiri dengan ASCI 13. dengan *baudrate* 9600, 8 bit, *no parity* dan jumlah bit *stop* 1.
4. RX : Digunakan untuk menerima data.
5. AN : Merupakan saluran output, dimana keluarannya berupa tegangan analog dengan skala (Vcc/512) per inci.
6. PW : Merupakan saluran output, dimana keluarannya berupa pulsa dengan skala 147 μ S per inci.
7. BW : Berfungsi sebagai pengontrol pengiriman data serial pada pin TX. Jika pin BW berkondisi high maka pin TX akan mengirimkan data serial.

2.8.Komunikasi Data Serial

2.8.1.Metode arah komunikasi

Dalam suatu sistem komunikasi dikenal 3 jenis arah komunikasi yaitu:

1. Arah komunikasi Simplex (satu arah)

   Yaitu suatu sistem yang berkomunikasi dalam satu arah saja yang dimana hanya memiliki salah satu bagian saja, contohnya Informasi Radio, TV dan lain-lain.
2. Arah komunikasi Half duplex (dua Arah)

   Yaitu suatu sistem yang berkomunikasi dalam dua arah tetapi dilakukan secara bergantian sehingga salah satu perangkat dapat berfungsi sebagai penerima atau pemancar saja, contohnya Radio *walky talky*, Radio CB, dan lain-lain.
3. Arah komunikasi Full duplex (dua Arah)

   Yaitu suatu sistem yang berkomunikasi dalam dua arah secara langsung atau bersamaan, sebagai contoh Telepon, Handphone, dan lain-lain.

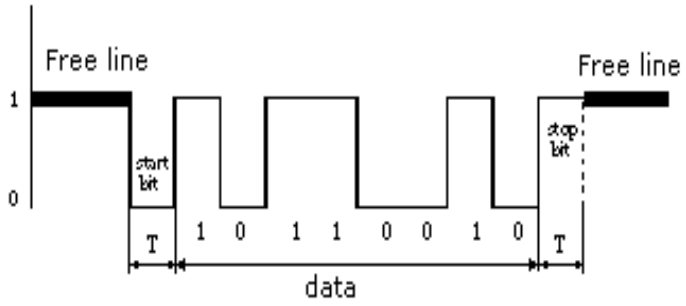
2.8.2.Tata Cara Komunikasi Data Serial

Dikenal dua cara komunikasi data secara serial, yaitu komunikasi data serial secara sinkron dan komunikasi data serial secara asinkron. Pada komunikasi data serial sinkron, clock dikirimkan bersama-sama dengan data serial, sedangkan komunikasi data

serial secara asinkron, clock tidak dikirimkan bersama data serial, tetapi dibangkitkan secara sendiri-sendiri baik pada sisi pengirim (*transmitter*) maupun pada sisi penerima (*receiver*). Pada IBM PC (*personal computer*) kompatibel port serialnya termasuk jenis asinkron. Komunikasi data serial ini dikerjakan oleh UART (*Universal Asynchronous Receiver/Transmitter*). IC UART dibuat khusus untuk mengubah data paralel menjadi data serial dan menerima data serial menjadi data paralel.

Pada UART, kecepatan pengiriman data (*baudrate*) dan fase clock pada sisi *transmitter* dan pada sisi *receiver* harus sinkron. Untuk itu diperlukan sinkronisasi antara *transmitter* dan *receiver*. Hal ini dilakukan oleh bit *start* dan bit *stop*. Ketika saluran tansmisi dalam keadaan *idle*, output UART adalah dalam keadaan logika '1'. Ketika *transmitter* ingin mengirimkan data, output UART akan diset dulu kelogika '0' untuk waktu 1 bit. Sinyal ini pada *receiver* akan dikenali sebagai sinyal *start* yang digunakan untuk mensinkronkan fase clocknya sehingga sinkron dengan fase clock *transmitter*, selanjutnya data akan dikirimkan secara serial dari bit paling rendah (bit 0) sampai bit tertinggi. Selanjutnya akan dikirim sinyal *stop* sebagai akhir dari pengiriman data serial.

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.

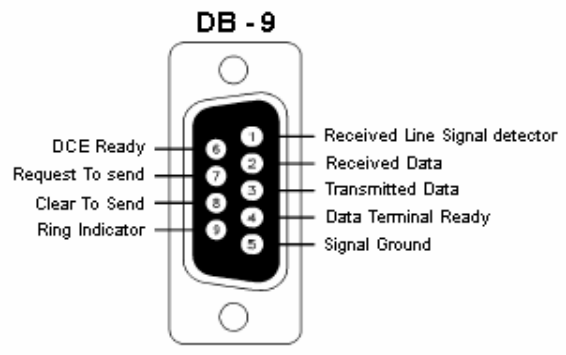


Gambar 2.8. Pengiriman data serial

Kecepatan transmisi (*baudrate*) dapat dipilih bebas dalam rentang tertentu. *Baudrate* yang umum dipakai adalah 110, 135, 150, 300, 600, 1200, 2400 dan 9600 (bit/detik). Dalam komunikasi data serial, *boudrate* dari kedua alat yang berhubungan harus diatur pada kecepatan yang sama, selanjutnya harus ditentukan panjang data (6, 7 atau 8 bit), paritas (genap, ganjil atau tanpa paritas), dan jumlah bit stop (1, 1.5 atau 2 bit).

2.8.3.Konfigurasi Port Serial

Gambar 2.9 merupakan konektor port serial DB-9 pada bagian belakang komputer. Pada komputer *IBM PC* kompatibel biasanya akan ditemukan dua konektor port serial DB-9 yang biasa dinamai COM1 dan COM2, terlihat pada Gambar 2.9 Port serial DB-9 terdapat 9 pin yang mempunyai fungsi yang berbeda, konfigurasi pin DB-9 dapat dilihat pada Tabel 2.1



Gambar 2.9. Konektor serial DB-9 pada bagian belakang CPU

Pada PC terdapat 2 macam konektor RS232 yaitu jenis 25 pin dan jenis 9 pin. Adapun sinyal dari pin-pin tersebut berisikan data yang dapat diperhatikan pada table berikut ini:

Tabel 2.1.. Konfigurasi pin dan nama sinyal konektor serial DB-9

Nomor Pin	Nama Sinyal	Direction	Keterangan
1	DCD	In	Data Carrier Detect / Received Line Singnal Detect
2	RxD	In	Receive Data
3	TxD	Out	Transmit Data
4	DTR	Out	Data Terminal Ready
5	GND	-	Ground
6	DSR	In	Data Set Ready
7	RST	Out	Request to Send
8	CST	In	Clear to Send
9	RI	In	Ring Indiator

Keterangan mengenai fungsi saluran RS232 pada konektor DB-9 adalah sebagai berikut:

1. *Received line signal detect*, dengan saluran ini DCE memberitahukan ke DTE bahwa terminal masukkan ada data masuk.
2. *Receive data*, digunakan DTE untuk menerima data dari DCE
3. *Transmit data*, digunakan DTE untuk mengirimkan data dari DCE.
4. *Data terminal ready*, pada saluran ini DTE memberitahukan kesiapan terminalnya.
5. *Signal ground*, sebagai saluran ground.
6. *DCE ready*, sinyal aktif pada saluran ini menunjukan bahwa DCE sudah siap.
7. *Reques to send*, dengan saluran ini DCE diminta mengirim data oleh DTE.
8. *Clear to send*, dengan saluran ini DCE memberitahukan bahwa DTE boleh mulai mengirimkan data.
9. *Ring indication*, pada saluran ini DCE memberitahu ke DTE bahwa sebuah station menghendaki hubungan dengannya.

Untuk dapat menggunakan port serial harus mengetahui alamatnya dulu. *Base Address* COM1 terdapat pada alamat 1016 (3F8h) dan COM2 terdapat pada alamat 760 (2F8h). Alamat tersebut adalah alamat yang secara umum digunakan, tergantung dari komputer yang digunakan. Tepatnya bisa dilihat pada peta memori tempat menyimpan alamat tersebut, yaitu memori 0000.0400h untuk *base address* COM1 dan memori 0000.0402 untuk *base address* COM2.

Setelah diketahui *base address* nya, maka dapat ditentukan alamat register-register yang akan digunakan untuk komunikasi port serial ini, register-register yang digunakan tersebut dapat dilihat pada Tabel 2.2.

Tabel 2.2. Nama register yang digunakan beserta alamatnya

Nama Register	COM 1	COM 2
TX Buffer	3F8h	2F8h
RX Buffer	3F8h	2F8h
Baud rate Devisor Latch LSB	3F8h	2F8h
Baud rate Devisor Latch MSB	3F9h	2F9h
Interupt Enable Register	3F9h	2F9h

Interupt Identifikation Register	3FAh	2Fah
Line Control Register	3FBh	2FBh
Modem Control Register	3FCh	2FCh
Line Status Register	3FDh	2FDh
Modem Status Register	3FEh	2Feh

Keterangan mengenai fungsi register-register tersebut adalah sebagai berikut :

1. *TX Buffer*, digunakan untuk menampung dan menyimpan data yang akan dikirim ke port serial.
2. *RX Buffer*, digunakan untuk menampung dan menyimpan data dari DCE.
3. *Boud Rate Divisor Latch LSB*, digunakan untuk menampung byte bobot rendah untuk pembagi *clock* pada *IC UART* agar didapat *baud rate* yang tepat.
4. *Boud Rate Divisor Latch MSB*, digunakan untuk menampung byte bobot rendah untuk pembagi *clock* pada *IC UART* sehingga total angka pembagi adalah 4 byte yang dapat dipilih dari 0001h sampai FFFFh. Berikut ini adalah tabel angka pembagi yang sering digunakan.

Tabel 2.3. Angka Pembagi Clock pada IC UART

Baud Rate (bit/detik)	Angka Pembagi
300	0180H
600	0C00H
1200	0060H
1800	0040H
2400	0030H
4800	0018H
9600	000CH

Sebagai catatan, *Register Boud Rate Divisor Latch* ini bisa diisi jika bit 7 pada *Register Line Control Register* diisi 1.

5. *Interrupt Enable Register*, digunakan untuk menset interupsi apa saja yang akan dilayani komputer. Berikut ini adalah tabel rincian bit pada *Interrupt Enable Register*.

Tabel 2.4. Rincian bit pada Interrupt Enable Register

Nomor Bit	Keterangan
0	1 : Interrupsi akan diaktifkan jika meneima data
1	1 : Interrupsi akan diaktifkan jika register Tx kosong
2	1 : Interrupsi diaktifkan jika ada perubahan keadaan pada Line Status Register
3	1 : Interrupsi diaktifkan jika ada perubahan keadaan pada Status Register
4, 5, 6, 7	Diisi 0

6. *Interrupt Identification Register*, digunakan untuk menentukan urutan prioritas interrupsi. Berikut adalah tabel rincian bit pada *Interrupt Identification Register.*

Tabel 2.5. Rincian bit pada Interrupt Identification Register

Nomor Bit	Keterangan
0	0 : Interrupsi menunggu 1 : *No Interrupt* pending
1 dan 2	00 : Prioritas tertinggi oleh *Line Status Register* 01 : Prioritas tertinggi oleh Register Rx jika menerima data 10 : Prioritas tertinggi oleh Register Tx jika telah kosong 11 : Prioritas tertinggi oleh Modem Status Register
3, 4, 5, 6, 7	Diisi 0

7. *Line Control Register*, digunakan untuk menentukan jumlah bit data, jumlah bit *parity*, jumlah bit *stop*, serta untuk menentukan apakah *baud rate divisor* dapat diubah atau tidak. Berikut ini adalah tabel rincian bit pada *Line Control Register*.

Tabel 2.6. Rincian bit pada Line Control Register

Nomor Bit	Keterangan
0 dan 1	Jumlah bit data 00 : jumlah bit data adalah 5 01 : jumlah bit data adalah 6 10 : jumlah bit data adalah 7 11 : jumlah bit data adalah 8
2	Bit stop 0 : jumlah bit stop adalah 1 1 : jumlah bit stop adalah 1,5 untuk bit data dan 2 untuk 6 hingga 8 bit data
3	Bit paritas 0 : tanpa paritas 1 : dengan paritas
4	0 : paritas ganjil 1 : paritas genap
5	1 : bit paritas ikut dikirimkan (*stick parity*)
6	0 : *set break control* tidak diaktifkan 1 : *set break control* diaktifkan
7	0 : *baud rate divisor* tidak dapat diakses 1 : *baud rate divisor* dapat diakses

8. *Modem Control Register*, digunakan untuk mengatur saluran pengatur modem terutama saluran DTR dan saluran RST. Berikut ini tabel rincian bit pada *Modem Control Register*.

Tabel 2.7. Rincian bit pada Modem Kontrol Register

Nomor bit	Keterangan
0	Bit DTR 0 : saluran DTR diaktifkan (aktif 0) 1 : saluran DTR dibuat normal (tidak aktif)
1	Bit RST 0 : saluran RST diaktifkan (aktif 0)

	1 : saluran RST dibuat normal (tidak aktif)
2	Bit OUT1, digunakan untuk penghubung ke perangkat lain, dapat dibuat logika *high* atau logika *low*. Secara normal tidak digunakan
3	Bit OUT2, digunakan untuk penghubung ke perangkat lain, dapat dibuat logika high atau logika low
4	0 : *Loop back* internal diaktifkan 1 : *Loop back* internal tidak diaktifkan
5, 6, 7,	Diisi 0

9. *Line Status Register*, digunakan untuk menampung bit-bit yang menyatakan keadaan penerimaan atau pengiriman data dan status kesalahan operasi. Berikut adalah tabel rincian bit pada *Line Status Register*.

Tabel 2.8. Rincian bit pada Line Status Register

Nomor bit	Keterangan
0	1 : menyatakan adanya data masuk pada *buffer* Rx
1	1 : data yang masuk mengalami *overrun*
2	1 : terjadi kesalahan pada bit *parity*
3	1 : terjadi kesalahan *framing*
4	1 : terjadi *break Interrupt*
5	1 : menyatakan bahwa register Tx telah kosong
6	1 : menyatakan bahwa *Transmitter Shift Register*
7	Diisi 0

10. *Modem Status Register*, digunakan untuk menampung bit-bit yang menyatakan status dari saluran hubungan dengan modem. Berikut ini tabel rincian bit pada *Modem Status Register*.

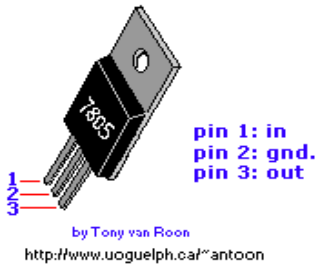
Tabel 2.9. Rincian bit pada Modem Status Register

Nomor bit	Keterangan
0	1 : menyatakan adanya perubahan keadaan di saluran *Clear to Send* (CTS)

1	1 : menyatakan adanya perubahan keadaan di saluran *Data Set Ready* (DSR)
2	1 : menyatakan adanya perubahan keadaan di saluran *Ring Indicator* (RI) dari *low* ke *high*
3	1 : menyatakan adanya perubahan di saluran *Receive Line Signal Detect* (DCD)
4	1 : menyatakan saluran *Clear to Send* (CST) sudah dalam keadaan aktif
5	1 : menyatakan saluran *Data Set Ready* (DSR) sudah dalam keadaan aktif
6	1 : menyatakan bahwa saluran *Ring Indocator* (RI) sudah dalam keadaan aktif
7	1 : menyatakan bahwa saluran *Receive Line Signal Detect* (DCD) sudah dalam keadaan aktif

2.9.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 2.10



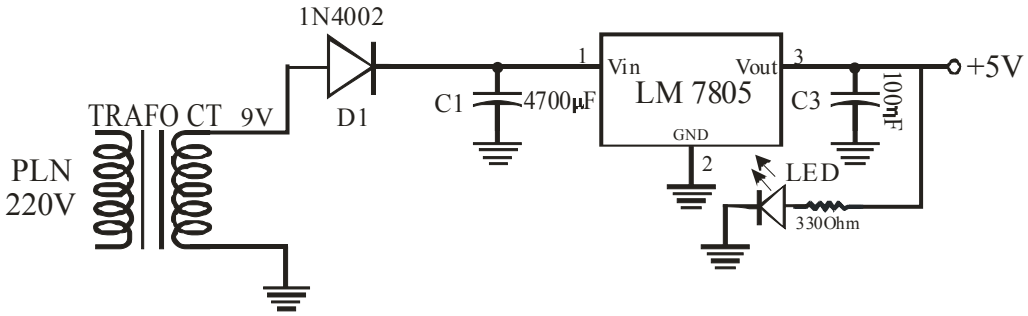
Gambar 2.10. *IC LM7805*

2.10.Catu Daya

Tegangan keluaran dari transformator yang masih dalam bentuk tegangan bolak-balik (AC) akan disearahkan oleh penyearah setengah gelombang. Sebelum tegangan diregulasi, dilewatkan dulu pada tapis atau filter untuk memperkecil tegangan kerut

(ripple). Komponen penapis digunakan kapasitor polar. Konsep dasar kerja kapasitor polar ini untuk menyimpan muatan-muatan listrik (Q), sehingga jika diberi tegangan kapasitor tersebut akan terisi namun tidak seketika penuh muatan, sebaliknya ketika kapasitor membuang muatan dengan diberikan beban (R) tidak seketika muatan kosong, tapi meluruh terhadap waktu dari besarnya kapasitansi dan resistor akan terpakai sesuai besar beban yang diberikan. Konstanta waktu pengisian sangatlah penting untuk pemangkas *rippel*. Besarnya muatan kapasitor menentukan agar rangkaian tetap stabil.

Untuk meregulasi tegangan digunakan regulasi positif yaitu regulator tegangan 5 Volt (LM7805). Dari hasil pengukuran tegangan keluaran setelah terlegulasi adalah 5,02 Volt.



Gambar 2.11. Rangkaian catu daya

2.11.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.

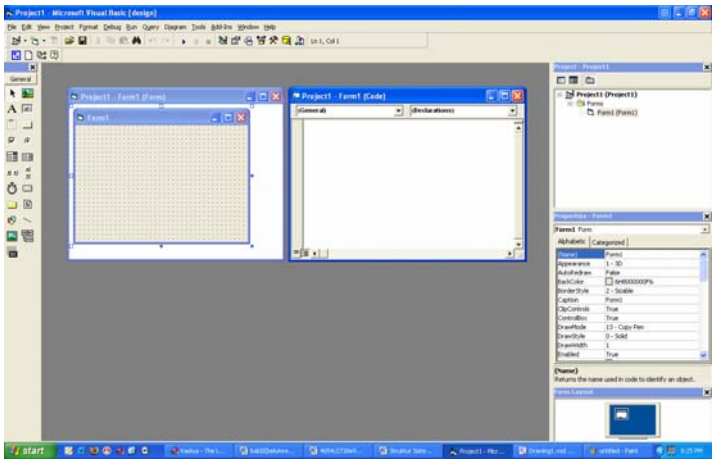
Bahasa pemrograman adalah bahasa yang dimengerti oleh *object* untuk melakukan tugas-tugas tertentu, salah satu contoh bahasa *Visual Basic*. Bahasa pemrograman *Visual Basic* yang dikembangkan oleh *Microsoft* sejak tahun 1991 merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (*Baginners 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 2.12 memperlihatkan contoh tampilan *Integrated Development Environment* (IDE) pada sebuah *project Visual Basic* dengan sebuah *form* dan sebuah *Command Button*.



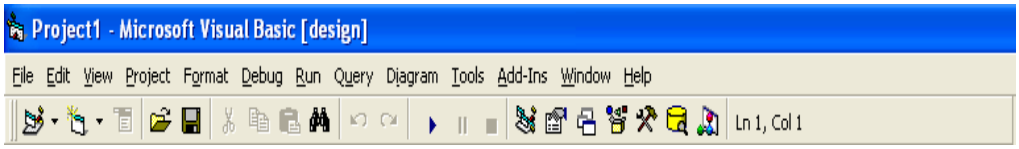
Gambar 2.12. Tampilan IDE Visual Basic

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 2.13.



Gambar 2.13. Menu Bar/*Toolbar*

2. Toolbox

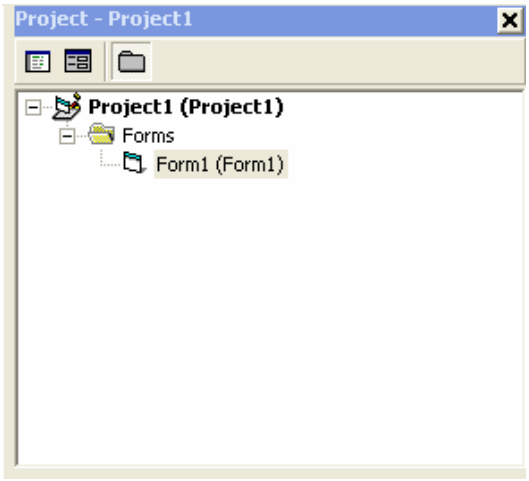
Toolbox adalah sebuah "kotak piranti" yang mengandung semua objek atau "*control*" yang dibutuhkan untuk membentuk suatu program aplikasi. Kontrol adalah suatu objek yang akan menjadi penghubung antara program aplikasi dan *user*nya yang kesemuanya harus diletakkan di dalam jendela *form*. *Toolbox* 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 2.14.



Gambar 2.14. *Toolbox*

3. Project Window

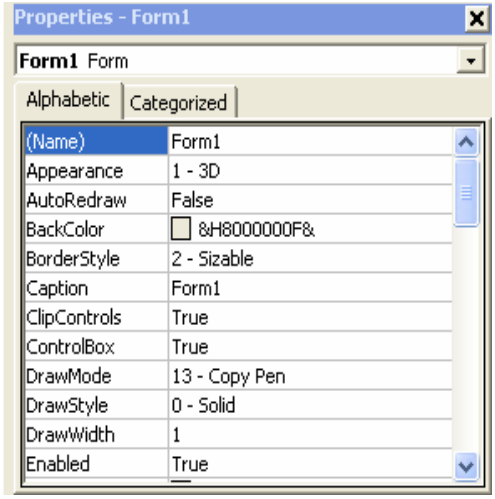
Window ini menampilkan seluruh *form*, *class*, *class module* dan komponen lainnya yang ada pada sebuah *project*. Ini dapat dilihat pada Gambar 2.15.



Gambar 2.15. *Project Window*

4. Property Window

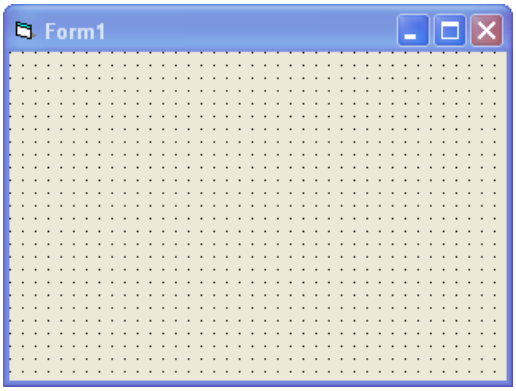
Window ini berisi seluruh *property* dari masing-masing *object* pada sebuah *project* yang meliputi *property form* dan kontrol-kontrol yang ada pada *form* tersebut. Beberapa *property* dapat diisikan pada tahap desain dan adapula *property* yang harus diisikan dengan menuliskan kode selama program dijalankan (*runtime*). Ini dapat dilihat pada Gambar 2.16.



Gambar 2.16. *Property Window*

5. Form

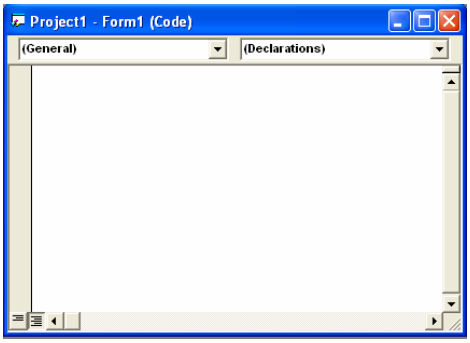
   Form adalah sebuah atau beberapa *window* untuk pembuatan program aplikasi. *Form* ini dapat memuat berbagai macam *control* (tombol-tombol maupun teks) yang diperlukan dalam desain program yang sesuai dengan kebutuhan program. Ini dapat dilihat pada Gambar 2.17.



Gambar 2.17. *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 2.18.



Gambar 2.18. *Code Window*

2.11.1.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 :

1. *CommPort* : digunakan untuk menentukan nomor *port* serial yang akan dipakai.
2. *Setting* : digunakan untuk menset nilai *baud rate*, *pariti*, jumlah bit data, dan jumlah *bit stop*.
3. *PortOpen* : digunakan untuk membuka ataupun menutup *port* serial yang dihubungkan dengan MSComm ini.
4. *Input* : digunakan untuk mengambil data string yang ada pada *buffer* penerima.
5. *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:

a. *Port* serial yang digunakan adalah Comm 1
b. *Setting* MSComm dengan *baud rate* 9600. tanpa bit paritas, jumlah data 8 bit dan jumlah *stop bit* adalah1
c. Membuka *port* serial Comm 1
d. Mengirim satu karakter (“A”)
e. 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.*

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

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

1. 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".
2. 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".
3. commEvCTS : Terjadi perubahan pada saluran *Clear to Send.*
4. commEvDSR : Terjadi perubahan pada saluran *Data Set Ready*.
5. commEvRing : Terjadi perubahan pada saluran *Carrier Detect.*
6. comEvRing : Terdeteksi adanya sinyal *Ring.*
7. 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.

2.11.2.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 2.10. 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.