BAB II
LANDASAN TEORI

2.1. Komunikasi Data

Pada dasarnya komunikasi data merupakan proses pengiriman informasi diantara dua titik melewati media transmisi. Hal itu bisa antara komputer dengan komputer, komputer dengan terminal, komputer dengan peralatan, atau peralatan dengan peralatan. Kegunaan dasar dari sistem komunikasi data adalah untuk menjalankan pertukaran data antara dua pihak.

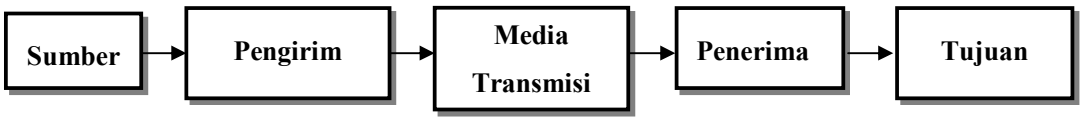
2.1.1. Tujuan Komunikasi Data

Tujuan dari komunikasi data adalah :

1. Memungkinkan pengiriman data dalam jumlah besar secara efisien dari suatu tempat ke tempat yang lain.
2. Memungkinkan penggunaan sistem komputer dan peralatan pendukung dari jarak jauh (*remote computer use*)
3. Memungkinkan penggunaan komputer secara terpusat maupun secara tersebar sehingga mendukung manajemen dalam hal kontrol, baik desentralisasi maupun sentralisasi.
4. Mempermudah kemungkinan pengelolaan dan pengaturan data yang ada dalam berbagai macam sistem komputer.
5. Mempercepat penyebarluasan informasi.

2.1.2. Sistem Komunikasi Data

Gambar 2.1 adalah contoh sistem komunikasi yang sederhana. [1]



Gambar 2.1. Sistem komunikasi sederhana

Pada diagram model komunikasi data sederhana dapat dijelaskan :

1. Sumber

Alat ini membangkitkan data sehingga dapat ditransmisikan, contohnya telepon dan PC.

2. Pengirim

Biasanya data yang dibangkitkan dari sistem sumber tidak ditransmisikan secara langsung dalam bentuk aslinya (data analog/digital) tetapi diubah menjadi bentuk sinyal. Sebuah transmitter cukup memindahkan dan menandai informasi dengan cara yang sama seperti menghasilkan sinyal-sinyal elektromagnetik yang dapat ditransmisikan melewati beberapa media transmisi.

3. Media Transmisi

Berupa jalur transmisi tunggal atau jaringan kompleks yang menghubungkan antara sumber dengan tujuan.

4. Penerima

Pada bagian ini sinyal dari pengirim diterima dari sistem transmisi dan memindahkan bentuk sinyal elekromagnetik menjadi sinyal informasi yang dapat ditangkap oleh tujuan.

5. Tujuan

Menerima data yang dihasilkan oleh penerima.

2.1.3. Media Transmisi

Transmisi data terjadi antara *transmitter* dan *receiver* melalui beberapa media transmisi sebagai jalur fisik. Media transmisi untuk gelombang elektromagnetik dibedakan menjadi dua yaitu terbimbing (*guided)* dan tidak terbimbing *(unguided*). Pada media *guided*, terdapat jalur fisik yang membatasi sinyal yang merambat. Contoh *guided media* adalah twisted pair tembaga, coaxial cable tembaga, serta serat optik. Sedangkan media *unguided* merupakan media dengan menggunakan pasangan antena di pengirim dan penerima. Contoh *unguided media* adalah udara, *vacuum*, dan air. [2]

2.2. Metode Arah Komunikasi

Dalam komunikasi data dikenal 3 jenis arah komunikasi yaitu :

a. *Simplex* (satu arah)

Yaitu sistem komunikasi satu arah. Contoh TV, radio.

b. *Half duplex* (dua Arah)

Yaitu komunkasi dua arah tapi bergantian dalam satu kanal radio yang sama. Contoh walky talky.

c. ***Full duplex*** **(dua Arah)**

Yaitu komunikasi dua arah tanpa bergantian sehingga dapat berkomunikasi dalam saat yang sama antara pengirim dan penerima. Contoh sistem telepon selular (telepon, handphone). Secara lengkap dapat dilihat pada gambar 2.2.



Gambar 2.2. Jenis Komunikasi Data

2.3. Gangguan Transmisi

Dalam sistem komunikasi, data yang diterima oleh penerima kemungkinan berbeda dengan data yang ditransmisikan oleh pengirim karena adanya gangguan transmisi. Gangguan yang ada pada transmisi data yaitu :

1. Terjadi sinyal yang diterima berbeda dengan sinyal yang dikirimkan.
2. Digital - bit errors, yaitu bit satu diubah menjadi biner nol dan seterusnya.
3. Gangguan terhadap amplitudo sinyal:
   a. Atenuasi

      Kekuatan sinyal berkurang ketika jarak transmisi data bertambah jauh pada media transmisi.

   b. Distorsi tunda

      Sinyal yang diterima terdistorsi akibat perbedaan waktu tunda antar komponen frekuensi.

   c. Derau

      Adalah sinyal-sinyal tidak diinginkan yang muncul antara pengirim dan penerima yang mengubah sinyal transmisi.

2.4. Jenis Modulasi

Setelah mengetahui sistem dasar komunikasi data, bagian lain yang perlu diketahui dari komunikasi data adalah jenis modulasi yang akan digunakan dari media transmisi. Dalam hal ini jenis modulasi dibagi menjadi 2 yaitu:

2.4.1. Modulasi Analog

Modulasi analog yaitu suatu komunikasi dimana sinyal informasi yang akan dikirimkan langsung memodulasi sinyal pembawa, tanpa melakukan proses digitalisasi terlebih dahulu. Adapun jenis-jenis dari modulasi analog yaitu :

a. Amplitude Modulation (AM)

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

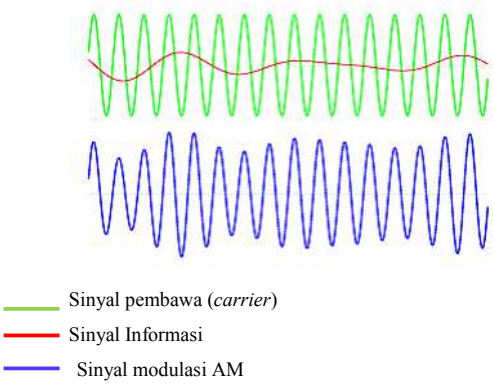
$s(t) = [1+n_a\, m(t)] \cos 2\pi f_c t$ [2] …… (2.1)

n_a = indeks modulasi

m(t) = sinyal informasi

$\cos 2\pi f_c t$ = sinyal pembawa

Bentuk gelombang dari *Amplitude Modulation* dapat dilihat pada gambar 2.3.

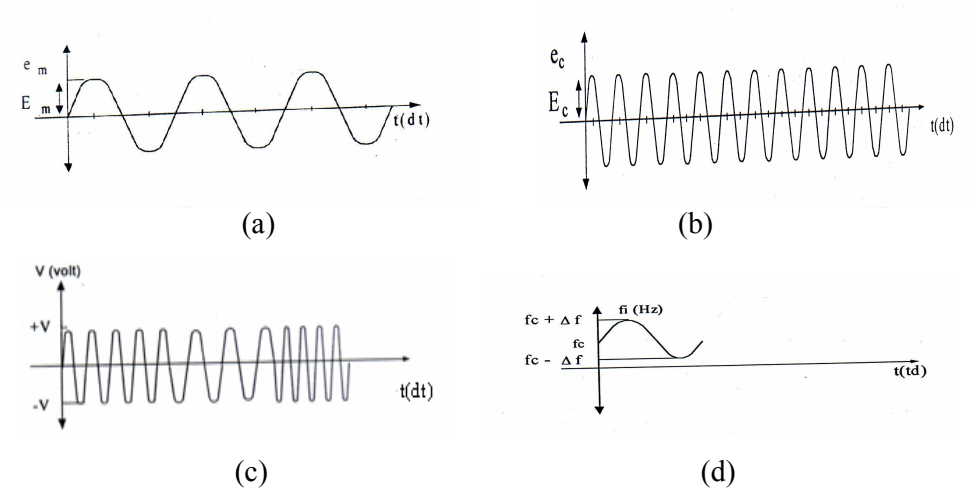


Gambar 2.3. Bentuk gelombang AM

Sehingga AM merupakan metode pertama kali yang digunakan untuk menyiarkan radio komersil. Namun AM memiliki suatu kelemahan yang mendasar yaitu rentan terhadap gangguan atmosfir, dan juga memiliki bandwith yang sempit sehingga membatasi kualitas suara yang dipancarkan [4].

***b. Frequency Modulation* (FM)**

Modulasi Frekuensi merupakan suatu proses modulasi dengan cara mengubah frekuensi gelombang pembawa sinusoidal, yaitu dengan cara memodulasikan sinyal informasi pada frekuensi pembawa / *carrier* tersebut.



(a) (b)

(c) (d)

Gambar 2.4. Gelombang-gelombang FM

Sinyal informasi pada gambar 2.4(a) dimodulasikan pada sinyal pembawa gambar 2.4(b), dengan cara mengubah lengkungan frekuensi sesaat fungsi waktu gambar 2.4(c) sehingga menghasilkan gelombang pembawa yang termodulasi, seperti pada gambar 2.4(d). Pada modulasi frekuensi ini sinyal modulasi (e_m) digunakan untuk mengubah frekuensi pembawa. [4]

Jika pengubahan frekuensi pembawa adalah $k.e_m$ dan k adalah konstanta deviasi frekuensi, maka frekuensi pembawa sesaat (f_i) sebagai berikut :

$f_i = f_c + k.e_m$ ..... (2.2)

k = 0,1,2,3,…..

f_c = frekuensi pembawa tanpa modulasi

Jika e_m suatu gelombang sinus, maka :

$e_m = E_{m\ maks}\ Sin\ \omega_m t$ ..... (2.3)

Frekuensi pembawa sesaat menjadi :

$f_i = f_c + k.E_{m\ maks}\ Sin\ \omega_m t$ ..... (2.4)

Deviasi frekuensi puncak dari sinyal ($\triangle f$) didefinisikan sebagai berikut :

$\triangle f = k.Em\ maks$ ..... (2.5)

sehingga persamaan (2.4) menjadi :

$f_i = f_c + \triangle f\ Sin\ \omega_m t$ ..... (2.6)

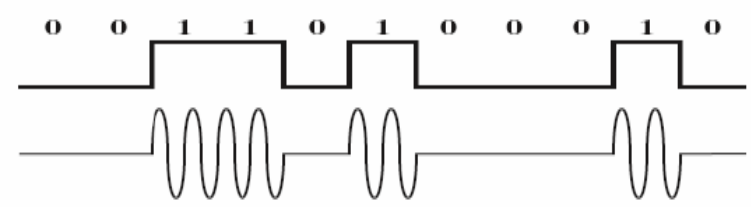
2.4.2. Modulasi Digital

Modulasi digital adalah suatu bentuk modulasi dengan data yang akan dikirimkan berupa data digital yang hanya memiliki logika 1 dan logika 0. Jika data digital tersebut akan dikirimkan melalui media transmisi, diperlukan suatu teknik modulasi agar data yang akan dikirimkan dapat diterima dengan baik. Adapun jenis dari teknik modulasi digital yaitu :

***a. Amplitude Shift Keying* (ASK)**

ASK merupakan suatu teknik modulasi yang memiliki frekuensi dan fasa dalam keadaan tetap, tetapi memiliki amplitude yang berbeda. Pada ASK, dua nilai biner (0 dan 1) dilambangkan sebagai dua buah amplitudo berbeda dari frekuensi sinyal pembawa. Umumnya, salah satu dari amplitudo adalah nol yaitu satu digit biner yang ditunjukkan melalui keberadaan sinyal pada amplitudo yang konstan dari suatu sinyal pembawa, sedangkan yang lain melalui ketidakadaan sinyal pembawa. Gambar 2.5 adalah contoh sinyal ASK. Sinyal s(t) ASK tersebut dirumuskan :

$$s(t) = \begin{matrix} A\cos(2\pi f_c t); biner 1 \\ A\cos(2\pi f_c t); biner 0 \end{matrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2.7)$$



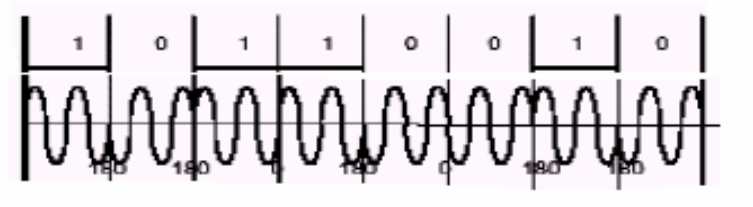
Gambar 2.5. Amplitude Shift Keying

***b. Phase Shift Keying* (PSK)**

Pada PSK, skema paling simpel adalah *Binary* PSK (BPSK) dengan menggunakan 2 fasa berbeda untuk merepresentasikan 0 dan 1. Gambar 2.6 adalah contoh sinyal BPSK.

Sinyal BPSK dapat dirumuskan sebagai berikut :

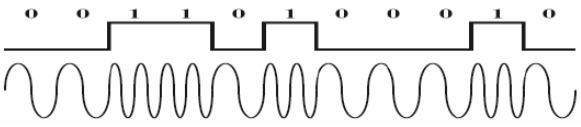
$$\text{s(t)} = \begin{matrix} A\cos(2\pi fct) = A\cos(2\pi fct); biner 1 \\ A\cos(2\pi fct + \pi) = -A\cos(2\pi fct); biner 0 \end{matrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2.8)$$



Gambar 2.6. BiPhase Shift Keying

***c. Frequency Shift Keying* (FSK)**

FSK (Frequency Shift Keying) adalah pengiriman sinyal melalui pergeseran frekuensi. Pada FSK, dua nilai biner yang ditunjukkan oleh dua frekuensi yang berbeda didekat frekuensi pembawa. Dimana f_1 dan f_2 penyeimbang khusus dari frekuensi pembawa fc oleh persamaan namun dengan jumlah yang berlawanan. Gambar 2.7 adalah contoh sinyal FSK. Sinyal FSK dapat dirumuskan :

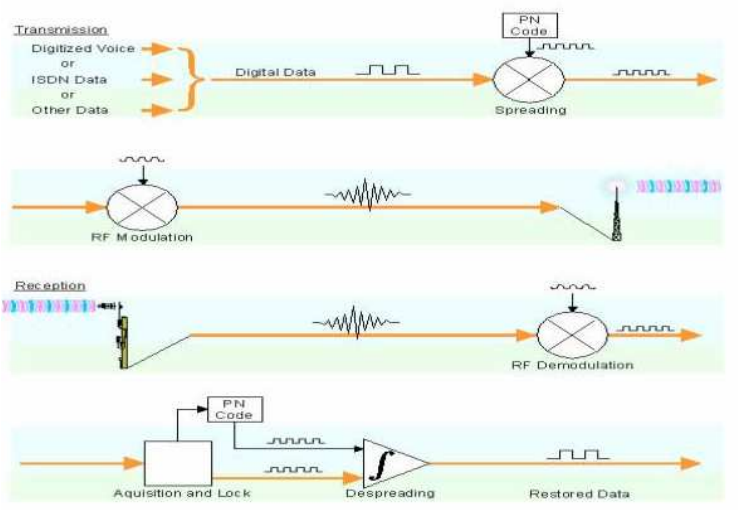


$$S(t) = \begin{matrix} A\cos f_1 t; biner 0 \\ A\cos f_2 t; biner 1 \end{matrix} \quad \text{.....} \quad (2.9)$$

Gambar 2.7. Frequency Shift Keying

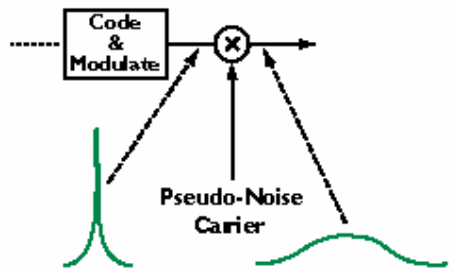
2.5. Direct Sequence Spread Spectrum (DSSS)

Teknik ini digunakan untuk mengirimkan data digital dengan sinyal analog. Tiap bit dalam sinyal asli diwakili oleh banyak bit dalam sinyal yang ditransmisikan. Untuk melindungi sinyal, kode yang digunakan adalah pseudorandom. Penerima dapat merekonstruksi kode untuk deteksi sinkron. Kode pseudorandom ini juga disebut dengan pseudonoise (PN).[8]



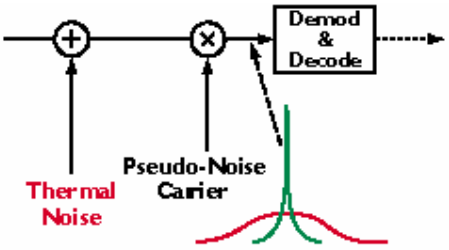
Gambar 2.8. Sistem Direct Sequence Spread Spectrum

Pada bagian pemancar dari sistem Direct sequence terjadi proses perkalian antara gelombang komunikasi dengan sequence biner pseudonoise (PN) ±1.



Gambar 2.9. Proses DSSS di pemancar

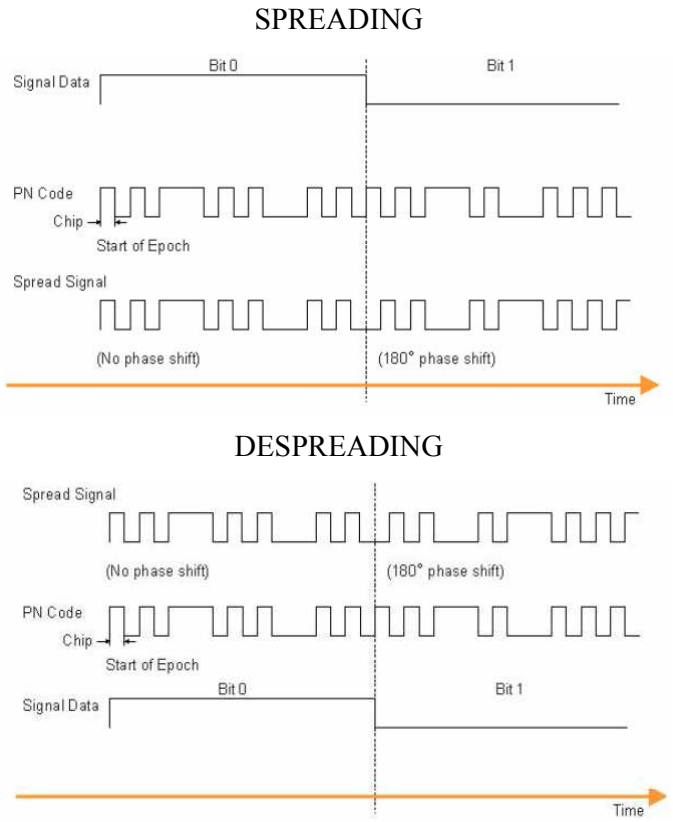
Proses spreading dilakukan setelah proses modulasi, semuanya dalam bentuk biner. Sinyal yang ditransmisikan dalam keadaan dibatasi bandwidthnya (bandlimited). Perkalian oleh replika dari sequence ±1 yang sama di penerima akan mengembalikan sinyal ke bentuk asalnya.



Gambar 2.10. Proses DSSS di penerima

Noise yang tidak mempunyai hubungan dengan sequence PN menjadi sinyal yang mirip noise (*noise-like*) dan bandwidthnya meningkat ketika mencapai detektor.

Gambar 2.10 menunjukkan sebuah contoh proses *spreading* dan *dispreading*.



Gambar 2.11. Proses spreading dan despreading

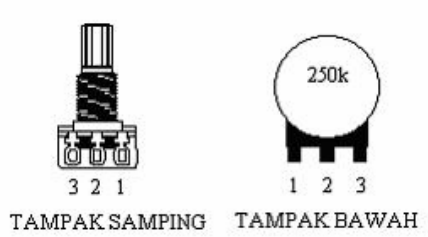
2.6. Telemetri

Telemetri berasal dari kata "Tele" yang berarti jauh dan "Metri" yang berarti ukuran, dengan demikian telemetri adalah suatu sistem komunikasi untuk transfer data pengukuran jarak jauh yang menggunakan medium transmisi tertentu. Biasanya media transmisi yang digunakan adalah kabel hingga yang bersifat wireless menggunakan satelit.[9]

Sistem telemetri biasanya banyak digunakan untuk parameter pengukuran fisis maupun kimiawi di daerah-daerah yang sukar untuk dijangkau manusia seperti gunung, gua, lembah, eksplorasi antariksa dan pada berbagai kegiatan proses industri/pabrik. Sehingga untuk mengukur aktivitas alamiah disana, misalkan aktivitas gunung berapi, temperatur yang sangat tinggi pada tangki reaktor, petugas cukup meletakkan alat ukur dan sistem telemetri di lokasi, dengan demikian tidak perlu membawa peralatan pengolah data.

2.7. Trimmer Potensiometer (Trimpot)

Potensiometer adalah resistor variabel, yang dapat dirubah resistansinya. Pada potensiometer terdapat 3 kaki seperti ditunjukkan pada gambar 2.12.



Gambar 2.12. Trimmer Potensiometer

Kaki 1 = Input

Kaki 2 = Output

Kaki 3 = Ground

Nilai resistansi variabel diperoleh diantara kaki tengah dengan salah satu kaki pada ujung kiri/kanan.

2.8. Multiplexing

Multiplexing atau mux adalah suatu alat yang memungkinkan beberapa sinyal komunikasi menggunakan sebuah channel transmisi bersama-sama. Dua bentuk yang paling umum dari multiplexing adalah [1]:

2.8.1. Frequency Division Multiplexing (FDM)

FDM bisa dipergunakan bersama-sama dengan sinyal-sinyal analog. Sejumlah sinyal secara simultan dibawa menuju media yang sama, dengan cara mengalokasikan band frekuensi yang berlainan ke masing-masing sinyal. Diperlukan peralatan modulasi untuk memindahkan setiap sinyal ke band frekuensi yang diperlukan, sedangkan peralatan multiplexing diperlukan untuk mengkombinasikan sinyal-sinyal yang dimodulasikan.

2.8.2. Synchronous Time Division Multiplexing (TDM Synchronous)

TDM synchronous bisa dipergunakan bersama-sama dengan sinyal digital atau sinyal-sinyal analog yang membawa data digital. Pada bentuk multiplexing ini, data dari berbagai sumber dibawa dalam frame secara berulang-ulang. Setiap frame terdiri dari susunan jatah waktu, dan setiap sumber ditetapkan bahwa setiap framenya terdiri dari satu atau lebih jatah waktu.

2.9. ADC 0808

Salah satu komponen penting dalam sistem akuisisi data adalah pengubah besaran analog ke digital atau disebut juga ADC (*Analog to Digital Converter*). Pengubah ini akan mengubah besaran-besaran analog menjadi bilangan-bilangan *digital* sehingga dapat diproses oleh PC. Perubahan-perubahan satuan fisis bisa dengan cepat ditanggapi oleh komputer. Hal yang penting dalam suatu rangkaian ADC adalah resolusi, yaitu besaran analog terkecil yang masih dapat dikonversi menjadi satuan digital, yaitu :

$$\text{Resolusi (r)} = \frac{1}{2^n}.Vref \quad \text{.....} \quad (2.10)$$

dimana n adalah banyaknya bit ADC dan Vref adalah tegangan referensi yang digunakan.

Data digital yang dihasilkan ADC hanyalah merupakan pendekatan proporsional terhadap masukan analog. Hal ini karena tidak mungkin melakukan konversi secara sempurna berkaitan dengan kenyataan bahwa informasi digital berubah dalam step-step, sedangkan analog berubahnya secara kontinu. [6]

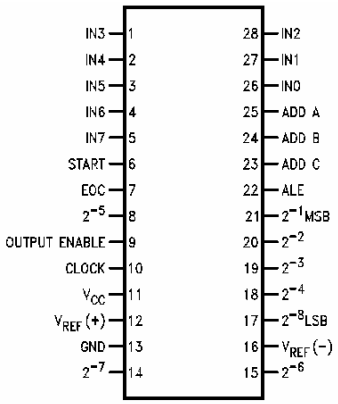
ADC membutuhkan sinyal detak atau *clock* untuk operasionalnya. Untuk kemudahannya, sinyal *clock* diambil dari kaki 3 output dari LM 555 disambungkan dengan kaki 10 dari ADC.

ADC 0808 merupakan konverter A/D 8 bit yang dapat mengkonversi data input sebanyak 8 kanal, sehingga dapat menerima 8 buah tranducer yang berbeda untuk setiap chipnya. Jika pada ADC 0808 ini diberikan tegangan referensi + 5V, maka resolusi adalah 20 mv.

2.9.1. Deskripsi fungsi pin ADC0808

1. Pin 1 – 5, 26 – 28 sebagai *channel* input *analog* (8 *channel*).
2. Pin 23 – 25 sebagai *multiplexer* internal (*MUX Address Line*) yang berfungsi untuk memilih input *analog* mana yang akan diaktifkan atau akan diubah menjadi data *digital*.
3. Pin 21,20,19,18,8,15,14,17 sebagai *digital* output (8 bit).
4. Pin 6 (**SC** atau *Start Conversion*). Diberikan untuk menandakan awal konversi, Pin ini berfungsi sebagai penanda input *analog* sudah siap untuk dikonversi menjadi *digital* atau belum. Jika pin ini diberi nilai 1 maka konversi sudah bisa dimulai, sebaliknya jika diberi nilai 0 maka menunggu sampai diberi nilai 1.

5. Pin 7 (***EOC*** atau *End of Conversion*). Merupakan penanda akhir konversi. Pin ini digunakan untuk menandakan apakah input yang diberikan sudah selesai dikonversi atau belum. Jika pin ini diberi nilai 1, maka data hasil konvesi sudah dapat dikirim ke mikrokontroller untuk diproses lebih lanjut.
6. Pin 9 (***OE*** atau *Output Enable*). Pin ini digunakan sebagai kendali apakah data yang telah selesai dikonversi dapat dikirim ke mikrokontroler atau tidak.
7. Pin 10 (***CLK***). Berfungsi sebagai jalur pengaturan Clock ADC.
8. Pin 11 (***VCC***) merupakan input catu daya untuk ADC0808 dan diberi tegangan sebesar 5 volt.
9. Pin 12 ($V_{Ref(+)}$). Merupakan tegangan referensi untuk memberikan batas atas konversi
10. Pin 13 (***GND***) adalah pin *ground* tegangan catu daya untuk ADC0808 dan diberi input tegangan 0 volt.
11. Pin 16 ($V_{Ref(-)}$). Merupakan tegangan referensi untuk memberikan batas bawah konversi
12. Pin 22 (***ALE*** atau *Address Latch Embedded*) digunakan untuk mengontrol *multiplexer* internal ADC0808. Juga berfungsi sebagai pembangkit pulsa ADC



Gambar 2.13. Konfigurasi Pin ADC 0808

2.9.2. Rancangan Pengubah Analog ke Digital

Sebenarnya rangkaian pengubah analog ke digital dapat dibuat dengan memakai komponen-komponen lepasan, akan tetapi ini akan memakan tempat dan linearitasnya tidak cukup baik. Karena itu dipilih pengubah dalam bentuk IC (*Integrated Circuit*) yang sudah ada dipasaran. Dari berbagai buku banyak

ditemukan bahasan tentang komponen dengan tipe ADC0808. Komponen ini memakai metode pendekatan berurutan (*successive approximation convertion)* dan hanya memerlukan sedikit komponen luar.

Adapun langkah-langkah pengkonversian pada ADC 0808 ini adalah sebagai berikut :

a. Siapkan sinyal analog di kanal yang dipilih (IN0...IN7), kemudian beri logika alamat sesuai dengan nomor kanal A2, A1, dan A0 digunakan untuk memilih alamat atau kanal 0 sampai dengan 7.
b. Beri sinyal ALE (*Address Latch Enable*) yaitu perubahan logika 0 ke 1.
c. Pulsa 1 di kaki START memulai pengkonversian.
d. Pengkonversian selesai bila pada kaki EOC (*End of Convertion*) ada perubahan dari logika 0 ke logika 1.
e. Keluaran (8 bit) dapat dibaca dengan cara memberi logika 1 di kaki OE (*Output Enable*).

Untuk memilih input yang digunakan, dapat melakukan pengdekoderan pada pin C,B, dan A. Untuk lebih jelasnya dapat melihat tabel 2.1 di bawah ini :

Tabel 2.1. Pemilihan input pada ADC 0808

Input Analog Channel	**Multriplexer Address Line**		
	C	**B**	**A**
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

2.10. Pengiriman Data Serial

Di dalam sistem komputer, karakter-karakter disajikan dalam bentuk data yang terdiri dari sederetan angka biner atau bit (*binary digit*). Setiap bit hanya bernilai biner ’1’ atau biner ’0’. Pemindahan, penyimpanan, dan pengolahan data di dalam komputer, atau mikroprosesor dapat dikerjakan berdasar atas operasi 8 bit, 16 bit atau 32 bit, tergantung jenis komputer yang digunakan. Setiap 8 bit disebut 1 *byte*. Diluar komputer atau mikroprosesor, data dapat dikirimkan ke terminal atau modem menggunakan cara pengiriman seri atau paralel.

Pengiriman seri biasanya digunakan untuk sambungan dengan jarak relatif lebih jauh. Data paralel internal dimasukkan ke pengubah paralel ke seri. Pengubah paralel ke seri biasanya dengan IC dan juga melakukan sejumlah fungsi yang lain dan dikenal sebagai UART, VART, ACIA, PIA dan lain-lain. Kanal seri mengirimkan setiap karakter per elemen sehingga hanya diperlakukan dua penghantar, yaitu kirim data (TXD) dan terima data (RXD). Bit-bit dikirimkan secara berurutan dan tidak serempak, kecepatan pemindahan data lebih rendah dibanding pengiriman secara paralel. Pengiriman akan dimulai dari LSB (*least significant bit*), dan diakhiri dengan MSB (*most significant bit*). Setiap karakter yang dikirimkan, disajikan dengan suatu urutan bit tertentu sesuai dengan sandi yang digunakan. Penerima harus mencacah isyarat data yang sama, pada waktu yang tepat sebelum membentuk kembali karakter yang diterima.

Pengiriman seri menimbulkan tiga masalah penyesuaian yaitu penyesuaian bit, penyesuaian karakter, dan penyesuaian blok. Ketika data seri akan dikirimkan, agar diterima dengan benar data seri tersebut, selang waktu yang digunakan oleh pengirim dan penerima harus sama satu terhadap yang lain. Untuk itu, pengirim dan penerima harus menambahkan detak. Istilah detak (*clock*) digunakan untuk menunjuk sembarang pulsa sumber pewaktuan (*timing pulse*). Detak penerima harus menunjukkan waktu yang tepat kapan isyarat harus dicacah oleh penerima untuk menentukan status logika dari setiap bit yang diterima

2.10.1. Tata Cara Komunikasi Data Serial

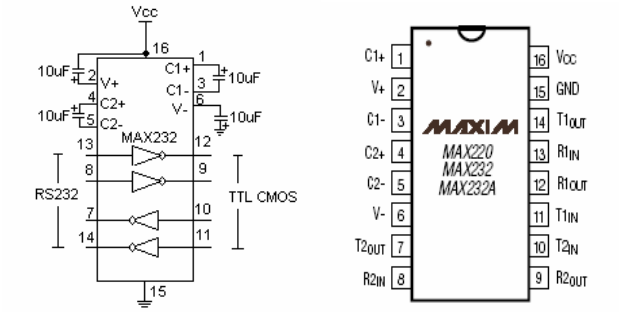
Dikenal dua cara komunikasi data secara serial, yaitu komunikasi data serial secara sinkron dan komunikasi data serial secara asinkron. Pada komunikasi data serial sinkron, *clock* dikirimkan bersama-sama dengan data serial, sedangkan komunikasi data serial secara asinkron, clock tidak dikirimkan bersama data serial, tetapi dibangkitkan secara sendiri-sendiri baik pada sisi pengirim (*transmitter*) maupun pada sisi penerima (*receive*). 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. Selain berbentuk IC mandiri, berbagai macam mikrokontroler ada yang dilengkapi UART, salah satunya adalah mikrokontroler AT89C51.

Pada UART, kecepatan pengiriman data (*baud rate*) dan *fase clock* pada sisi *transmitter* dan pada sisi *receiver* harus sinkron. Untuk itu diperlukan sinkroniasi antara *transmitter* dan *receiver*. Hal ini dilakukan oleh bit *start* dan bit *stop*. Ketika saluran transmisi dalam keadaan *idle*, output UART adalah dalam keadaan logika '1'. Ketika *transmitter* ingin mengirimkan data, output UART akan diset lebih dulu ke logika '0' untuk waktu satu bit. Sinyal ini pada *receiver* akan dikenali sebagai sinyal *start* yang digunakan untuk mensikronkan *fase clock*nya sehingga sinkron dengan *fase clock transmitter*, 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.

Kecepatan transmisi (*baud rate*) dapat dipilih bebas dalam rentang tertentu. *Baud rate* yang umum dipakai adalah 110, 135, 150, 300, 600, 1200, 2400 dan 9600 (bit/detik). Dalam komunikasi data serial, *baud rate* 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)

2.10.2. RS 232

Port serial pada PC dengan standar RS232 merupakan salah satu fasilitas yang tersedia agar sebuah PC dapat dihubungkan dengan perangkat luar. Diperlukan sebuah chip interface untuk mengkonversi level tegangan yang berbeda yaitu chip MAX232. Level tegangan yang berbeda pada tiap saluran port serial adalah ± 12 Volt DC. Pada saat level logika "1", maka tegangan yang keluar pada port serial adalah sebesar -12 Volt dan sebaliknya pada level logika "0", maka tegangan yang keluar pada port serial adalah sebesar +12 Volt. Hal ini disebut *negative logic* karena kondisi logika tertentu justru menghasilkan tegangan yang terbalik. MAX232 akan merubah level tegangan RS232 menjadi level tegangan TTL (5 volt) sekaligus mengubah logika negatif menjadi logika positif. Konfigurasi MAX232 ditunjukkan dalam gambar 2.14.

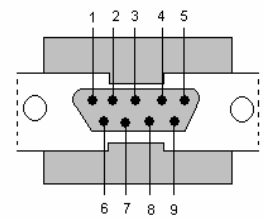


Gambar 2.14. IC dan Konfigurasi Pin MAX232

Dari gambar 2.14 diatas terlihat adanya 4 buah kapasitor yang berfungsi sebagai "pompa elektronik", keempat kapasitor inilah yang berfungsi untuk menyesuaikan level tegangan yang berbeda antara level tegangan RS232 dengan TTL.

2.10.3. DB 9

Bagian yang sangat penting pada komunikasi serial adalah DB9 dan RS232. DB9 merupakan konektor yang digunakan untuk menghubungkan perangkat keras dengan komputer. Konektor port serial DB9 pada komputer biasanya disebut dengan COM1 dan COM2. Gambar 2.15 menunjukkan konektor serial dari DB9.



Gambar 2.15. Konektor Serial DB9

Tabel 2.2. Deskripsi dan Signal dari Pin DB9

Pin	***Signal***	***Pin***	***Signal***
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Gound		

Pada tabel 2.2 diatas menunjukan sinyal-sinyal yang ada berdasarkan standard RS232. Berikut ini fungsi dari kaki pin DB9, yaitu :

1. DCD (*Data Carrier Detect*) : Sinyal ini aktif jika suatu modem mendeteksi suatu "*carrier*" dari modem lain.
2. RxD (*Received Data*) : Untuk penerimaan data serial.
3. TxD (*Transmitted Data*) : Untuk pengiriman data serial.
4. DTR (*Data Terminal Ready*) : Memberitahukan modem bahwa UART siap melakukan hubungan komunikasi.
5. GND : Sinyal *Ground*.
6. DSR (*Data Set Ready*) : Memberitahukan UART bahwa modem siap melakukan hubungan komunikasi (*link*).
7. RTS (*Request to Send*) : Sinyal untuk menginformasikan modem bahwa UART siap melakukan pertukaran data.
8. CTS (*Clear to Send*) : Untuk memberitahukan bahwa modem siap untuk melakukan pertukaran data.
9. RI (*Ring Indicator*) : Aktif jika modem mendeteksi adanya sinyal dering.

2.11. Bahasa *Assembly*

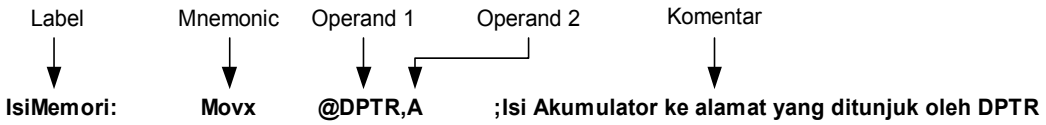
Assembler adalah program komputer yang menterjemahkan program dari bahasa *assembly* ke bahasa mesin. Sedangkan bahasa *assembly* adalah ekuivalensi bahasa mesin dalam bentuk alpanumerik. *Mnemonics* alpanumerik digunakan sebagai alat bantu bagi programer untuk memprogram mesin komputer daripada menggunakan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

2.11.1. Konstruksi Program *Assembly*

Program sumber *assembly* terdiri dari kumpulan baris-baris perintah dan biasanya disimpan dengan *extension*.ASM dengan 1 baris untuk satu perintah, setiap baris perintah tersebut bisa terdiri atas beberapa bagian, yakni bagian label, bagian *mnemonic*, bagian operand yang bisa lebih dari satu dan terakhir bagian komentar seperti yang terlihat pada gambar 2.16. Program sumber (*source code*) dibuat dengan program editor seperti Notepad atau Editor DOS, selanjutnya program sumber diterjemahkan ke bahasa mesin dengan menggunakan program *assembler*. Hasil kerja program *assembler* adalah "program objek" dan juga "*assembly listing*". Ketentuan penulisan *source code* adalah sebagai berikut:

1. Masing-masing bagian dipisahkan dengan spasi atau TAB, khusus untuk *operand* yang lebih dari satu masing-masing operand dipisahkan dengan koma.

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



Gambar 2.16. Bentuk program sumber assembly

2.11.2. Instruksi MCS-51 yang Digunakan

Beberapa instruksi yang digunakan dalam penyusunan program system telemetri adalah sebagai berikut:

1. EQU, merupakan konstanta data yang dideklarasikan dengan nilai pada variabel tertentu.
2. ORG, digunakan untuk menunjukkan lokasi memori tempat instruksi atau perintah yang ada dibawahnya disimpan.
3. MOV, merupakan instruksi ini melakukan pemindahan data dari variabel pada kode operasi kedua dan disimpan di variabel pada kode operasi pertama.
4. SETB, digunakan untuk memberikan nilai ‘1’ pada bit tertentu.
5. ACALL, melakukan lompatan dengan area sebesar 2 Kbyte.
6. CLR, digunakan untuk memberikan nilai ‘0’ pada bit tertentu.
7. INC, digunakan untuk menambahkan nilai variabel yang ditunjuk dengan ‘1’ dan hasilnya disimpan di variabel tersebut.
8. CJNE, merupakan instruksi ini melakukan perbandingan antara data tujuan dan data sumber serta melakukan lompatan ke alamat yang ditentukan jika hasil perbandingan tidak sama.
9. SJMP, digunakan untuk melakukan lompatan untuk jarak yang pendek (*Short JMP*) dengan lompatan maksimum sebesar 128 byte.
10. JNB merupakan instruksi untuk beralih ke alamat tertentu jika Bit tertentu bernilai ’0’.

11. RET, digunakan untuk melakukan lompatan ke alamat yang disimpan dalam SP dan SP-1. Instruksi ini biasa digunakan pada saat kembali dari subrutin yang dipanggil dengan instruksi ACALL atau LCALL.
12. DJNZ, digunakan untuk melakukan pengurangan pada Rn (R0…R7) dengan '1' dan lompat ke alamat yang ditentukan apabila hasilnya bukan 00. Apabila hasilnya telah mencapai 00, maka program akan terus menjalankan instruksi dibawahnya.
13. JBC, lompat jika *bit* tertentu bernilai'1', artinya jika bit yang telah ditentukan bernilai '1', maka lompat ke alamat yang telah ditentukan dan dilanjutkan ke instruksi berikutnya. C*lear* pada bit tersebut setelah lompatan dilakukan.
14. END, biasanya diletakkan di akhir baris dari *file* program sumber *assembler* sebagai tanda akhir pernyataan (*statement*) bagi program *assembler* dalam melakukan proses *assembly*.

2.11.3. Antarmuka Serial AT89C51

Pada port serial AT89C51 penerimaan dan pengiriman data port serial melalui register SBUF. Penulisan ke SBUF berarti mengisi register pengiriman ke SBUF, sedangkan pembacaan dari SBUF berarti membaca register penerimaan SBUF. Port serial pada AT89C51 bisa digunakan dalam 4 mode kerja yang berbeda, terdiri dari 1 mode bekerja secara sinkron dan 3 lainnya bekerja secara asinkron. Adapun mode kerja dari port serial, antara lain yaitu :

1. *Mode 0 :* Mode ini bekerja secara sinkron, data serial dikirim dan diterima melalui kaki P3.0 (Rxd), sedangkan kaki P3.1 (Txd) digunakan untuk menyalurkan detak pendorong data serial yang dibangkitkan oleh AT89C51. Data dikirim dan diterima 8 bit sekaligus dimulai dari bit LSB dan diakhiri dengan bit MSB. Kecepatan *baud rate* 1/12 frekuensi kristal yang digunakan
2. *Mode 1 :* Pada mode ini data dikirim melalui kaki P3.1 (Txd) dan diterima melalui kaki P3.0 (Rxd) secara asinkron (juga mode 2 dan 3). Pada mode 1 data dikirim atau diterima 10 bit sekaligus, diawali dengan 1 bit *start*, disusul dengan 8 bit data yang dimulai dari bit LSB dan diakhiri dengan 1 bit *stop*. Pada AT89C51/52 yang berfungsi sebagai penerima bit *stop* adalah RB8 dalam register SCON. Kecepatan *baud rate* bisa diatur sesuai dengan keperluan dengan menggunakan *timer*. Mode 2 dan 3 yang umum dikenal dengan UART.

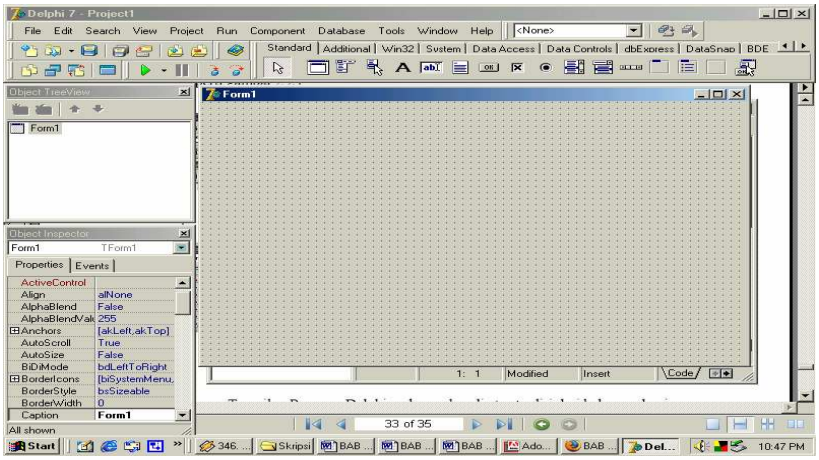
3. *Mode 2 :* Data dikirim atau diterima 11 bit sekaligus, diawali dengan 1 bit *start*, disusul 8 bit data, kemudian bit ke 9 yang bisa diatur lebih lanjut, diakhiri dengan 1 bit *stop*. Pada AT89C51/52 yang berfungsi sebagai pengirim, bit 9 tersebut berasal dari bit TB8 dalam register SCON dan yang berfungsi sebagai penerima, bit 9 ditampung pada bit RB8 dalam register SCON, sedangkan bit *stop* diabaikan tidak ditampung. *Baud rate* bisa dipilih antara 1/32 atau 1/64 frekuensi kristal yang digunakan.
4. *Mode 3 :* Mode ini sama dengan mode 2 hanya saja *baud rate*-nya bisa diatur sesuai dengan keperluan seperti mode 1.

2.12. Bahasa Pemrograman *Borland Delphi*

Bahasa pemrograman Borland Delphi bekerja dalam sistem operasi Windows. Delphi mempunyai kemampuan yang luas dan canggih. Berbagai aplikasi dapat dilakukan seperti mengolah teks, grafik, angka, *database* dan aplikasi *web*. Keunggulan Delphi terletak pada produktivitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi dan pola desain yang menarik. Selain itu Delphi juga dapat menangani data dalam berbagai format *database*, misalnya format MS-Acces, Sybase, Oracle, FoxPro dan lain-lain. Untuk memudahkan pemrograman, Delphi menyediakan fasilitas pemrograman dalam dua kelompok yaitu, objek dan bahasa pemrograman. Objek adalah suatu komponen yang mempunyai bentuk fisik dan biasanya dapat dilihat. Objek biasanya dipakai untuk melakukan tugas tertentu dan mempunyai batasan-batasan tertentu. Sedangkan bahasa pemrograman adalah kumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu untuk menjalankan tugas tertentu. Delphi menggunakan struktur bahasa pemrograman Pascal yang sudah sangat dikenal dikalangan pemrogram. Gabungan dari objek dan bahasa pemrograman berorientasi objek disebut dengan *Object Oriented Programming* (OOP).

2.12.2. IDE Delphi

Integrated Development Environment atau IDE Delphi terdiri dari : *Main Window, Toolbar, Component Pallet, Form, Code Editor* dan *Object Inspector*. Untuk memulai Delphi dari sistem Operasi Windows pilih Program, kemudian pilih Borland Delphi lalu klik Delphi maka akan terlihat tampilan utamanya seperti gambar 2.17.

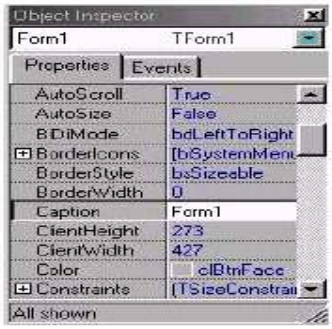


Gambar 2.17. Tampilan dasar Borland Delphi

Tampilan Program Delphi pada gambar di atas terdiri dari beberapa bagian yang tampilannya dapat diubah-ubah yaitu :

a. Windows Utama merupakan pusat pengaturan di dalam Delphi yang terletak di bagian atas. Jendela ini berisi menu, *toolbar* dan kumpulan *tab (page)* atau lembaran yang berisi *icon object* sesuai kategori yang disebut sebagai *Component Pallete*.

b. Windows Object Inspector Terletak di bagian kiri bawah. Windows ini memiliki dua buah halaman, yaitu halaman propertis dan halaman *event*. Halaman propertis digunakan untuk mengubah properti komponen. Properti dengan tanda + menunjukkan bahwa properti tersebut mempunyai subproperti. Dan *event* berfungsi untuk menangani kejadian-kejadian berupa prosedur yang dapat direspon oleh sebuah komponen.



Gambar 2.18. Tampilan Windows Object Inspector

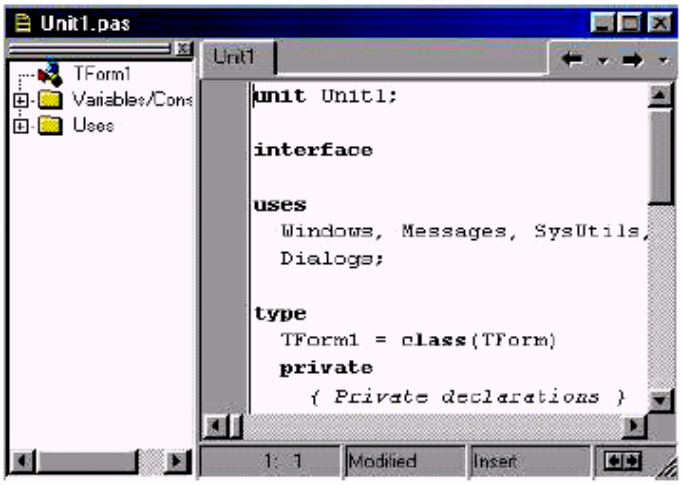
c. Windows Form

Terletak di bagian kanan *Windows Editor Program*. Untuk proses yang baru Windows ini biasanya ditampilkan dengan judul *"Form1"*. *Form* dipakai

untuk merancang Windows bagi aplikasi baru yang sedang dibuat. Sebuah aplikasi dapat berisi beberapa *form* dan minimal harus memiliki sebuah *form* yang nantinya dipakai untuk mendesain tampilan program aplikasi.

d. Code Editor

Tempat untuk menuliskan kode program. *Code Editor* dilengkapi dengan fasilitas *highlight* yang memudahkan pemakai menemukan kesalahan pada kode program. *Title bar* yang terletak pada bagian bawah terdapat nomor baris/kolom, *modified* dan *insert/overwrite*.



Gambar 2.19. Lembar Kerja Code Editor

2.12.2. Dasar Pembuatan Program

Langkah-langkah dasar yang harus ditempuh untuk membuat program aplikasi dan yang perlu dipahami oleh pemakai Delphi adalah :

1. Merancang antarmuka visual dari aplikasi dengan memilih komponen-komponen yang diinginkan dari *Component Pallet* dan menempatkannya pada *form*.
2. Dengan memakai *Windows Object Inspector* ubah nilai propertis milik *form* dan objek yang terdapat di dalamnya. Ini bertujuan untuk mendapatkan tampilan yang dikehendaki.
3. Tulis kode pada editor program untuk *event* pada objek yang diinginkan. Perlu diketahui *event* adalah mekanisme penghubung antara suatu kejadian (seperti gerakan *mouse*, penekanan tombol dan lain-lain) pada komponen dengan prosedur yang merespon (menerjemahkan dan merealisasikannya menjadi suatu tindakan kejadian tersebut). [4]