

BAB I

PENGENALAN BAHASA DELPHI

1.1 Mengenal Sepintas Borland Delphi 7

Borland Delphi 7 adalah bahasa pemrograman yang bekerja pada lingkup sistem operasi windows, dan merupakan salah satu program yang berorientasi object (OOP). kemampuannya dapat dipakai untuk merancang program aplikasi yang berpenampilan seperti program aplikasi lainya yang berbasis Windows.

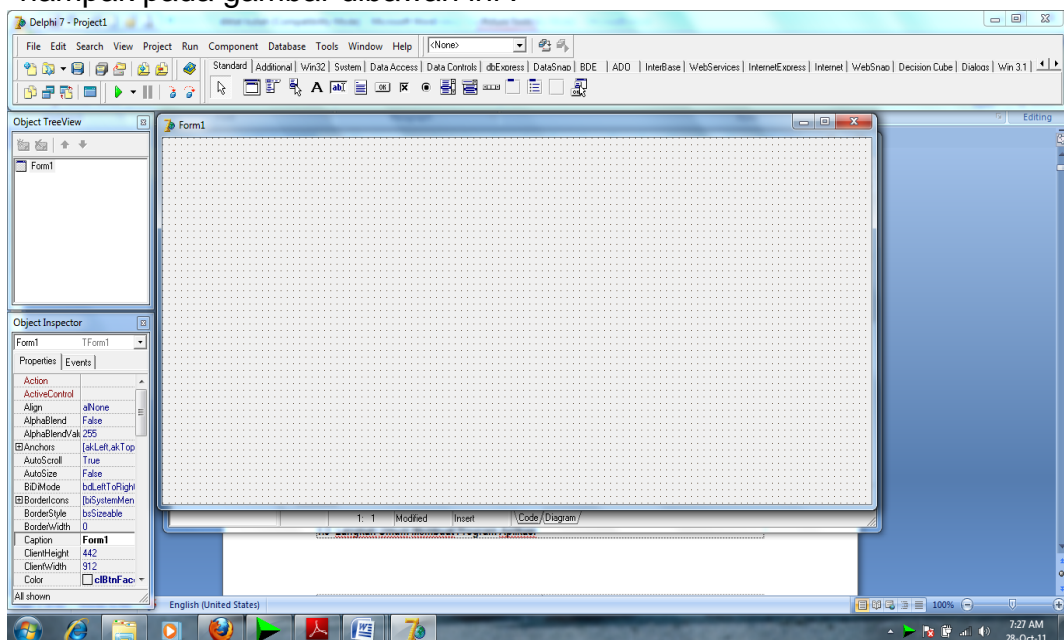
Kemampuan Borland Delphi 7 secara umum adalah menyediakan komponen komponen yang memungkinkan anda membuat program aplikasi yang sesuai dengan tampilan dan cara kerja windows, diperkuat dengan bahasa pemrograman terstruktur yang sangat handal, yaitu bahasa pemrograman Object Pascal yang sangat terkenal.

Khusus untuk pemrograman database Borland Delphi 7 menyediakan fasilitas object yang sangat kuat dan lengkap. Selain menyediakn format database Paradoxs dab dBase Borland Delphi 7 juga menangani bebrbagai macam format database seperti: MS-Access, ODBC, SyBASE, Oracle dan lain lain.

1.2 Memulai Program Borland Delphi 7

Untuk memulai menjalankan Borland Delphi 7 lakukan langkah langkah sebagai berikut :

- Klik Icon Start pada sistem operasi Windows (pastikan program Borland Delphi 7 sudah di install)
- Tunjuk program, lalu pilih Borland Delphi 7 atau Delphi 7, kemudian akan muncul program Borland Delphi 7 yang sudah siap dipakai, seperti yang nampak pada gambar dibawah ini :



1.3 Langkah Umum Membuat Program Aplikasi

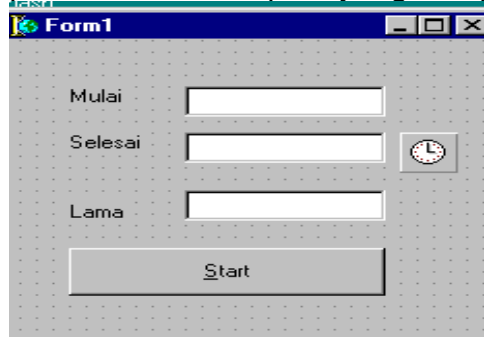
Program berikut program yang sederhana brtujuan untuk memudahkan dalam memahami membuat program aplikasi Borland Delphi 7 . Langkah langkah umum untuk membuat program aplikasi dengan Borland Delphi 7 adalah sebagai berikut :

- Gambar object dan tata letak kedalam jendela form menggunakan ikon ikon object dalam komponen palette.
- Bila perlu, tentukan properti pada tiap komponen menggunakan lembar properties pada jendela Object Inspector.

- c. Tuliskan kode program untuk event pada object yang diinginkan. Event adalah suatu kejadian yang dirasakan object, misalnya tunjuk klik, atau kejadian lainnya.

1.3.1 Membuat Program Aplikasi Pertama

Sebagai contoh yang pertama kita membuat program program aplikasi pencatat waktu seperti yang nampak pada gambar berikut :



Untuk mebuat contoh program diatas, lakukan langkah langkah sebagai berikut :

1. klik ganda pada icon label yang terdapat pada tab standart dalam konponen Palette, sehingga muncul object yang bernama label1.
2. ubah properties caption-nya pada jendela object inspector menjadi Mulai, kemudian atur tata letaknya seperti pada gambar diatas.
3. lakukan cara yang sama untuk mendapatkan label Selesai dan Lama.
4. tambahkan object edit yang letaknya disebelah kanan label. Dan kosongkan properties Text-nya.
5. tambahkan oject button, letakkan dibawah object label dan edit seperti yang nampak pada gambar. Ubah properties captionya menjadi Start.
6. tambahkan object timer yang letaknya pada tab System dalam komponen palette.
7. tekan tombol F12 untuk menampilkan editor kode program, kemudian tuliskan kode programnya. Ingat Borland Delphi 7 sudah memberikan sebagian kode program anda tinggal melengkapinya saja. Untuk kembali ke design form tekan F12 lagi. Jadi tombol ini untuk menuju design Form dan kode program secara bolak balik. Atau dengan cara meng-kilk object yang akan diisi program.
8. setelah selesai, jalankan program diatas dengan memilih menu Run lalu pilih Run atau tekan tombol F9 atau klik Icon Run.
9. Kode program dapat anda lihat seperti dibawah ini:

```
Unit lat01;  
Interface
```

```
Uses
```

```
Windows, messages, SysUtils, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, ExtCtrls;
```

```
Type
```

```
Tform1 = class(Tform)  
    Timer1:Ttimer;  
    Edit1:Tedit;  
    Edit2:Tedit;  
    Edit3:Tedit;  
    Label1:Tlabel;  
    Label2:Tlabel  
    Label3:Tlabel
```

```

Button1:Tbutton;
Procedure FormCreate(Sender: Tobject);
Procedure Button1Clik(Sender: Tobject);
Procedure Timer1Timer(Sender: Tobject);
Private
    { Private Declaration }
public
    { Public Declaration }
end;
var
    Form1      :Tform1;
    Awal,akhir : tDateTime;
    Lama       : Real;
Implementation
(SR *.DFM )

    procedure TForm1.FormCreate (Sender: Tobject);
    begin
        timer1.Interval:=1;
    end;

    procedure TForm1.Button1Click(Sender: Tobject);
    begin
        if Button1.Caption='Start' then
            begin
                awal:=time;
                edit1.text:=timetostr(time);
                Button1.caption='Stop';
            end
        else if button1.caption='Stop' then
            Button1.caption='Selesai'
        Else
            Aplication.terminate
    End;

    Procedure TForm1.Timer1Timer(Sender: Tobject);
    Var slama : String;
    Begin
        If Button1.Caption='Stop' Then
            Begin
                Akhir:=time;
                Edit2.text:TimeToStr(akhir);
                Lama:=(akhir-awal)*100000;
                Str(lama:12:2,Slama);
                Edit3.text:=slama;
            End;
        End;
    End.

```

1.3.2Menyimpan Program Aplikasi

Sebelum program dijalankan disarankan untuk disimpan dahulu. Untuk penyimpanan ada beberapa cara dan pilihan. Untuk menghindari kesalahan biasakan untuk memilih **Save All** atau **Save project as**. Setelah anda pilih pilihan itu maka anda akan disuruh menyimpan sebanyak dua kali yaitu menyimpan form/unitnya dan menyimpan projectnya dari unit tersebut.

1.3.3 Menutup Program Aplikasi

Untuk menutup program aplikasi Borland Delphi 7 anda cukup memilih file kemudian pilih **Close all**, untuk keluar Klik File kemudian **Exit** atau tombol close yang terletak pada pojok kanan atas.

1.4 Beberapa Komponen Borland Delphi 7

Untuk dapat menguasai pemrograman Borland Delphi 7 dengan baik maka kita harus tau beberapa komponen yang ada di Borland Delphi 7 dan mempelajarinya dengan baik, adapun komponen tersebut antara lain:

1.4.1 Project

Project adalah sekumpulan form, unit dan beberapa hal lain artinya project adalah program itu sendiri. File project disimpan dengan akhiran .dpr, beberapa file yang diperlukan untuk project adalah sebagai berikut :

a. File Unit (.pas)

Dipakai untuk menyimpan program (kode program), unit ini berhubungan langsung dengan form, tapi kadangkala unit hanya berupa procedure dan function yang tidak berhubungan dengan form.

b. File Form (.dfm)

Dipakai untuk menyimpan semua informasi mengenai form

c. File project Option(.dfo)

Dipakai untuk menyimpan semua setting option (pilihan) project.

d. File Resource(.res)

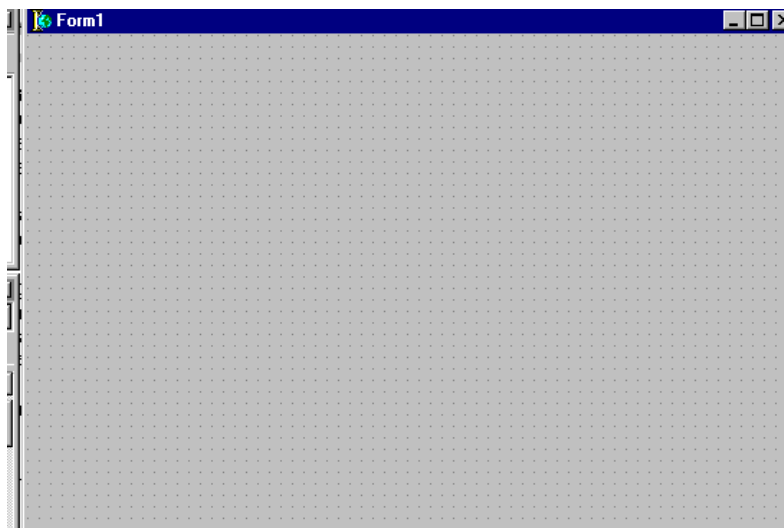
Dipakai untuk menyimpan icon yang dipakai project.

e. File Backup(.~dp, .~df, .~pa)

Dipakai untuk menyimpan file backup project, form dan unit.

1.4.2 Form

Form adalah suatu object yang dipakai sebagai tempat bekerja program aplikasi. Dalam satu project bisa dibuat lebih dari satu form. Setiap form mengandung unit. Unit dalam form dipakai untuk mengatur dan mengendalikan form, gambar form seperti nampak pada gambar dibawah ini :



1.4.3 Unit

Unit adalah modul kode program, satu program mungkin memiliki lebih dari satu unit. Setiap kali kita membuat form maka otomatis Borland Delphi 7 memberikan unitnya juga. Ada juga unit yang terpisahkan dengan form yang berisi sekumpulan procedure dan fungsi.

Manfaat penggunaan unit dalam program antara lain sebagai berikut :

- Membagi program yang besar kedalam beberapa unit, sehingga kita dapat mengedit unit-unit tertentu saja.
- Membuat library (daftar pustaka) berupa procedure dan function sehingga memudahkan sharing antar program.
- Unit dapat dikompilasi terpisah dari program aplikasi, sehingga program aplikasi lain yang memerlukan unit serupa dapat menggunakannya tanpa harus menulis kembali kode program tersebut.

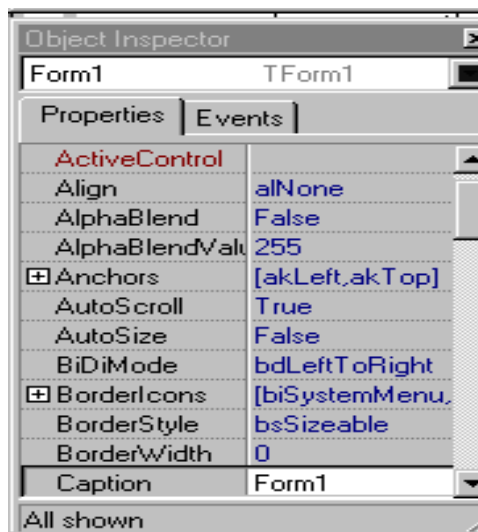
1.4.4 Program

Sebuah program secara umum mempunyai struktur sebagai berikut :

- Heading Program, yaitu bagian yang menunjuk nama program tersebut.
- Pernyataan USES, yang berisi daftar unit yang dipakai program
- Blok deklarasi dan pernyataan, yaitu bagian yang berisi deklarasi dan pernyataan program yang dilaksanakan pada saat program dijalankan. Bagian ini harus diakhiri dengan pernyataan End diikuti tanda titik.

1.4.5 Properties

Properties digunakan untuk menentukan setting suatu object. Suatu object biasanya mempunyai beberapa properti. Properties sendiri letaknya bersebelahan dengan event pada jendela Object Inspector. Gambar properties dapat dilihat pada gambar dibawah ini :



1.4.6 Event

Event adalah peristiwa atau kejadian yang diterima oleh suatu object misalkan klik, drag dan lain lain. Sebagai contoh dapat dilihat potongan kode program sebagai berikut :

```

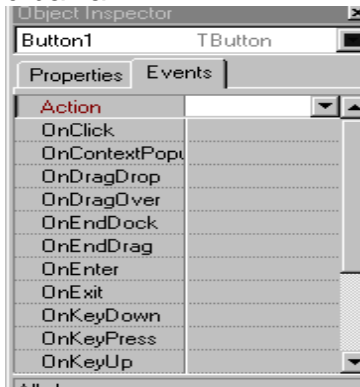
procedure TForm1.Button1Click(Sender: TObject);
begin
  if Button1.Caption='Start' then
  begin
    awal:=time;
    edit1.text:=timetostr(time);
    Button1.caption='Stop';
  end
  else if button1.caption='Stop' then
    Button1.caption='Selesai'
  Else
    Application.terminate

```

End;

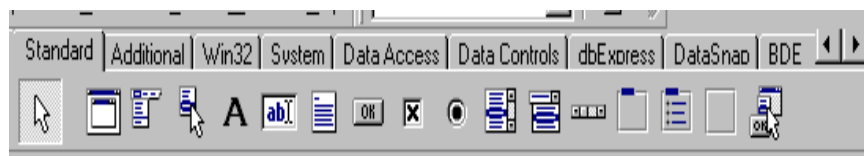
Program diatas menunjukkan event Click pada object button1.

Beberapa pilihan event pada object button dapat dilihat pada gambar dibawah ini :



1.4.7Komponen Palette

Komponen palette adalah kumpulan object yang dipakai untuk mendesign program. Beberapa komponen palette anantara lain komponen standart, additional, win32, System dan lain lain yang berisi object yang berbeda beda sesuai dengan masing masing komponen tersebut. Gambar komponen pallete dapat dilihat pada gambar dibawah ini :



1.5 Kegunaan Delphi

1. Untuk membuat aplikasi windows
2. Untuk merancang aplikasi program berbasis grafis
3. Untuk membuat program berbasis jaringan (client/server)
4. Untuk merancang program .Net (berbasis internet)

1.6 Keunggulan Delphi

1. IDE (***Integrated Development Environment***) atau lingkungan pengembangan aplikasi sendiri adalah satu dari beberapa keunggulan delphi, didalamnya terdapat menu – menu yang memudahkan kita untuk membuat suatu proyek program.
2. Proses Kompilasi cepat, pada saat aplikasi yang kita buat dijalankan pada Delphi, maka secara otomatis akan dibaca sebagai sebuah program, tanpa dijalankan terpisah.
3. Mudah digunakan, source kode delphi yang merupakan turunan dari pascal, sehingga tidak diperlukan suatu penyesuaian lagi.
4. Bersifat multi purphase, artinya bahasa pemograman Delphi dapat digunakan untuk mengembangkan berbagai keperluan pengembangan aplikasi.

1.7 File-file Pendukung Project Delphi

- a) File Project (.Dpr)
File ini disimpan dengan ber-ekstenion **.dpr**. File ini berisi informasi mengenai seluruh proyek program
- b) File Unit (.Pas)

File ini merupakan kumpulan dari barisan kode program yang terdapat di jendela code editor, baik itu yang dituliskan oleh programmer maupun oleh system. Extension file ini adalah **. pas**

File Unit dibagi menjadi 3 :

- ✓ Unit Form
Unit ini dibuat secara otomatis oleh Delphi. Satu unit untuk satu form.
- ✓ Unit component
Unit yang terbentuk pada saat membuat komponen baru.
- ✓ Unit umum
Unit yang dibuat untuk tipe data, variable, procedure dan class yang dapat digunakan dan diaplikasikan.

c) File Form (.Dfm)

Berisi tentang seluruh informasi yang ada kaitannya dengan form yang dibuat, meliputi tinggi, lebar, posisi form atau tentang komponen didalamnya. Penggunaan file ini tidak dianjurkan karena untuk pengaturan sudah disediakan **object inspector** sebagai media pengaturan semua komponen.

1.8 Aplikasi Console

Aplikasi console merupakan sebuah aplikasi yang tidak mempunyai form. Aplikasi ini berbasis text mode dan umumnya berjalan pada command prompt. Delphi menyediakan fasilitas untuk membangun aplikasi console. Aplikasi ini digunakan untuk membuat sourcecode object pascal. Memulai Aplikasi Console yaitu dengan cara mengklik menu File-New-Other-Console Application.

1.9 Beberapa aturan penulisan dalam Pascal/Delphi :

- ✓ Akhir sebuah program Pascal ditandai dengan tanda baca titik (.) setelah END yang paling akhir.
- ✓ Tanda titik koma (;) merupakan pemisah antar instruksi satu dengan lainnya.
- ✓ Beberapa statement boleh ditulis menjadi satu baris dipisahkan dengan tanda baca titik koma (;)
Contoh : `simpan := akhir; akhir := simpan + awal;`
- ✓ Baris komentar diletakkan diantara tanda (* dan *) atau diantara tanda { dan }
Contoh : `Var rerata : real; (*nilai rata-rata*) Nil1 : real; {nilai ujian}`

BAB II

TIPE DATA

Secara sederhana tipe data dapat didefinisikan dengan istilah tempat untuk menentukan pemberian nilai terhadap suatu variabel sesuai atau tidak dengan nilai yang diberikan oleh user. Dalam versi lain tipe data juga diartikan sebagai batasan terhadap fungsi tanda pengenal terhadap semua nilai yang diterima. logika yang dapat kita berikan adalah ketika kita menempatkan tanda pengenal **harga** hanya mengenal angka, maka ketika kita memberikan nilai berupa string maka secara otomatis data tersebut akan ditolak karena nilai tersebut tidak dikenali oleh **tipe data** yang diberikan.

Ada beberapa hal yang perlu diperhatikan dalam menentukan tipe data :

- ✓ Penggunaan memori
Masing-masing tipe data memiliki perbedaan dalam hal penggunaan memori. Dalam hal ini usahakan untuk menggunakan tipe data yang memiliki memori yang kecil.
- ✓ Ketelitian Penghitungan
Walaupun unsure penghematan memori sangat penting untuk diperhatikan, namun kebenaran dalam hal ketelitian penghitungan jauh lebih penting. Oleh karena itu gunakanlah tipe data yang memiliki tingkat ketelitian yang tinggi.

2.1 Tipe Data

a. Tipe Integer.

Tipe data integer digunakan untuk menyatakan bilangan yang tidak mempunyai angka desimal, yang termasuk tipe data integer antara lain

Tipe	Rentang Nilai	Byte
Byte	0-255	1
Word	0-65535	2
ShortInt	-128 – 127	1
SmallInt	- 32768 - 32767	2
Integer	- 2147483648 - 2147483647	4
Cardinal	2147483647	4
LongInt	0 –2147483647 -2147483648 2147483647	4

b. Tipe Real

Tipe data real digunakan untuk menyatakan bilangan yang mempunyai angka desimal, yang termasuk tipe data real antara lain :

Tipe	Rentang Nilai	Byte
Real48	2.9×10^{-39} s.d 1.7×10^{38}	6
Single		4
Double	4.1×10^{-45} s.d 3.4×10^{38}	8
Extended		10
Comp	5.0×10^{-324} s.d 1.7×10^{308}	8
Currency	3.6×10^{-4951} s.d 1.1×10^{-4932} $-2^{63}+1$ s.d $2^{63}-1$ - 922337203685477.5808 s.d 922337203685477.5807	8

c. Tipe Boolean

Tipe data boolean digunakan untuk menyatakan data logika yaitu True (T) dan False (F), yang termasuk tipe data boolean antara lain :

Tipe	Byte
Boolean	1
ByteBool	1
WordBool	2
LongBool	4

Disarankan yang digunakan adalah tipe Boolean karena tipe yang lain hanya digunakan untuk menjaga kompatibilitas yaitu jika program dihubungkan dengan program dalam bahasa yang lain atau dengan windows yang menggunakan tipe yang sama.

d. Tipe Character

Tipe data Character digunakan untuk menyatakan karakter satu huruf. Yang termasuk tipe Character antara lain :

Tipe	Byte	Isi
Char	1	1 Character
AnsiChar	1	Ansi
WideChar	2	1 Character Ansi 1 Character Unicode

e. Tipe String

Tipe data string dipakai untuk menyatakan sederetan karakter yang membentuk satu kesatuan, misal nama , alamat dll. Yang termasuk tipe data string antara lain :

Tipe	Byte	Isi Maximum
ShortString	2 s.d 256	256
AnsiString	4 s.d 2gb	character
WideString	4 s.d 2gb	2^{31} Character 2^{30} Character

f. Tipe Array

Array adalah variabel tunggal yang dapat dipakai untuk menyimpan sekumpulan data sejenis. Sebagai contoh perhatikan potongan program berikut :

```

Var
    Bulan : array[1..12] of ansiString;
Begin
    Bulan[1]:='Januari';
    Bulan[2]:='Februari';
    ....
    ....
    Edit1.text:=bulan[1];

```

g. Tipe Record

Tipe data record dipakai untuk menyimpan sekumpulan data yang mungkin berbeda tipe, tapi saling berhubungan, perhatikan contoh dibawah ini:

Type

```
RecBrg=record  
Kode : AnsiString[6];  
Nama : AnsiString[30];  
Hsatuan : Single;  
End;
```

Var

```
Barang :recBrg;
```

Begin

```
Barang.kode:='PS.001';  
Barang.nama:='Pensil 2B' ;  
Barang.harga:=2500;
```

h. Tipe Terenumerasi dan Subrange

Tipe Terenumerasi dan Subrange dipakai untuk menyatakan data berurutan yang bertipa sama, perhatikan contoh dibawah ini:

- contoh Tipe data Terenumerasi

```
type  
warna=(merah,biru,hijau,kuning,orange,hitam,putih);  
var  
    warnacat:warna;
```

- contoh Tipe data Subrange

```
type  
    warnaku          = Biru .. Orange;  
    hurufBesar      = 'A' .. 'Z';  
    Nilai           = 0..100;
```

Var

```
Warnacat : Warnaku;  
Nilaiujian :nilai;  
Nilaiakhir :hurufbesar;
```

2.2 Konstanta

Konstanta adalah sesuatu nilai yang bersifat tetap, untuk lebih jelasnya perhatikan contoh berikut ini :

Begin

```
.....  
.....  
discountanggota:=jumlah*0.1;  
discounttunai:=jumlah*0.15;
```

potongan program diatas dapat dituliskan menjadi seperti contoh berikut ini :

Const

```
Discountanggota=0.1;  
Discounttunai=0.15;
```

Begin

```
.....  
.....  
jumlahdiscountanggota:=jumlah*discountanggota;  
jumlahdiscounttunai:=jumlah*discounttunai;
```

Keuntungan penggunaan pernyataan konstanta antara lain:

- a. Program lebih mudah dimengerti/terbaca
- b. Menghindarkan salah ketik, karena jika anda salah mengetik nilai konstanta, maka pada saat kompilasi tidak dianggap salah. Sedangkan jika salah mengetik nama konstanta maka pada saat kompilasi akan dianggap salah.
- c. Jika dalam suatu program nilai konstanta dipakai berulang kali, maka jika ada perubahan data nilai konstanta, cukup yang diubah deklarasi konstantanya saja.

2.3 Variabel

Variabel adalah suatu tempat dalam memori komputer yang diberi nama (sebagai pengenalan) dan dialokasikan untuk menampung data sementara.

Perbedaan antara variabel dengan konstanta adalah sebagai berikut :

- a. Konstanta dipergunakannya pada saat kompilasi program, sedang variabel dipergunakan pada saat program dijalankan.
- b. Konstanta tidak dapat berubah pada saat program berjalan. Sedangkan variabel dapat berubah atau diubah nilainya pada saat program berjalan.

Aturan pemberian nama variabel adalah sebagai berikut :

- a. Harus dimulai dengan karakter alfabet(huruf) atau under score (_).
- b. Harus unique (tidak boleh ada nama yang sama) pada ruang lingkup variabel yang sama.
- c. Panjangnya boleh berapa saja, tetapi hanya 63 karakter pertama yang dibaca dan 64 seterusnya akan diabaikan.
- d. Beberapa karakter tidak diperbolehkan dipakai antara lain : , + - * / < >

Contoh penggunaan variabel dapat dilihat pada potongan program berikut ini :

```
Var
```

```
    Form1:Tform1;
```

```
    A : Single;
```

```
Implementation
```

```
( SR *.DFM )
```

```
procedure Tform1.FormCreate(Sender: TObject);
```

```
begin
```

```
    A:=5;
```

```
    // variabel ini bersifat global
```

```
end;
```

```
procedure Tform1.FormClick( Sender: TObject);
```

```
var
```

```
    A : Single;
```

```
Begin
```

```
A:=10;
```

```
// variabel ini bersifat lokal
```

```
end;
```

2.4 Operator

Hirarti operator dalam Borland Delphi 7 adalah sebagai berikut:

Operator	Hierarki
@, not	Tertinggi
*, /, div, mod, and, shl, shr, as	Kedua
+, -, or, xor	Ketiga
=, <>, <, >, <=, >=, in, is	Keempat

Anda juga bisa mengubah hierarki dengan memberi tanda kurung pada operasi yang akan didahulukan.

a. Operator Penugasan (Assignment)

Operator penugasan disimbolkan dengan tanda sama dengan didahului titik dua dan berfungsi untuk memasukkan sesuatu kedalam suatu variabel, penulisanya adalah sebagai berikut :

<variabel> := <ekspresi>

contoh :

bilangan :=0;

harga :=500;

banyak :=5;

jumlah:=banyak*harga;

no:=no+1;

b. Operator Aritmatika

Digunakan untuk melakukan operasi aritmatika. Operator aritmatika terdiri dari :

Operator	Operasi	Tipe diproses	Tipe Hasil
*	Perkalian	Integer, real	Integer, real
/	Pembagian	real	real, real
div	Pembagian	Integer, real	Integer
mod	bulat	real	Integer
+	Sisa	Integer	Integer, real
-	pembagian	Integer	Integer, real
	Penambahan	Integer, real	
	Pengurangan	Integer, real	

Contoh penggunaan operator aritmatika dapat dilihat pada contoh dibawah ini :

A:= 2* 3; // hasilnya 6

A:= 5 / 2 ; // hasilnya 2.5

A:= 5 div 2 ; // hasilnya 2 dibulatkan kebawah

A:=5 mod 2 ; // hasilnya 1 dibulatkan keatas

A:= 5 + 2 ; // hasilnya 7

A:= 5 - 2 ; // hasilnya 3

Contoh pangkat :

A:= 2 ;

B:= 3 ;

C:=exp(B*Ln(A)); // C berisi 9

D:=exp(B*Ln(5)); // D berisi 125

c. Operator Relasi

Operator relasi digunakan untuk membandingkan suatu data dengan data yang alain dan menghasilkan suatu nilai logika benar atau salah. Operator penggabungan dalam operator relasi nampak pada tabel dibawah ini:

Operator	Keterangan
=	Sama dengan
<>	Tidak sama dengan
<	Lebih kecil
>	Lebih besar
<=	Lebih kecil atau sama dengan
>=	Lebih besar atau sama dengan

Contoh penggunaan operator relasi adalah sebaga berikut :

```
A:= 2 > 3 ; // False
A:= 2+2 >3 ; // True
A:= 3 > 2+ 2 ; // False
A:= 3 >= 3 ; // True
```

d. Operator Logika

Operator logika digunakakn untuk mengekspresikan satu atau lebih data logika yang menghasilkan data logika yang baru. Macam operator logika adalah sebagai berikut:

Operator	Keterangan
Not	Tidak
And	Dan
Or	Atau
Xor	Exclusive or

- tabel penggunaan operator NOT tampak pada tebel berikut ini:

Ekpresi	Hasil
Not	False
True	True
Not	
False	

Contoh hasil penggunaan operator logiak NOT adalah sebagai berikut :

```
Tes := not (5>4);//tes berisi false
Tes := not (5=4);//tes berisi true
```

- tabel penggunaan operator AND tampak pada tebel berikut ini:

Ekspresi	Hasil
False and	False
false and	False
False and	False
true and	True
True and	
false and	
True and	
true and	

Contoh hasil penggunaan operator logiak AND adalah sebagai berikut :

```
Tes := (5<4) and (3<2) ;           //tes berisi false
Tes := (5<4) and (3>2) ;           //tes berisi false
Tes := (5>4) and (3<2) ;           //tes berisi false
Tes := (5>4) and (3>2) ;           //tes berisi true
```

- tabel penggunaan operator OR tampak pada tabel berikut ini:

Ekspresi	Hasil
False or	False
false	True
False or true	True
True or false	True
True or true	

Contoh hasil penggunaan operator logiak OR adalah sebagai berikut :

```
Tes := (5<4) or (3<2) ;           //tes berisi false
Tes := (5<4) or (3>2) ;           //tes berisi true
Tes := (5>4) or (3<2) ;           //tes berisi true
Tes := (5>4) or (3>2) ;           //tes berisi true
```

- tabel penggunaan operator XOR tampak pada tabel berikut ini:

Ekspresi	Hasil
False xor	False
false	True
False xor	True
true	False
True xor	
false	
True xor true	

Contoh hasil penggunaan operator logiak XOR adalah sebagai berikut :

```
Tes := (5<4) xor (3<2) ;           //tes berisi false
Tes := (5<4) xor (3>2) ;           //tes berisi true
Tes := (5>4) xor (3<2) ;           //tes berisi true
Tes := (5>4) xor (3>2) ;           //tes berisi false
```

BAB 3

STRUKTUR KONDISI

3.1.1 Percabangan If

Digunakan untuk menjalankan satu pernyataan atau satu blok pernyataan, tergantung dari nilai yang diuji.

➤ **Percabangan untuk kondisi pilihan tunggal**

Merupakan operator percabangan yang digunakan untuk menentukan sebuah pilihan dengan kondisi tunggal

Bentuk Umum

If Syarat **then** hasil;

Contoh

If Nilai > 80 **then** keterangan = 'Lulus';

➤ **Percabangan untuk kondisi majemuk**

Merupakan operator percabangan yang digunakan untuk menentukan pilihan dengan kondisi yang harus dipeuhi lebih dari satu.

Bentuk Umum

If Syarat1 **then**

Hasil 1

Else

If syarat2 **then**

Hasil2 **end**;

Contoh

If nilai > 80 **then**

Grade = "A"

Else

if nilai > 70 **then grade = "B" else**

if nilai > 60 **then grade = "C" else**

grade = "E";

3.1.2 Percabangan Bersyarat Case

Digunakan untuk menjalankan satu pernyataan atau satu blok pernyataan, tergantung dari nilai yang diuji. Biasanya digunakan untuk yang mempunyai banyak pilihan.

Penulisanya adalah sebagai berikut :

```
Case <ekspresipilihan> of
  <daftar1> : <pernyataan1>;
  .....
  .....
  <daftarx> : <pernyataanx>;
end;
atau,
```

```
Case <ekspresipilihan> of
  <daftar1> : <pernyataan1>;
  .....
  .....
```

```
    <daftarx> : <pernyataanx>;  
    else  
        <pernyataanx>;  
end;
```

<ekspresipilihan> adalah suatu ekspresi bernilai ordinal. Program akan menguji satu persatu <daftar> yang ada. Jika <daftar> ada atau benar maka <pernyataan> disebelah kananya akan dikerjakan, jika tidak ada maka <pernyataan> setelah else yang akan dikerjakan. Contohnya adalah sebagai berikut :

```
// contoh pertama  
case I of  
    1 .. 5 : snilai := 'Rendah';  
    6 .. 10 : snilai := 'Tinggi';  
    0,11.. 99 : snilai := 'Tidak Sah';  
else  
    snilai := ' ';  
end;
```

```
// contoh kedua  
case warna of  
    merah : x :=1;  
    hijau : x :=2;  
    biru : x :=3;  
    kuning,orange,hitam : x :=0;  
end;
```


BAB IV

STRUKTUR PENGULANGAN

3.1 Struktur Pengulangan Borland Delphi 7

Merupakan control program yang digunakan untuk suatu proses yang akan berjalan terus menerus. Kondisi perulangan merupakan proses berjalannya program secara terus menerus dan akan berhenti ketika proses mendapatkan kondisi yang sudah ditentukan.

Di dalam Delphi mengenal 3 jenis perulangan :

a. Perulangan For – To - Do

Perulangan dengan statement for adalah perulangan yang digunakan untuk melakukan suatu proses dalam sebuah blok program. Proses perulangan For – To – Do dimulai dengan nilai terkecil ke besar.

Bentuk Umum :

For variabel := nilai awal to nilaiakhir statement

Contoh

```
procedure TForm4.Button1Click(Sender: TObject);
var
i: integer;
begin
for i := 1 to 5 do
begin
edit1.SelText := inttostr(i);
end;
end.
```

Bila Program tersebut dijalankan maka nilai i akan ditampilkan pada edit 1 dengan hasil sebagai berikut =

A screenshot of a Delphi edit control (TEdit) showing the number '12345'. The text is displayed in a monospaced font within a rectangular box with a thin border.

Catatan : semua variabel yang berhubungan dengan perulangan harus mempunyai tipe data sama.

b. Perulangan For – DownTo-Do

Perulangan For-Dowbto-Do adalah perulangan yang menghitung suatu proses dengan nilai awal besar dan nilai akhirnya lebih kecil, maka variabel sebagai control program yang diperoleh adalah dari besar ke kecil.

Bentuk Umum

For Variabel := nilai-awal Downto Nilai-akhir Do Pernyataan

Contoh

```
procedure TForm4.Button2Click(Sender: TObject);
var
i : integer;
begin
for i := 5 bownto 1 do
edit2.SelText := inttostr(i);
end;
```

Bila Program tersebut dijalankan maka nilai i akan ditampilkan pada edit 1 dengan hasil sebagai berikut :

54321

c. Pengulangan While Do

Perintah ini untuk mengulang satu pernyataan atau satu blok pernyataan jika atau selama (while) suatu kondisi (syarat) bernilai true, penulisannya adalah sebagai berikut :

While <ekspresi> do <pernyataan>

Jalannya struktur pengulangan diatas adalah sebagai berikut :

- a. Program akan menguji nilai <ekspresi>
- b. Jika <ekspresi> berniali false (salah) maka <pernyataan> tidak dilaksanakan dan langsung kelangkah e.
- c. Jika <ekspresi> bernilai true (benar), maka <pernyataan> akan dilaksanakan satu kali.
- d. Kembali ke prosedur a
- e. Program menghentikan pengulangan dan menjalankan baris berikutnya (jika ada)

Berikut adalah contoh perulangan menggunakan while....do :

Contoh

```
procedure TForm4.Button3Click(Sender: TObject);
var
  i : integer;
begin
  i := 0;
  while i < 5 do
  begin
    edit3.5elText := inttostr(i);
    i := i + 1;
  end;
end;
```

Bila program tersebut dijalankan maka hasil yang akan diperoleh adalah sebagai berikut :

01234

3.1.3 Pengulangan Repeat Until

Pengulangan ini hampir sama dengan pengulangan while ... do, perbedaanya antara lain sebagai berikut :

1. dapat melaksanakan pengulangan lebih dari satu pernyataan, baik berupa kumpulan pernyataan tunggal atau kumpulan blok pernyataan, jadi segala sesuatu yang ada diantara perintah repeat Until akan diulang.
2. pasti melaksanakan (minimal satu kali) segala sesuatu yang berada diantara pernyataan repeat ... until , karena pengujian terletak di bagian bawah pernyataan.

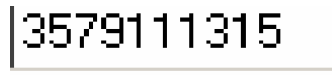
Penulisan struktur Repeat ... until adalah sebagai berikut :

Repeat - Statement/penyataan — Until - - - ungkapan logika

Contoh

```
procedure TForm4.Button4Click(Sender: TObject);
var
  i : integer;
begin i := 1;
repeat
  i := i + 2;
  edit4.5elText := inttostr(i);
until i = 15;
end;
```

Bila program tersebut dijalankan maka akan diperoleh hasil sebagai berikut :



Berikut ini perbedaan ketiga macam pengulangan tersebut :

For..to..do	While..do	Repeat..until
<ul style="list-style-type: none">▪ Pengulangan sampe batas maksimum yang diberikan▪ Bersifat ascending dan descending▪ Tidak membutuhkan syarat atau kondisi dalam pengulangan	<ul style="list-style-type: none">▪ Pengulangan berdasarkan syarat atau kondisi tertentu▪ Syarat atau kondisi pengulangan berada di awal pengulangan▪ Pengulangan akan terus berlangsung selama kondisi terpenuhi▪ Pengulangan akan berhenti jika kondisi tidak terpenuhi	<ul style="list-style-type: none">▪ Pengulangan berdasarkan syarat atau kondisi tertentu▪ Syarat atau kondisi pengulangan berada di akhir pengulangan▪ Pengulangan akan terus berlangsung jika kondisi tidak terpenuhi▪ Pengulangan akan berhenti jika kondisi terpenuhi

BAB V PEMROGRAMAN MODULAR

5.1. Procedure

Prosedur adalah suatu program terpisah dan berdiri dalam suatu blok program dan berfungsi sebagai sebuah sub program (program bagian). Penulisan prosedur diawali dengan kata **Procedure** pada bagian deklarasi program dan cukup menuliskan nama prosedur yang dibuat pada bagian **Implementasi**.

Alasan penggunaan prosedur

1. Digunakan untuk penggalan program yang akan digunakan secara berulang – ulang dalam suatu proses program.
2. Digunakan untuk memecah – mecah program menjadi sebuah modul program, sehingga listing program menjadi lebih sederhana.

Syarat penulisan nama procedure

1. harus diawali dengan karakter.
2. untuk nama prosedur dengan menggunakan dua kata atau lebih penulisannya tidak boleh menggunakan spasi, harus digabung atau dihubungkan dengan underscore (_)
3. tidak mengenal tanda baca.

Contoh

a. Pendeklarasian Prosedur

Penulisan pendeklarasian sebuah prosedur dalam Delphi ditempatkan setelah kata **public** agar dapat dikenali oleh semua object yang ada dalam proses program tersebut. Penulisannya diawali dengan kata **Procedure** diikuti nama procedure.

```
Public  
    Procedure bersih;
```

b. Penulisan prosedur Penulisan pogram prosedur terdapat dalam bagian

```
implementation,  
Procedure TForm1.Bersih;  
Begin  
    Enama.text := "  
    Ealamat.text := "  
    Ekota := "  
End;
```

c. Pemanggilan Prosedur

Pemanggilan prosedur terdapat dalam bagian implementation, pada umumnya pemanggilan prosedur terdapat dalam suatu ruang lingkup object yang mempunyai sebuah event. Cara pemanggilannya cukup hanya dituliskan nama prosedurnya saja.

```
Procedure TForm1.button1click(sender: TObject);  
Begin  
    Bersih;  
End;
```

atau ketika program prosedur digunakan pada object yang lain untuk memanggil program yang sama kita hanya cukup menuliskan nama prosedurnya tanpa dibutuhkan pendeklarasian prosedur baru.

5.2. Function

Fungsi adalah merupakan sebuah penggalan program yang terpisah dari program utama dan berfungsi sebagai sebuah program bagian dari program utama. Penulisan fungsi diawali dengan kata cadangan *function* dan dideklarasikan dalam bagian deklarasi fungsi. Dan penulisan program fungsi ditempatkan pada program utama. Satu hal yang perlu diperhatikan dalam penulisan fungsi adalah harus diikuti dengan tipe datanya.

a. Bentuk Umum penulisan Fungsi

Function indentifier (daftar-parameter) : type;

Contoh

Function hitung (var a,b : real) :real;

Penulisan blok fungsi diawali dengan kata cadangan *begin* dan diakhiri dengan *end*;

Contoh :

```
Function Hitung ( Var A, B : integer ) : integer ;
Begin
  Hitung := A+B ;
End ;
Var
  X, Y : integer ;
BEGIN
  Write ( ' Nilai X ? ' ) ; readln ( X ) ;
  Write ( ' Nilai Y ? ' ) ; readln ( Y ) ;
  Writeln ;
  Writeln ( X, ' + ', Y, ' = ', Hitung ( X, Y ) ) ;
  Readln ;
END.
```

} Program Utama

Parameter pada fungsi sama dengan parameter yang ada pada prosedur yaitu :

a. Pemanggilan secara Nilai (By Value)

Contoh :

Uses crt ;

Function Hitung (X, Y, Z : integer) : integer ;

Begin

Z := X+Y ;

Writeln (' Nilai X = ', X) ;

Writeln (' Nilai Y = ', Y) ;

Writeln (' Nilai Z = ', Z) ;

End ;

Var

A, B, C : integer ;

BEGIN

A := 5 ; B := 7 ; C := 3 ;

Hitung (A, B, C) ;

Writeln (' Nilai A =', A, ' Nilai B = ', B, ' Nilai C = ', C) ;

Readln ;

END.

a. Pemanggilan secara Reference (Acuan)

Contoh :

Uses crt ;

Function Hitung (var A, B, C : integer) : integer ;

Begin

 Hitung := A+B ;

 C := A * B ;

End ;

Var

 X, Y, Z : integer ;

BEGIN

 Write (' Nilai X : ') ; readln (X) ;

 Write (' Nilai Y : ') ; readln (Y) ;

 Writeln ;

 Writeln (X , ' + ' , Y , '=' , Hitung (X, Y, Z)) ;

 Writeln (X , ' * ' , Y , '=' , Z) ;

 Readln ;

END.

BAB VI ARRAY

6.1. Pengertian Array

Array (larik) merupakan tipe data terstruktur dimana didalamnya terdiri dari komponen – komponen yang mempunyai tipe data yang sama. Didalam suatu array jumlah komponen banyaknya adalah tetap. Didalam suatu larik atau array setiap kompoenen ditunjukkan oleh suatu index yang unik. Index dari setiap komponen array menunjukkan urutan data atau identitas yang mewakili data yang ada didalamnya.

Logika sederhananya array itu bisa disamakan dengan dua orang dengan nama yang sama didalam suatu komunitas, untuk membedakan antara nama yang satu atau dengan nama yang lain maka diberikan initial tambahan untuk setiap nama.

6.2. Deklarasi Array

Didalam penulisan bahasa pemograman setiap penggunaan array harus dideklarsikan terlebih dahulu. Pendeklarasian array diawali dengan nama variabel array diikuti dengan indeks array yang dituliskan didalam tanda “[]” , diikuti dengan kata cadangan of dan tipe data yang dibutuhkan.

Bentuk Umum Penulisan

Tanda_pengenal : array [..tipe index ..] of

tipe data; **Contoh :**

Var

A : array[1. .4] of integer;

B : array[1..5] of

string; C:

array[1..10] of

real;

Keterangan :

A,B,C merupakan tanda pengenal/ nama variabel dari array;

1. .4 : merupakan tipe indek dari array, yang menunjukkan banyaknya data yang mampu disimpan.

Integer : menunjukkan bahwa data yang diinput berupa bilangan bulat.

6.3. Alokasi Penggunaan Array a.

a. Array Static (*Static Array*)

array static adalah model pendeklarasian array dimana tipe data yang digunakan mempunyai nilai yang tetap. Nilai yang digunakan untuk menentukan jangkauan pada umumnya bernilai integer. Array Static juga bisa disebut Array dengan deklarasi tipe indeks subrange integer.

Bentuk Umum

array[indexType1, ..., indexTypeN] of baseType

Keterangan = index type menunjukkan tipe data ordinal yang menunjukkan batasan atau elemen maksimal terhadap seberapa besar variabel tersebut menyimpan komponen.

Contoh

Var arrayku :

array[1..5] of char

Atau juga

type

jangkauan = 1..5;

var

nilai : array[jangkauan] of integer;

b. Array Dinamis (*Dynamic arrays*)

Larik atau array dinamis merupakan array yang tidak mempunyai suatu jangkauan atau ukuran yang tetap. Tetapi ketika program dijalankan maka memori untuk suatu array dinamis direalokasikan ketika kita menugaskan suatu nilai kepada array. Dynamic-Array jenis ditandai oleh konstruksi (menyangkut) format

Bentuk Umum

array of baseType

Contoh

var nilai: array of Real;

Dari deklarasi tersebut nilai yang merupakan deklarasi array belum memperoleh nilai yang tetap, tetapi hanya diberikan batasan sebagai tipe data real. Untuk mendeklarasikan array tersebut kita harus menempatkan array didalam suatu memori, caranya adalah dengan memanfaatkan fungsi dari perintah *sellength*.

Sellength(nilai,20)

Dari penggalan program tersebut nilai untuk array nilai tersebut mempunyai range sebanyak atau cakupan 20 untuk tipe data real, dengan indeex dimulai dari 0 sampai dengan 20.

BAB VII OPERASI PADA STRING

a. Menggabungkan String

Menggabungkan string adalah proses merangkai dua karakter atau lebih menjadi satu kalimat atau kata yang baru. Operator string yang dikenal dalam Bahasa Delphi adalah operator dengan symbol '+'.

b. Menghapus Teks

Prosedur standart ini digunakan untuk menghapus atau mengurangi sebagian atau seluruh karakter terhadap string atau teks.

Bentuk umum

penulisan

Delete(teks, index,

jumlah)

Delete adalah prosedur standart yang digunakan untuk menghapus teks. *Teks* merupakan kalimat atau string yang akan dihapus, *Index* menunjukkan posisi awal yang akan dihapus. *Jumlah* menunjukkan jumlah karakter yang akan dihapus.

Model Penulisan

```
var
    nama : string[20];
    saya : string[15];
    namasaya : string[50];
Begin
    Nama := 'Nama Saya adalah
    = '; Saya := 'Irnawan';
    Namasaya := 'nama+'
    '+saya; Ehasil.text :=
    namasaya;
End.
```

c. Mencari Panjang teks (length)

Fungsi ini digunakan untuk menghitung panjang atau jumlah karakter dari suatu teks atau kalimat. Dalam menghitung jumlah karakter dalam suatu teks spasi akan dibaca sebagai satu karakter. Sebagai catatan hasil yang diperoleh dari perhitungan adalah bilangan bulat positif.

Bentuk Umum

Length(teks)

d. Mencari Karakter Pada Teks (POS)

Fungsi ini digunakan untuk mencari letak atau posisi dari suatu karakter dari suatu nilai string. Nilai yang dihasilkan adalah berupa byte.

Bentuk umum

Pos(sustr, string);

Contoh Kasus :

1. Menggunakan Struktur Kondisi

- a) Menghitung Gaji Pegawai
 - 1) Design Form

Form Data Gaji Pegawai

DAFTAR GAJI PEGAWAI DINAS "XXX"

Data Pegawai

NIP: ENIP

Nama Pegawai: ENama

Golongan: ComboBox1

Gaji Pokok: 0

Status:

Sendiri

Menikah

Jumlah Anak: EAnak OK Tunjangan Anak: 0

Potongan dan Lain-lain

Iuran Koperasi EKoperasi

Bank BTN EBTN

Iuran Anggota EAnggota

Pajak 10% EPajak

Pendapatan

Total Pendapatan: 0

Jumlah Potongan: 0

Gaji Bersih: 0

Buttons: Data Baru, Hitung, Keluar

Ketentuan :

- Design interface awal aplikasi adalah seperti pada gambar diatas
- Ketika pertama kali program dijalankan maka akan ada aksi berikut :
 - Label jumlah anak, Edit Anak, tombol OK, Label tunjangan anak, label 0 akan disembunyikan
 - Edit EKoperasi, Edit EBTN, Edit EAnggota dan Edit EPajak akan disembunyikan
- Ketika Tombol Data Baru diklik maka ENIP dan ENama dibersihkan. Kemudian juga Nilai Total Pendapatan, Jumlah potongan dan gaji bersih berisikan nilai 0. Untuk combobox1 langsung berubah menjadi Pilih. Berikut tampilannya ketika pertama kali dijalankan :

Form Data Gaji Pegawai

DAFTAR GAJI PEGAWAI DINAS "XXX"

Data Pegawai

NIP:

Nama Pegawai:

Golongan: Pilih

Gaji Pokok: Rp 0

Status:

Sendiri

Menikah

Buttons: Data Baru

Potongan dan Lain-lain

Iuran Koperasi

Bank BTN

Iuran Anggota

Pajak 10%

Pendapatan

Total Pendapatan: 0

Jumlah Potongan: 0

Gaji Bersih: 0

Buttons: Hitung, Keluar

- Ketika pilihan status menikah diklik maka Label jumlah anak, Edit Anak, tombol OK, Label tunjangan anak, label 0 akan ditampilkan
- Ketika salah satu atau semua pilihan check pada potongan dan lain-lain dicentang maka EKoperasi, EBTN, EAnggota dan EPajak akan ditampilkan. Berikut ini tampilan aplikasi programnya jika sudah diisi dengan data :

- Berikut diberikan daftar gaji pokok pegawai berdasarkan golongan :

Golongan	Gaji Pokok (Rp)
IIA	1.200.000
IIB	1.300.000
IIIA	1.900.000
IIIB	2.200.000
IVA	2.900.000
IVB	3.100.000

- Ketika pertama kali dijalankan, pada form tidak tampil atribut : Jumlah Anak, EAnak, Tombol OK, Tunjangan Anak dan label nilai 0 (akan muncul ketika status menikah diklik)
- Pada kelompok potongan, EKoperasi, EBTN, EAnggota dan EPajak akan disembunyikan ketika form pertama kali dijalankan dan akan muncul ketika masing-masing pilihan cek box dicentang dan akan kembali disembunyikan ketika centangannya dilepas.
- Jika jumlah anak ≥ 2 maka tunjangan anak = $5\% \times \text{Gaji Pokok}$
- Nilai pajak ditentukan dengan $10\% \times \text{Gaji Pokok}$
- Nilai Total Pendapatan dihitung dengan menjumlahkan Gaji Pokok dengan Tunjangan Anak
- Sedangkan, Jumlah Potongan dihitung dengan menjumlahkan seluruh jenis potongan yang ada
- Gaji bersih dicari dengan rumus : Total Pendapatan – Jumlah Potongan

2) Sourcode Program

- ✓ Menentukan Golongan (Combobox1)

```
procedure TForm7.ComboBox1Change(Sender: TObject);
begin
  case ComboBox1.ItemIndex of
    0:gapok:= 1200000;
    1:gapok:= 1300000;
    2:gapok:= 1900000;
    3:gapok:= 2200000;
    4:gapok:= 2900000;
    5:gapok:= 3100000;
  end;
  LGapok.Caption:= CurrToStr(gapok);
  LGapok.Caption:= FormatCurr('Rp ##,##0', gapok);
end;
```

- ✓ Menentukan perhitungan status

```
procedure TForm7.RadioGroup1Click(Sender: TObject);
begin
  if RadioGroup1.ItemIndex = 0 then
  begin
    tunj_anak:=0;
    Label5.Visible:= false;
    EAnak.Visible:= false;
    BTunjangan_Anak.Visible:= false;
    label7.Visible:= false;
    LTunjangan.Visible:= false;
  end else
  begin
    Label5.Visible:= true;
    EAnak.Visible:= true;
    BTunjangan_Anak.Visible:= true;
    label7.Visible:= true;
    LTunjangan.Visible:= true;
    EAnak.SetFocus;
  end;
end;
```

- ✓ Tombol menentukan tunjangan anak

```
procedure TForm7.BTunjangan_AnakClick(Sender: TObject);
begin
  j_anak:= StrToInt(EAnak.Text);
  if j_anak >= 2 then
    tunj_anak:= 0.5*gapok
  else
    tunj_anak:= 0;
    LTunjangan.Caption:= CurrToStr(tunj_anak);
    LTunjangan.Caption:= FormatCurr('Rp #,##0', tunj_anak);
  end;
```

- ✓ Menentukan potongan
 - a. Koperasi


```

procedure TForm7.koperasiClick(Sender: TObject);
begin
  if koperasi.Checked then
  begin
    kop:= 10000;
    EKoperasi.Text:= CurrToStr(kop);
    EKoperasi.Text:= FormatCurr('Rp #,##0', kop);
    EKoperasi.Visible:= true;
  end else
  begin
    kop:=0;
    EKoperasi.Visible:= false;
  end;
end;
          
```
 - b. Bank BTN


```

procedure TForm7.btnClick(Sender: TObject);
begin
  if btn.Checked then
  begin
    bbtn:= 650000;
    EBTN.Text:= CurrToStr(bbtn);
    EBTN.Text:= FormatCurr('Rp #,##0', bbtn);
    EBTN.Visible:= true;
  end else
  begin
    bbtn:=0;
    EBTN.Visible:= false;
  end;
end;
          
```
 - c. Iuran Anggota


```

procedure TForm7.anggotaClick(Sender: TObject);
begin
  if anggota.Checked then
  begin
    angg:= 5000;
    EAnggota.Text:= CurrToStr(angg);
    EAnggota.Text:= FormatCurr('Rp #,##0', angg);
    EAnggota.Visible:= true;
  end else
  begin
    angg:=0;
    EAnggota.Visible:= false;
  end;
end;
          
```
 - d. Pajak


```

procedure TForm7.pajakClick(Sender: TObject);
begin
  if pajak.Checked then
  begin
    pj:= 0.10*gapok;
    EPajak.Text:= CurrToStr(pj);
          
```

```

EPajak.Text:= FormatCurr('Rp #,##0', pjk);
EPajak.Visible:= true;
end else
begin
pjk:=0;
EPajak.Visible:= false;
end;
end;

```

✓ Menentukan group pendapatan

```

procedure TForm7.BHitungClick(Sender: TObject);
begin
tot_pend:= gapok+tunj_anak;
jml_pot:= kop+bbtn + angg + pjk;
gaber:= tot_pend-jml_pot;
LTotPend.Caption:= CurrToStr(tot_pend);
LTotPend.Caption:= FormatCurr('Rp #,##0', tot_pend);
LPotongan.Caption:= CurrToStr(jml_pot);
LPotongan.Caption:= FormatCurr('Rp #,##0', jml_pot);
LGaber.Caption:= CurrToStr(gaber);
LGaber.Caption:= FormatCurr('Rp #,##0', gaber);
end;

```

✓ Tombol keluar

```

procedure TForm7.BKeluarClick(Sender: TObject);
begin
close;
end;

```

end.

2. Menggunakan Struktur Pengulangan
 - a) Membuat perhitungan transaksi penjualan
 - 1) Design Form

Ketentuan :

- ✓ Ketika tombol ISI DATA LAGI diklik maka seluruh field yang berisikan nilai akan dibersihkan
- ✓ Ketika form pertama kali dijalankan maka nilai tanggal sudah otomatis terisikan
- ✓ Memasukkan jumlah item untuk menginputkan daftar nama barang dan harganya ketika tombol OK diklik dan langsung mengisi nilai dari total penjualan
- ✓ Tombol close untuk keluar dari program

2) Sourcode Program

- ✓ Isi data lagi

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  Edit1.Text:="";
  //Edit2.Text:="";
  Edit3.Text:="";
  ListBox1.Clear;
  ListBox2.Clear;
  Edit4.Text:="";
  Edit1.SetFocus;
end;

end.
```

- ✓ OK

```
procedure TForm1.Button1Click(Sender: TObject);
var
  inputnama:string;
  inpuharga:string;
  i,j:integer;
  harga:real;
begin
  harga:=0;
  j:=0;
  j:= StrToInt(Edit1.Text);
  if j<= 0 then
    begin
      ShowMessage('Data tidak boleh lebih kecil dari nol');
      exit;
    end else
    begin
      for i:= 1 to j do
        begin
          inputnama:= InputBox('Input : ','Ketikkan Nama Barang','');
          inpuharga:= InputBox('Input', 'Ketikkan Harga Barang','');
          ListBox1.Items.Add(inputnama);
          ListBox2.Items.Add(inpuharga);
          harga:= harga+StrToFloat(inpuharga);
        end;
      end;
      Edit4.Text:= FloatToStr(harga);
      Edit4.Text:= FormatFloat('Rp #,##0',harga);
    end;
end;
```


✓ Close
procedure TForm1.Button3Click(Sender: TObject);
begin
close;
end;

✓ Tanggal (klik 2 kali di area form)
procedure TForm1.FormCreate(Sender: TObject);
begin
DateSeparator:='-';
ShortDateFormat:='dd/mm/yyyy';
Edit2.Text:='Hari ini : ' + DateToStr(date);
end;

3) Berikut tampilan ketika program diisi data

Form Transaksi Penjualan

Daftar Penjualan

Total Item yang dibeli: 2 [OK]

Tanggal: Hari ini : 22-11-2011

No Faktur: []

Daftar Barang Yang Dibeli

- baju
- celana

Daftar Harga Barang

- 25000
- 50000

Total Penjualan: Rp 75,000

[ISI DATA LAGI] [CLOSE]

DAFTAR PUSTAKA

Pranata, Antony, *Tip dan Trik Pemrograman Delphi*, Andi Offset, Yogyakarta, 1997

Pranata, Antony, *Pemrograman Borland Delphi*, Andi, Yogyakarta, 1998