

Minggu ke : 2
Penyusun : Dian Usdiyana
Rini Marwati
Materi : 2. Tipe Data
2.1 Tipe Data Ordinal
2.2 Tipe Data Real

Dalam pembembutaan suatu program, Anda tidak dapat menghindarkan diri dari pemakaian tipe data. Oleh karena itu, pengetahuan tentang berbagai macam tipe data sangat diperlukan. Delphi menyediakan enam macam tipe data yang dapat digunakan dalam pemrograman, yaitu tipe data sederhana, string, terstruktur, pointer, procedural, dan variant. Tipe data sederhana meliputi dua macam, yaitu ordinal dan real. Tipe data ordinal menyatakan himpunan dari nilai-nilai yang bertingkat.

2.1 Tipe data ordinal

Yang tergolong dalam tipe data ordinal adalah: integer, character, boolean, enumerated, dan subrange.

Integer

Tipe data integer (bulat) banyak sekali kita temukan di dalam kegiatan sehari-hari. Secara umum Delphi mengenal dua macam tipe, yaitu data integer dan cardinal. Data integer memiliki rentang -2147483648 s.d. 2147483647, sedangkan data cardinal memiliki rentang 0 s.d. 4294967295.

Secara rinci, tipe data integer dapat dipilah menjadi beberapa tipe dasar, yaitu: ShortInt, SmallInt, LongInt, Int64, Byte, Word, dan LongWord. Rentang masing-masing tipe data tersebut terdapat pada Tabel 5.1.

Tabel 2.1 Tipe data integer

Tipe	Rentang			bit	Memori
ShortInt	-128	s.d.	127	8	1 byte
SmallInt	-32768	s.d.	32767	16	2 byte
LongInt	$-0,5 \times 2^{32}$	s.d.	$0,5 \times 2^{32}-1$	32	4 byte
Int64	$-0,5 \times 2^{64}$	s.d.	$0,5 \times 2^{64}-1$	64	8 byte
Byte	0	s.d.	255	8	1 byte
Word	0	s.d.	65535	16	2 byte
LongWord	0	s.d.	$2^{32}-1$	32	4 byte

Tipe data yang digunakan sangat berperan dalam pengaturan alokasi memori program yang Anda buat. Oleh karena itu pada saat Anda mendeklarasikan suatu variabel sebaiknya mempertimbangkan pemilihan tipe data ini. Misalnya, jika Anda akan menggunakan suatu variabel berisi bilangan positif yang dibatasi hanya sampai 65000 tentu akan salah jika mendeklarasikannya sebagai tipe Byte, tetapi juga kurang bijaksana jika mendeklarasikannya sebagai tipe LongWord.

2.2 Tipe data real

Tipe data real menyatakan suatu himpunan bilangan yang dapat dinyatakan dengan notasi *floating-point* ($a \times 10^b$). Delphi menyediakan enam macam tipe data real, yaitu: Real48, Single, Double, Extended, Comp, dan Currency. Secara lengkap tipe data real bersama dengan alokasi memorinya terdapat pada Tabel 2.5.

Secara umum tipe real, di dalam implementasinya, sesuai dengan Double dengan rentang nilai $5,0 \times 10^{-324} \dots 1,7 \times 10^{308}$.

Tabel 2.5 Tipe data real

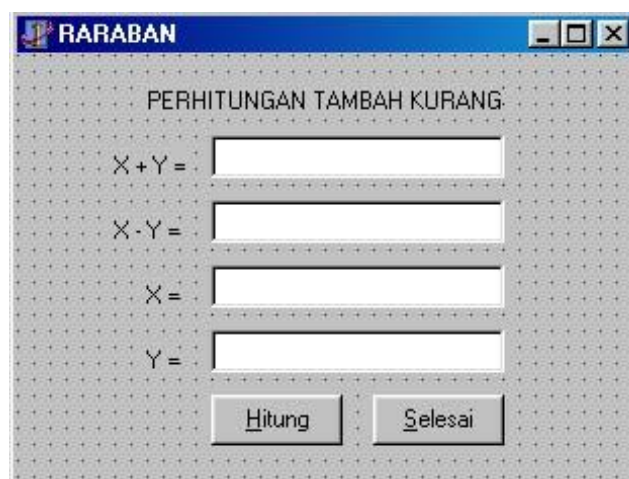
Type	Rentang			Memori
Real48	$2,9 \times 10^{-39}$	s.d.	$1,7 \times 10^{38}$	6 byte
Single	$1,5 \times 10^{-45}$	s.d.	$3,4 \times 10^{38}$	4 byte
Double	$5,0 \times 10^{-324}$	s.d.	$1,7 \times 10^{308}$	8 byte
Extended	$3,6 \times 10^{-4951}$	s.d.	$1,1 \times 10^{4932}$	10 byte
Comp	$-2^{63}+1$	s.d.	$2^{63}-1$	8 byte
Currency	-922337203685477,5808	s.d.	922337203685477,5807	8 byte

Untuk memahami bagaimana tipe data ordinal, seperti yang telah dijelaskan di atas, berikut ini disajikan contoh pembuatan proyek untuk membuat kalkulator sederhana yang diberi nama raraban.

Proyek 2.1:

Buatlah kalkulator sederhana yang hanya memiliki operasi hitung tambah, kurang, dan kali. Fasilitas yang terdapat di dalam kalkukator tersebut seperti terlihat pada Gambar 2.1

Buatlah tampilan untuk operasi tambah dan kurang, seperti pada gambar berikut:



The image shows a Windows-style application window titled "RARABAN". Inside the window, the text "PERHITUNGAN TAMBAH KURANG" is displayed. Below this text, there are four input fields arranged vertically. The first two fields are preceded by the labels "X + Y =" and "X - Y =" respectively. The next two fields are preceded by the labels "X =" and "Y =" respectively. At the bottom of the window, there are two buttons: "Hitung" (Calculate) and "Selesai" (Finish).

Gambar 2.1 Tampilan Form Raraban

1. Buat form baru, misalnya dengan menggunakan menu **File-New Application**, kemudian ubah propertiesnya menjadi sebagai berikut:

Properties	Isi
Caption	RARABAN
Name	FormRaraban

2. Buat judul masalah di atas dengan menggunakan komponen Label, caranya adalah sebagai berikut:
 - Klik satu kali tombol label
 - Tempatkan pointer pada posisi tertentu, kemudian klik satu kali
 - Ubahlah properties labelnya seperti berikut:

Properties	Isi
Caption	PERHITUNGAN TAMBAH KURANG
Name	LabelJudul

3. Siapkan masing-masing empat palette label dan edit, ubahlah masing-masing propertiesnya menjadi seperti pada tabel berikut:

Komponen	Properties	Isi
Label 1	Caption	$X + Y =$
	Name	LabelTambah
Label 2	Caption	$X - Y =$
	Name	LabelKurang

Komponen	Properties	Isi
Label 3	Caption	$X =$
	Name	LabelX
Label 4	Caption	$Y =$
	Name	LabelY

Komponen	Properties	Isi
Edit 1	Name	EditTambah
	Text	Kosongkan
	Enable	False
Edit 2	Name	EditKurang
	Text	Kosongkan
	Enable	False
Edit 3	Name	EditX
	Text	Kosongkan
	Enable	True
Edit 4	Name	EditY

	Text	Kosongkan
	Enable	True

4. Tambahkan pula dua tombol komponen button (Hitung dan Selesai) untuk melaksanakan perhitungan dan mengakhiri penghitungan, ubahlah propertiesnya seperti pada tabel berikut

Komponen	Properties	Isi
Button 1	Caption	&Hitung
	Name	Hitung
Button 2	Caption	&Selesai
	Name	Selesai

5. Klik dua kali tombol Selesai, kemudian lengkapi prosedur yang muncul menjadi seperti berikut:

```
procedure TFormRaraban.SelesaiClick(Sender: TObject);
begin
    Close;
end;
```

6. Klik dua kali tombol Hitung, kemudian lengkapi prosedur yang muncul menjadi seperti berikut:

```
procedure TFormRaraban.HitungClick(Sender: TObject);
var x,y,z:integer;
begin
    //Menyimpan bilangan dari EditX dan EditY
    //ke dalam x dan y
    x := StrToInt(EditX.Text);
    y := StrToInt(EditY.Text);

    //Operasi jumlah
    z := x + y;

    //Menampilkan hasil operasi Tambah
    EditTambah.Text := IntToStr(z);

    //Operasi kurang
    z := x - y;

    //Menampilkan hasil operasi Kurang
    EditKurang.Text := IntToStr(z);
end;
```

Pada langkah 6 Anda memasukkan deklarasi variabel-variabel dengan bilangan bertipe integer, `var x,y,z:integer;`. Selain itu ditambahkan pula pernyataan `x := StrToInt(EditX.Text);`, yang dapat dipahami bahwa `Text` yang terdapat di dalam `x`

diubah tipenya dari string menjadi integer dan selanjutnya dinyatakan sebagai x. Untuk y juga dapat diartikan sama dengan x.

Setelah proses memberi nilai kepada masing-masing variabel selesai, kemudian dilakukan proses perhitungan. Untuk operasi tambah dan kurang menggunakan operator + dan -, sedangkan untuk operasi kali menggunakan operator *.

Pernyataan `EditKutang.Text.Text := IntToStr(z);` dapat diartikan nilai z diubah dari data bertipe integer menjadi string, yang selanjutnya dinyatakan sebagai Text di dalam `EditTambah.Text` dan `EditKurang.Text`. Pernyataan ini menyebabkan hasil perhitungan ditampilkan pada jendela edit. Secara lengkap, proyek raraban ini terdapat pada List Program 2.1.

Unit selengkapnya terlihat seperti pada listing berikut

Listing 2.1: Menghitung Tambah Kurang

```
unit UnitRaraban;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  StdCtrls;

type
  TFormRaraban = class(TForm)
    LabelJudul: TLabel;
    LabelTambah: TLabel;
    LabelKurang: TLabel;
    LabelX: TLabel;
    LabelY: TLabel;
    EditTambah: TEdit;
    EditKurang: TEdit;
    EditX: TEdit;
    EditY: TEdit;
    Hitung: TButton;
    Selesai: TButton;
    procedure SelesaiClick(Sender: TObject);
    procedure HitungClick(Sender: TObject);
  private
    { Private declarations }

  public
    { Public declarations }
  end;
var
  FormRaraban: TFormRaraban;
implementation
{$R *.DFM}

procedure TFormRaraban.SelesaiClick(Sender: TObject);
begin
  Close;
```

```

end;

procedure TFormRaraban.HitungClick(Sender: TObject);
var x,y,z:integer;
begin
    //Menyimpan bilangan dari EditX dan EditY
    //ke dalam x dan y
    x := StrToInt(EditX.Text);
    y := StrToInt(EditY.Text);

    //Operasi jumlah
    z := x + y;

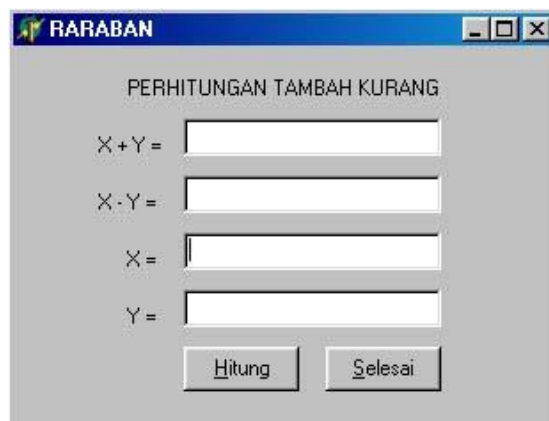
    //Menampilkan hasil operasi Tambah
    EditTambah.Text := IntToStr (z);

    //Operasi kurang
    z := x - y;

    //Menampilkan hasil operasi Kurang
    EditKurang.Text := IntToStr (z);
end;
end.

```

Hasil tampilan program tersebut adalah seperti pada Gambar 2.2 berikut ini



Gambar 2.2 Tampilan Program Raraban

Cobalah masukkan bilangan-bilangan genap untuk masing-masing nilai x dan y, kemudian tekan tombol Hitung. Apa yang terjadi? Nilai penjumlahan dan pengurangan langsung ditampilkan pada jendela editnya masing-masing.

Bagaimana jika Anda ingin mengulang perhitungan? Tentu saja program ini dapat melakukannya tetapi caranya dengan menimpa bilangan-bilangan yang tertera pada jendela edit untuk nilai x dan y. Agar lebih bagus sebaiknya ditambahkan tombol penghapus nilai-nilai yang ada pada jendela edit. Oleh karena itu, tambahkanlah satu palette Button, kemudian letakkan seperti pada Gambar 2.3 berikut:



Gambar 2.3 Tampilan Form Raraban dilengkapi penghapus

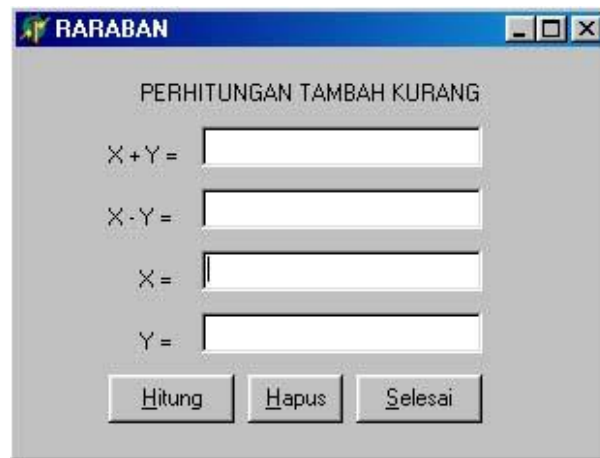
Ubahlah propertiesnya menjadi seperti berikut:

Komponen	Properties	Isi
Button 1	Caption	&Hapus
	Name	Hapus

Klik dua kali tombol Hapus, kemudian lengkapi prosedur yang muncul seperti berikut:

```
procedure TFormRaraban.HapusClick(Sender: TObject);
begin
    EditTambah.Text:='';
    EditKurang.Text:='';
    EditX.Text:='';
    EditY.Text:='';
end;
```

Jalankan program yang telah diperbaiki tersebut, maka akan muncul tampilan seperti Gambar 2.4 berikut



Gambar 2.4 Tampilan Program Raraban dilengkapi penghapus

Coba masukkan masing-masing nilai x dan y, kemudian tekan tombol Hitung. Maka akan muncul nilai-nilai hasil perhitungannya. Sekarang coba tekan Hapus, semua nilai yang ada pada jendele edit akan dihapus, program siap untuk menghitung nilai-nilai bilangan yang baru.