Menggunakan Modul dan Prosedur

Tujuan;

1. Membuat modul standar
2. Membuat variabel publik dan prosedur sendiri
3. Memanggil variabel dan prosedur dari event procedure

Modul standar adalah sebuah kontainer terpisah pada program yang mengandung variabel global (atau public) serta prosedur Function dan Sub.

A. Modul Standar

Pada praktikum sebelumnya, apabila Anda akan menuliskan program yang panjang, Anda akan membutuhkan beberapa fom dan event procedure yang menggunakan beberapa variabel dan rutin yang sama. Secara default, variabel mempunyai sifat lokal untuk sebuah event procedure, artinya variabel tersebut hanya bisa dibaca atau diubah dalam event procedure di mana ia dibuat. Begitu juga dengan event procedure, sifatnya lokal untuk form tempat ia diciptakan. Contohnya, Anda tidak dapat memanggil event procedure dari cmdOK_Click dari Form2 apabila event procedure tersebut berhuInterestn dengan Form1.

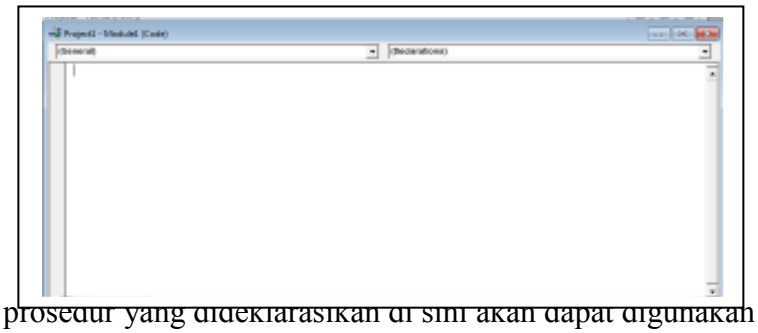
Untuk membangi variabel dan prosedur untuk semua form dan event procedure pada sebuah proyek, Anda perlu mendeklarasikannya pada salah satu modul standar untuk proyek tersebut. Modul standar adalah sebuah file khusus yang memiliki eksistensi .bas dan mengandung variabel dan prosedur yag dapat digunakan di seluruh bagian program. Sama halnya seperti form, modul standar dilampirkan secara terpisah pada jendela Project, dan bisa disimpan ke dalam disk menggunakan perintah Save Module1 As pada menu File. Tetapi modul standar tidak mengandung objek atau setting properti, melainkan hanya kode-kode yang dapat ditampilkan dan disunting pada jendela Code.

B. Membuat Modul Standar

Untuk membuat modul standar baru dalam program, klik Add Form pada toolbar, kemudian klik Module, atau klik perintah Add Module pada menu Project. Apabila Anda membuat modul standar yang baru, modul akan langsung tampak pada jendela Code. Modul standar yang pertama kan diberi nama Module1, tetapi Anda bisa mengubah nama ini pada saat melakukan penyimpanan ke dalam disk.

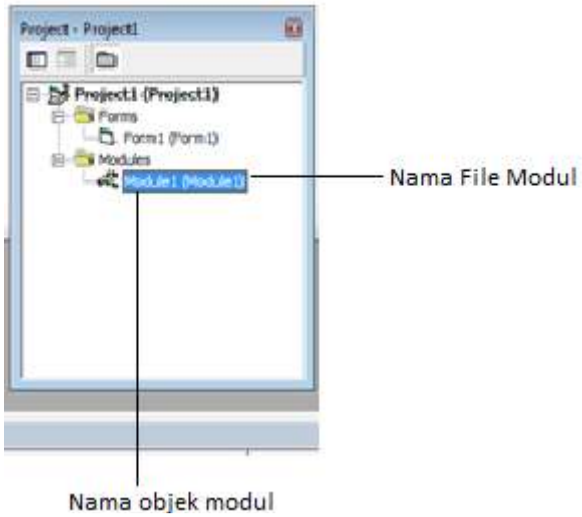
Membuat dan menyimpan modul standar

1. Bukalah sebuah proyek standar baru, kemudian klik perintah Add Module pada menu Project dan klik Open.



Variabel dan prosedur yang dideklarasikan di sini akan dapat digunakan oleh seluruh bagian program.

2. Untuk melihat keseluruhan jendela Project, klik ganda jendela Project.
   Jendela Project mucul seperti gambar berikut ini:



Jendela Project menampilkan modul standar yang Anda tambahkan ke dalam program dalam sebuah folder baru.

3. Pada menu File, klik perintah save Module1 As untuk menyimpan modul standar yang kosong ke dalam disk.
4. Simpan dengan nama MyTestMod.bas

   Modul standar telah disimpan ke dalam disk sebagai file .bas.
5. Jendela Properties muncul seperti gambar di bawah ini:

Karena modul tidak memiliki objek, maka properti hanya memiliki Name, yang berfungsi untuk menentukan nama objek untuk modul tersebut. Name tersebut juga dapat untuk membedakan modul satu dengan modul lainnya jika Anda membuat lebih dari satu modul, sesuai aturan pemberian nama modul dengan awalan mod.

6. Ubahlah properti Name menjadi modVariables kemudian enter.

C. Variabel Publik

Mendeklarasikan variabel gobal atau publik adalah dengan cara mengetikkan kata *Public* yang diikuti dengan nama variabel. Contoh:

Public RunningTotal

Akan mendeklarasikan sebuah variabel publik yang bernama RunningTool dalam modul standar. Secara default variable public dideklarasikan sebagai tipe variant dalam modul. Anda dapat menentukan tipe lain menggunakan perintah As dan menentukan tipenya. Contoh:

Public LastName As String

Praktek berikut ini akan mendemonstrasikan cara menggunakan variabel publik bernama Wins dalam modul standar. Anda kan membuka kembali Lucky Seven dan akan menggunkan variable Wins untuk mencatat beberapa kali putaran yang Anda menangkan.

Proyek Lucky Seven

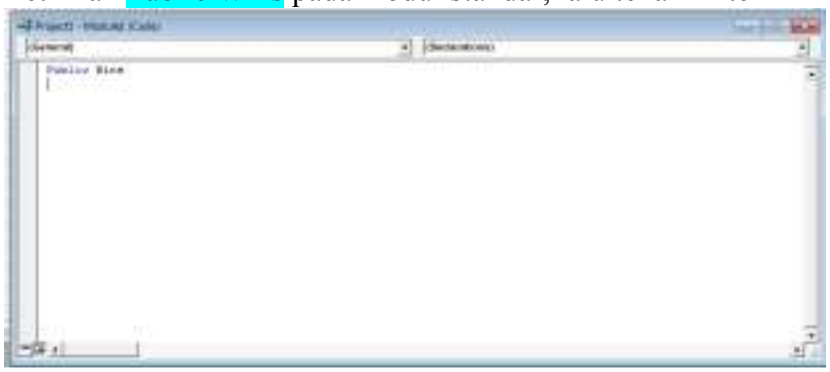
1. Open Project, klik No untuk membuang perubahan, lalu buka proyek Lucky Seven.
2. Bukalah object Lucky Seven
3. Jalankan program, kemudian klik tombol spin enam atau tujuh kali, dan catat kejadian tersebut.
4. Pada praktek ini Anda akan menambahkan sebuah label baru ke dalam form serta sebuah variable public yang menyimpan jumlah kemenangan yang Anda dapatkan.
5. Pada menu File klik perintah Save Lucky.frm As laulu simpan dengan nama MyWins.frm
6. Klik perintah Save Project As, dan simpan dengan nama MyWins.vbp.

Menambahkan modul standar

1. Klik kontrol Label, lalu buatlah label segiempat di bawah label Lucy Seven.
2. Ubahlah properti seperti di bawah ini:

Objek	Properti	Setting
Label5	Alignment	2-Center
	Caption	"Wins: 0"
	Font	Arial, Bold Italic, 12-point
	ForeColor	Green
	Name	LblWins
Form1	Caption	"Lucky Seven"

3. Tambahkan modul standar dengan perintah Add Module pada menu Project, kemudian Open
4. Ketikkan Public Wins pada modul standar, lalu tekan Enter



5. Pada menu File, klik perintah Save Module1 As, ketikkan MyWIns.bas, lalu tekan Enter
6. Tambahkan pernyataan berikut pada Event procedure Command1_Click untuk tombol Spin di bawah pernyataan Beep

```
Wins = Wins + 1
LblWins.Caption = "Wins: " & Wins
```

7. Save Project dan jalankan 10 kali, dan amati apa yang tejadi.

D. Membuat Prosedur General-Purpose

Selain untuk menyimpan variabel publik, modul standar juga dapat digunakan untuk menyimpan prosedur general-purpose yang bisa dipanggil dimana saja dalam program. Perbedaan antara Prosedure general-purpose dan event procedure adalah, prosedur general-purpose tidak berhuInterestn dengan event atau objek yang dibuat dengan kontrol toolbox.

Tiga jenis prosedur general-purpose dalam modul standar:

- Prosedur function.
  Prosedur ini dipanggil berdasarkan namanya dari dalam event procedure atau prosedur lain. Prosedur ini bisa menerima argumen dan selalu mengembalikan suatu nilai dalam nama fungsinya. Biasanya digunakan untuk perhitungan.

- Prosedur Sub
  Prosedur ini dipanggil berdasarkan namanya dari dalam event procedure atau prosedur lain. Bisa menerima argumen, dan juga dapat digunakan untuk melakukan tugas dalam prosedur dan mengembalikan nilai. Tetapi tidak seperti fungsi, Sub tidak mengembalikan nilai yang berhuInterestn dengan nama Sub tertentu (walaupun bisa mengembalikan nilai melalui nama variabel). Prosedur sub biasanya digunakan untuk menerima atau memproses masukan, menampilkan keluaran, atau mengatur properti.

- Prosedur Property.
  Prosedur ini digunakan untuk membuat dan memanipulasi properti buatan dalam program, sarana yang bermanfat untuk mengubah kontrol Visual Basic yang sudah da dan meningkatkan bhasa Visual Basic dengan membuat objek, properti, srta metode baru.

E. Menulis Prosedur Function

Pernyataan dalam fungsi ini melakukan suatu tugas, biasanya memproses teks, menangani masukan, atau menghitung nilai numerik. Arugumen adalah data yang digunakan untuk membuat fungsi bekerja.

Sintak fungsi

```
Function FunctionName([arguments]) [As Type]
    Function statements
End Function
```

Contoh prosedur function bernama TotalTax untuk menghitung pajak kota dan negara untuk sebuah item lalu membeikan hasilnya ke nama TotalTax:

```
Function TotalTax(Cost)
    StateTax = Cost * 0.05          'State tax is 5%
    CityTax = Cost * 0.015          'City tax is 1.5%
End Function
```

Untuk memanggil fungsi TotalTax pada event procedure, Anda harus menggunakan pernyataan yang serupa, yaitu:

```
lblTaxes.Caption = TotalTax(500)
```

fungsi TotalTax juga bisa menggunakan variabel sebagai argumennya, seperti contoh berikut:

```
TotalCost = SalesPrice + TotalTax(SalesPrice)
```

Baris ini menggunakan fungsi TotalTAx untuk menentukan pajak untuk nilai pada variable SalesPrice lalu menambahkannya ke dalam SalesPrice untuk mendapatkan Cost total dari sebuah item.

F. Menggunakan Fungsi untuk Melakukan Perhitungan

Pada praktek ini, Anda akan menambhakan sebuah fungsi ke dalam program Lucky Seven untuk menghitung tingkat kemenangan dalam permainan. Untuk melakukan hal tersebut, maka perlu ditambahkan sebuah fungsi bernama Rate dan sebuah variable publik bernama Spins ke dalam modul standar. Kemudian Anda akan memanggil fungsi Rate setiap kali tombol Spin di klik. Anda akan menampilkan hasilnya dalam sebuah label baru yang akan Anda buat dalam form.

1. Bukalah jendela project
   Komponen proyek MyWins.vbp muncul, Anda akan menyimpan komponen proyek ini dengan nama MyRate.
2. Klik tombol form MyWins.frm. pada menu File klik perintah Save MyWins.frm As. Simpanlah form tersebut dengan nama MyRate.frm
3. Klik modul standar MyWins.bas pada jendelan Project. Klik paerintah Save MyWins.bas As pada menu File, kemudian simpan dengan nama MyRate.bas
4. Pada menu File, klik perintah Save Project As simpanlah proyek tersebut dengan MyRate.vbp
5. Antar muka untuk proram Lucky Seven muncul
6. Pindahkan label Wins dekat abel Lucky Seven untuk menyediakan ruang bagi label baru.
7. Gunakan kontrol Label untuk membuat sebuah label baru di bawah label Wins. Ubahlah properti label tersebut:

Objek	Properti	Setting
Label5	Alignment	2 – Center
	Caption	“0.0%”
	Font	Arial, Bold Italic, 12-point
	ForeColor	Red
	Name	LblRate

8. Klik modul MyRate.bas, lalu klik tombol View Code pada jendela project.
   Modul standar Module1 muncul pada jendela Code.
9. Ketikkan variabel publik berikut ini di bawah pernyataan Public Wins:

```
Public Spins
```

10. Ketikkan deklarasi fungsi berikut

```
Function Rate(Hits, Attempts) As String
        Percent = Hits/Attempts
        Rate = Format(Percent, “0.0%”)
End Function
```

11. Tutuplah jendela Code, tampilkan event procedure Command1_Click.
    Di bawah baris keempat event procedure yang mengandung fungsi Rnd ketikkan pernyataan berikut:

```
Spins = Spins + 1
```

12. Lalu ketikkan pernyataan berikut sebagai baris terakhir pada event procedure Command1_Click, diantara pernyataan End If dan End Sub

```
lblRate.Caption = Rate(Wins, Spins)
```

13. Simpan dan jalankan program serta amati yang terjadi
14. Klik tombol spin 10 kali. Catat kemenangan Anda pada 5 kali pertama, lanjutkan hingga 10 kali.

G. Menulis Prosedur Sub

Prosedur Sub mirip dengan prosedur Function, kecuali Sub tidak mengembalikan nilai yang berhuInterestn dengan namanya. Sub biasanya digunakan untuk mendapatkan *input* dari pemakai, menampilkan atau mencetak informasi, atau memanipulasi beberapa properti yang berhuInterestn dengan suatu kondisi. Sub juga digunakan untuk memproses dan mengembalikan beberapa variabel selama pemanggilan fungsi. Sebagian besar fungsi hanya mengembalikan satu nilai, tetapi prosedur Sub bisa mengembalikkan banyak nilai.

Sintaks Prosedur Sub

Sintaks dasar dari prosedur Sub adalah sebagai berikut:

```
Sub ProcedureName([arguments])
    Procedure statements
End Sub
```

Keterangan:

- *ProcedureName* adalah nama prosedur Sub yang Anda buat
- *Arguments* adalah daftar argumen opsional (dipisahkan oleh koma, jika ada lebih dari satu) yang akan digunakan dalam Sub.
- *Procedure Statements* adalah blok pernyataan yang melaksanakan tugas dari prosedur.

Pada pemanggilan prosedur, nilai, dan tipe argumen yang dikirimkan ke dalam prosedur Sub harus sesuai dengan nilai dan tipe dari argumen yang berada dalam deklarasi Sub. Jika variabel yang dikirimkan ke dalam Sub diubah selama prosedur, variabel yang diubah tersebut akan dikembalikan ke dalam program. Secara *default*, prosedur Sub yang dideklarasikan dalam modul bersifat publik, juga bisa dipanggil oleh semua *event procedure.*

Anda bisa menggunakan prosedur Sub berikut ini untuk menambahkan nama ke dalam kotak daftar pada *form* pada saat program berjalan. Prosedur tersebut menerima satu variabel string yang dikirimkan berdasarkan referensi.

Jika prosedur Sub ini dideklarasikan dalam modul standar, ia bisa dipanggil dari semua *event procedure* pada program:

```
Sub AddNameToListBox(person$)
    If person$ <> " " Then
            Form1.List1.AddItem person$
            Msg$ = person$ & "added to the list box."
    Else
            Msg$ = "Name not specified"
    End If
    MsgBox (Pesan$) , ,  "Add Name"
End Sub
```

Prosedur AddNameToListBox menerima nama yang akan ditambahkan menggunakan argumen person$, sebuah variabel string yang diterima berdasarkan referensi selama pemanggilan prosedur. Jika nilai person$ tidak kosong atau *null*, nama yang ditentukan akan ditambahkan ke dalam objek kotak daftar List1 menggunakan metode AddItem, dan pesan konfirmasi ditampilkan oleh fungsi MsgBox. Jika argumen itu null, prosedur melompati metode AddItem dan menampilkan pesan "Add Name".

Memanggil Prosedur Sub

Untuk memanggil prosedur Sub pada program, Anda perlu menentukan nama prosedur tersebut lalu menampilkan argumen yang dibutuhkan oleh Sub. Sebagai contoh, untuk memanggil prosedur AddNameToListBox menggunakan string literal (memanggilnya berdasarkan nilai), Anda bisa mengetikkan pernyataan berikut ini:

```
    AddNameToListBox "Kimberly"
```

Hal yang sama, Anda bisa memanggil prosedur menggunakan variabel (memanggilnya berdasarkan referensi) dengan mengetikkan pernyataan berikut:

```
    AddNameToListBox NewName$
```

Pada kedua kasus di atas, prosedur AddNameToListBox akan menambahkan nama yang ditentukan ke dalam kotak daftar. Pada prosedur Sub ini, pemanggilan berdasarkan nilai dan referensi akan menghasilkan hasil yang sama karena argumennya tidak dimodifikasi dalam prosedur.

Keuntungan penghematan menggunakan prosedur Sub menjadi jelas apabila memanggil prosedur ini berulang kali, seperti ditunjukkan pada contoh di bawah ini:

```
AddNameToListBox "Kimberly"          'alway add two names
AddNameToListBox "Rachel"
Do
    NewName$ = InputBox("Enter a List Box Name." , "Add Name")
    AddNameToListBox$
Loop Until NewName$ = ""
```

Disini pemakai diizinkan memasukkan sebanyak mungkin nama ke dalam kotak list.

Menggunakan Prosedur Sub untuk Menangani Input

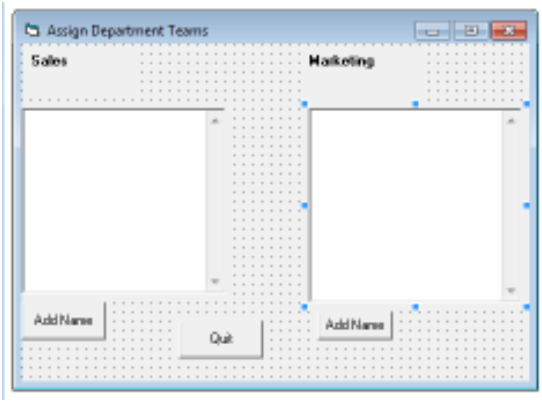
Prosedur seringkali digunakan untuk menngani input dalam program apabila informasi berasal dari dua atau lebih sumber dan ahrus memiliki format yang sama.

Praktek membuat prosedur Sub kotak teks

1. Pada menu file, klik perinah New Project, kemudian klik OK untuk membuka aplikasi standar yang baru. Sebuah form kosng muncul.
2. Gunakanlah kontrol ToolBox untuk membuat dua buah kotak teks yang berdampingan di tengah-tengah form
   Anda akan menggunakan kotak teks ini untuk menyimpan nma pegawai yang akan ditugaskan kepada dua departemen.
3. Gunakanlah kontrol Label untuk membuat dua buah label di atas kotak teks. Label-label ini mengandung nama departemen-departemen.
4. Gunkanlah kontrol CommandButton untuk membuat sebuah tombol perintah di bawah masing-masing kotak teks dan sebuah tombol perintah yang terpisah di bagian bawah form.
   Anda akan menggunaan dua tombol perintah pertama untuk menambah pegawai ke departemen mereka. Anda akan menggunakan tombol perintah terakhir untuk keluar dari program.
5. Ubahlah properti seperti pada tabel di bawah ini:

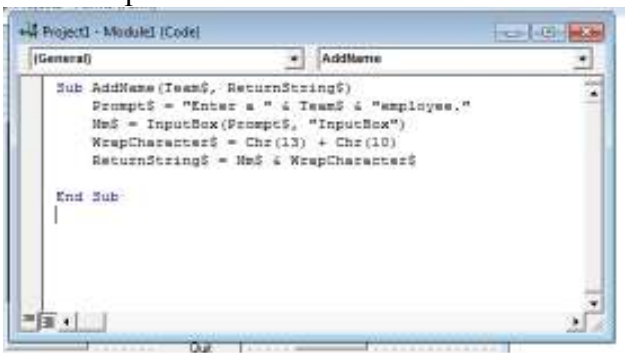
Objek	Properti	Setting
Text1	Text	(Empty)
	MultiLine	True
	ScrollBars	2-Vertical
	TabStop	False
	Locked	True
	Name	txtSaes
Text2	Text	(Empty)
	MultiLine	True
	ScrollBars	2-Vertical
	TabStop	False
	Locked	False
	Name	txtMkt
Label1	Caption	“Sales”
	Font	Bold
	Name	lblSales
Label2	Caption	“Marketing”
	Font	Bold
	Name	lblMkt
Command1	Caption	“Add Name”
	Name	cmdSales
Command2	Caption	“Add Name”
	Name	cmdMkt
Command3	Caption	“Quit”
	Name	cmdQuit
Form1	Caption	“Assign Departement Teams”

Bentuk form:

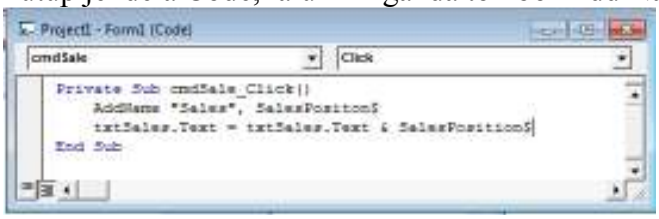


Kini Anda akan menambahkan sebuah modul standar dan membuat prosedur Sub bernama AddName.

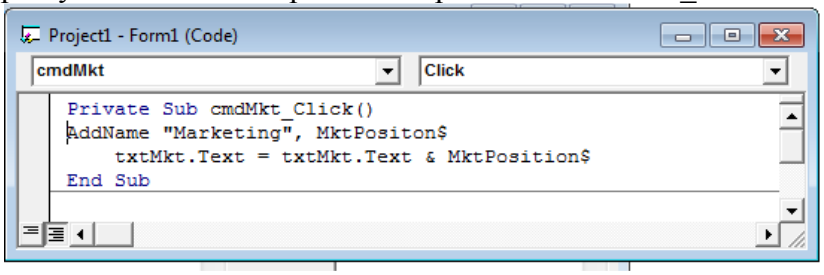
6. Pada menu Project, klik perinth Add Module, lalu klik OK
7. Ketikkan prosedur AddName berikut ini ke dalam modul standar:



8. Tutup jendela Code, lalu klik ganda tombol Add Name pertama dan ketikkan



9. Bukalah kotak daftar Objek pada jendela Code, lalu ketikkan cmdMkt. Ketikkan pernyataan berikut ini pada event procedure cmdMkt_Click:



10. Jalankan program, simpan modul standar dengan nama MyTeams.bas. Simpan project dengan nama MyTeams.frm. simpan proyek dengan nama MyTeams.vbp

H. Mengirimkan Argumen Berdasarkan Nilai

Pada pembahasan mengenai prosedur Sub, Anda telah belajar bahwa argumen bisa dikirimkan ke dalam prosedur berdasarkan referensi atau berdasarkan nilai. Apabila variabel dikirimkan berdasarkan referensi (*default*), semua perubahan yang dilakukan kepada variabel akan dikirimkan kembali ke prosedur pemanggilnya. Mengirimkan

argumen berdasarkan referensi memiliki beberapa keuntungan, asalkan Anda berhati-hati tidak mengubah variabel tersebut secara tidak sengaja dalam prosedur. Sebagai contoh perhatikanlah deklarasi dan pemanggilan prosedur Sub berikut ini.

```
Sub CostPlusInterest(Cost, Total)
    Cost = Cost * 1.05
    Total = Int(Cost)        ‘ buat menjadi integer dan kembalikan
End Sub
.
.
.
Price = 100
Total = 0
CostPlusInterest Price, Total
Print Price; “at 5% Interest  is” ; Total
```

Pada contoh ini, *programmer* mengirimkan dua buah variabel berdasarkan referensi kepada prosedur CostPlusInterest yaitu Price dan Total. *Programmer* berencana menggunakan variabel Total yang di-update pada metode Print berikutnya, tetapi sayangnya ia lupa kalau variabel Price juga ikut di-update pada langkah berikutnya dalam prosedur. (Karena Price dikirimkan berdasarkan referensi, perubahan pada Cost secara otomatis akan menghasilkan perubahan yang sama pada Price). Hal ini menghasilkan hasil *error* berikut ini apabila program dijalankan:

```
    105 at 5% Interest is 105
```

Perintah ByVal

Cara yang jelas untuk menghindari masalah ini adalah dengan tidak memodifikasi variabel yang dikirimkan kepada prosedur. Tetapi solusi ini membutuhkan kode program yang banyak dan tidak baik digunakan jika Anda bekerja dengan beberapa *programmer* lain. Cara yang lebih baik adalah menggunakan perintah ByVal pada daftar argumen pada waktu Anda mendeklarasi sebuah prosedur. Dengan cara ini, Visual Basic akan menyimpan salinan dari argumen asli dan mengembalikannya dengan tidak berubah apabila prosedur berakhir (bahkan jika variabel tersebut dimodifikasi dalam prosedur). ByVal digunakan pada daftar argumen dengan cara berikut ini:

```
    Sub CostPlusInterest(ByVal Cost, Total)
```

Apabila argumen Cost dideklarasikan menggunakan ByVal, program menghasilkan *output* yang benar:

```
100 at 5% Interest is 105
```

Mengirimkan Variabel Berdasarkan Nilai

Jika Anda tidak ingin menggunakan perintah ByVal, Anda bisa menggunakan cara lain untuk mencegah perubahan pada variabel yang dikirimkan: Anda bisa mengubahnya menjadi nilai literal dengan meletakkannya dalam tanda kutip. Trik yang jarang dipakai ini selalu berhasil pada Visual Basic, dan membuat pemanggilan prosedur Anda lebih mudah. Jika Anda mengirimkan variabel berdasarkan nilai, Anda lebih akan tahu apa

yang Anda maksudkan. Ini juga terkadang merupakan cara yang efisien untuk mengirimkan variabel berdasarkan nilai. Sintaks untuk memanggil prosedur CostPlusInterest dan mengirimkan variabel Price berdasarkan nilai adalah:

```
CostPlusInterest (Price), Total
```

Jika program contoh dipanggil dengan cara ini, hasil yang tepat dihasilkan

```
100 at 5% Interest  is 105
```