

PERTEMUAN KE-11

Bekerja dengan Koleksi dan Array

Tujuan:

1. Bekerja dengan koleksi
2. Memproses koleksi menggunakan loop For Each ... Next
3. Mengorganisir variabel ke dalam array

A. Bekerja dengan Koleksi Objek

Anda telah mengetahui bahwa objek-objek pada *form* disimpan bersama-sama dalam *file* yang sama. Tetapi apakah Anda juga mengetahui bahwa Visual Basic menganggap objek-objek tersebut merupakan anggota dari grup atau kelompok yang sama. Pada terminologi Visual Basic, seluruh kelompok objek pada *form* disebut dengan *Controls Collection* atau koleksi *Controls*. Kontrol *Controls* diciptakan secara otomatis pada saat Anda membuka *form* baru dan menambahkan objek ke dalam *form*. Pada kenyataannya, Visual Basic memelihara beberapa koleksi objek standar yang bisa Anda gunakan apabila Anda menulis program.

B. Notasi Koleksi

Setiap koleksi pada program memiliki namanya sendiri, jadi Anda bisa memanggilnya sebagai unit yang terpisah pada kode program. Sebagai contoh, seperti yang Anda pelajari, koleksi yang mengandung semua objek pada *form* disebut koleksi Controls. Tetapi karena Anda bisa memiliki lebih dari satu *form* pada program (dan oleh karena itu bisa memiliki lebih dari satu koleksi Controls), Anda perlu menyertakan nama *form* apabila menggunakan kontrol Controls pada program yang mengandung lebih dari satu *form*. Anda harus menggunakan nama berikut ini pada kode program:

Form1.Controls

Tanda titik di antara nama objek *Form1* dan Controls membuat Controls nampak seperti properti pada notasi ini, tetapi *programmer Visual Basic menjelaskan koleksi Controls sebagai objek yang dikandung oleh objek Form1*. Hubungan (atau hirarki) antara objek-objek sama seperti hubungan antara folder pada nama path. Selain bekerja dengan objek dan koleksi, Visual Basic juga mengizinkan Anda menjelajahi sistem untuk objek aplikasi lain dan menggunakannya pada program.

Anda bisa merujuk setiap objek pada koleksi, atau anggota individual dari koleksi dengan beberapa cara. Yaitu:

1. Cara pertama adalah dengan menentukan objek menggunakan nama mereka langsung pada pernyataan penugasan. Sebagai contoh, pernyataan berikut ini:

Form1.Controls!Label1.Caption = "Employees"

akan merujuk kepada objek Label1 pada koleksi Controls dan mengubah properti Caption menjadi "Employees.". Tanda seru(!) digunakan untuk menghubungkan objek Label1 dengan koleksi Control. Walaupun pernyataan ini mungkin rumit, tetapi dapat menjelaskan hirarki dalam koleksi tersebut.

2. Cara kedua untuk merujuk objek dalam koleksi adalah dengan menentukan posisi indeks dari objek tersebut dalam grup. Visual Basic menyimpan objek koleksi dalam urutan terbalik pada saat mereka diciptakan, jadi Anda bisa menggunakan “urut-urutan kelahiran” objek untuk merujuk beberapa objek secara individual, atau Anda bisa menggunakan loop untuk merujuk beberapa objek sekaligus. Sebagai contoh, untuk mengenali objek terakhir yang dibuat pada *form*, Anda harus menentukan indeks 0 (nol) seperti contoh berikut ini:

Form1.Controls(0).Caption = “Business”

Pernyataan ini mengubah properti Caption dari objek terakhir pada *form* menjadi “Business”. (Objek terakhir kedua yang dibuat memiliki indeks 1, objek terakhir ketiga yang dibuat memiliki indeks 2, dan seterusnya).

C. Menulis Loop For Each ... Next

Walaupun Anda bisa merujuk anggota individual dari suatu koleksi, cara paling bermanfaat untuk memanipulasi objek dalam koleksi adalah dengan memprosesnya sebagai grup. Pada kenyataannya, alasan kenapa koleksi digunakan adalah agar Anda bisa memproses kelompok objek secara efisien. Sebagai contoh, Anda mungkin menampilkan, memindahkan, mengurutkan, mengubah nama, atau mengubah ukuran seluruh koleksi objek sekaligus.

Untuk menangani tugas-tugas ini, Anda bisa menggunakan sebuah loop khusus bernama For Each...Next untuk berputar melewati objek dalam koleksi satu persatu. Loop For Each ... Next serupa dengan loop For ... Next. Apabila loop For Each ... Next digunakan dengan koleksi Controls, tampak seperti ini:

For Each Control in FormName.Controls

Process object

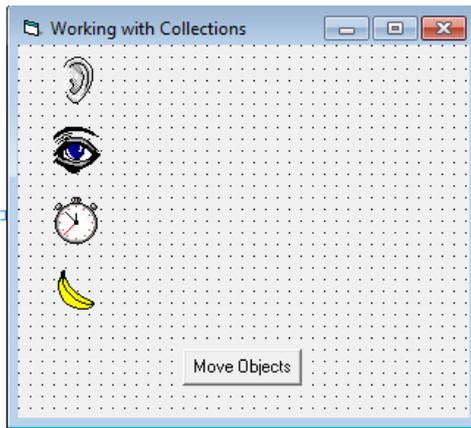
Next Control

Control adalah sebuah variabel khusus yang merepresentasikan objek saat ini dalam koleksi, dan *FormName* adalah nama *form*. Badan loop ini digunakan untuk memproses objek-objek individual dari koleksi. Sebagai contoh, Anda mungkin ingin mengubah properti Enabled, Left, Top, Caption, atau Visible dari objek-objek dalam koleksi, atau Anda ingin menampilkan nama setiap objek dalam kotak list.

Memindahkan Koleksi Objek

Pada raktek berikut ini, Anda akan menggunakan koleksi Controls untuk memindahkan sekelompok objek dari kiri ke kanan pada saat yang sama. Program ini menggunakan loop For Each...Next untuk memindahkan objek setiap kali pemakai mengklik tombol perintah bernama Move Object.

1. Jalankan Visual Basic
2. Buatlah form berikut:



3. Klik ganda Move Object, ketikkan pernyataan program berikut:

```
Private Sub cmdButton_Click()
  For Each Ctrl In Controls
    Ctrl.Left = Ctrl.Left + 200
  Next Ctrl
End Sub
```

4. Simpan dengan nama MyMove.frm dan MyMove.vbp

Menggunakan Properti Tag pada Loop For Each...Next

1. Klik save MyTag.frm dan MyTag.vbp
2. Ubahlah pernyataan program menjadi seperti berikut

```
Private Sub cmdButton_Click()
  For Each Ctrl In Controls
    If Ctrl.Tag <> "Button" Then
      Ctrl.Left = Ctrl.Left + 200
    End If
  Next Ctrl
End Sub
```

D. Menggunakan Koleksi Visual Basic

Visual Basic menyediakan dukungan built-in untuk koleksi-koleksi berikut ini selain koleksi Controls.

Tabel
Koleksi Dalam Visual Basic

Koleksi	Keterangan
Koleksi <i>Forms</i>	Koleksi semua <i>form</i> yang di-load ke dalam program. Menggunakan loop For Each...Next, Anda bisa mengatur karakteristik dari satu atau lebih <i>form</i> atau satu atau lebih objek yang terdapat dalam <i>form</i> .
Koleksi Printers	Koleksi semua printer yang tersedia pada sistem komputer Anda. Menggunakan For Each...Next dan metode AddItem, Anda bisa menampilkan nama-nama dari semua printer yang tersedia pada kotak list, lalu

menizinkan pemakai memilih salah satu printer untuk digunakan mencetak.

Koleksi
Database

Berbagai koleksi yang berhubungan dengan akses data dan pengelolaan *database*. Koleksi *database* yang sangat berguna di antaranya mencakup Columns, Containers, Indexes, dan *Databases*.

E. Bekerja dengan Array Variabel

Pada bagian ini, Anda akan belajar cara menggunakan teknik yang sama untuk menyimpan variabel dalam array. Sama seperti array kontrol dan koleksi, array variabel (disebut array saja) mengizinkan Anda merujuk seluruh kelompok nilai menggunakan satu nama lalu memproses setiap nilai secara individual atau sebagai grup menggunakan loop For...Next atau loop Do.

Array dibuat atau dideklarasikan dengan cara yang sama seperti variabel. Tetapi letak pendeklarasian array menentukan di mana array tersebut bisa digunakan pada program. Jika array dideklarasikan secara lokal, ia hanya bisa digunakan pada prosedur tempat ia dideklarasikan. Jika array dideklarasikan secara publik pada modul standar, ia bisa digunakan di mana saja pada program. Apabila Anda mendeklarasikan array, Anda perlu menyertakan informasi berikut ini pada pernyataan deklarasi:

- Nama Array : Nama yang Anda gunakan untuk merepresentasikan array dalam program. Secara umum, nama array mengikuti aturan yang sama dengan nama variabel.
- Tipe Data : Tipe data yang akan Anda simpan dalam array. Dalam kebanyakan kasus, semua variabel dalam array memiliki tipe yang sama. Anda bisa menentukan salah satu dari tipe data dasar, atau jika Anda tidak yakin tipe data yang akan disimpan pada array atau Anda ingin menyimpan lebih dari satu tipe data, gunakanlah tipe variant.
- Jumlah Dimensi : Jumlah dimensi yang akan dikandung oleh array. Kebanyakan array berupa satu dimensi atau dua dimensi, tetapi Anda bisa menambahkan dimensi lagi jika Anda bekerja dengan model matematika kompleks bentuk tiga dimensi.
- Jumlah Elemen : Jumlah elemen yang akan dikandung oleh array. Elemen pada array berhubungan dengan indeks array. Secara *default*, index array pertama adalah 0 (nol), sama seperti array kontrol.

F. Mendeklarasikan Array Berukuran Tetap

Sintaks dasar untuk Array berukuran tetap adalah sebagai berikut:

```
PublicArrayName(Dim1Elements,Dim2Elements,...)AsTypeData
```

Keterangan:

Public adalah perintah yang menciptakan array global.

- *ArrayName* adalah nama variabel array.
- *Dim1Elements* adalah jumlah elemen pada dimensi pertama array

- *Dim2Elements* adalah jumlah elemen pada dimensi kedua array (dimensi tambahan bisa disertakan)
- *TypeData* adalah perintah yang berhubungan dengan tipe data yang akan disertakan dalam array.

Sebagai contoh, untuk mendeklarasikan sebuah array publik string satu dimensi bernama Karyawan yang memiliki ruang untuk 10 nama pegawai, Anda bisa menyetikkan pernyataan berikut ini pada modul standar:

Public Employee(9) As String

Apabila Anda membuat Array, Visual Basic akan menyediakan ruang untuk array tersebut dalam memori. Gambar di bawah ini menunjukkan bagaimana array diorganisir. Kesepuluh elemen array diberi nomor 0 sampai 9, bukan dari 1 sampai 10, karena indeks array dimulai dari 0 kecuali jika Anda menggunakan pernyataan Option Base.

	Employee
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Untuk mendeklarasikan sebuah array publik dua dimensi bernama ScoreBoard yang memiliki ruang 2 baris dan 9 kolom data bertipe variant, Anda harus menyetikkan pernyataan berikut ini pada modul standar.

Public ScoreBoard(1, 8) As Variant

Apabila Anda mendeklarasikan array dua dimensi, Visual Basic menyediakan ruang dalam memori. Anda lalu bisa menggunakan array ini pada program sama seperti tabel nilai seperti ditunjukkan pada gambar dibawah ini

Papan Nilai

		Kolom								
		0	1	2	3	4	5	6	7	8
Baris	0									
	1									

G. Bekerja Dengan Elemen Array

Setelah Anda mendeklarasikan array menggunakan perintah Static atau Public, Anda siap menggunakan array tersebut pada program. Untuk merujuk elemen dari suatu array, Anda harus menggunakan nama array dan indeks array dalam tanda kutip. Indeks tersebut harus berupa nilai integer. Sebagai contoh, indeks bisa berupa angka sederhana atau variabel integer. Pernyataan berikut ini akan menugaskan nilai "Leslie" ke elemen 5 dari array Employees:

Employees(1) = "John"

Employees(5) = "Leslie"

Ini menghasilkan hasil seperti gambar di bawah ini pada array Karyawan.

	Karyawan
0	
1	John
2	
3	
4	
5	Leslie
6	
7	
8	
9	

Begitu juga pernyataan berikut ini akan menugaskan angka 4 ke dalam bari 0, kolom 2 pada contoh array ScoreBoard;

ScoreBoard(0, 2) = 4

ScoreBoard(1, 2) = 6

ScoreBoard(0, 6) = 3

ScoreBoard(1, 6) = 7

Ini akan menghasilkan seperti gambar berikut ini

Papan Nilai

Kolom

		0	1	2	3	4	5	6	7	8
Baris	0			4				3		
	1			6				7		

Anda bisa menggunakan teknik pengindeksan ini untuk menugaskan atau mengambil setiap elemen array.

Contoh program membuat array berukuran tetap untuk menyimpan temperatur. Program ini menggunakan sebuah array publik satu dimensi bernama Temperatur untuk mencatat temperatur selama seminggu. Program ini mendemonstrasikan bagaimana Anda bisa menggunakan array untuk menyimpan dan memproses sekumpulan nilai yang saling berhubungan dalam program. Temperatur disimpan dalam array menggunakan fungsi *InputBox* dan loop *For...Next*. Counter loop digunakan untuk merujuk setiap elemen array. Isi array lalu ditampilkan pada *form* menggunakan loop *For...Next* dan metode *Print*, dan temperatur rata-rata dihitung dan ditampilkan.

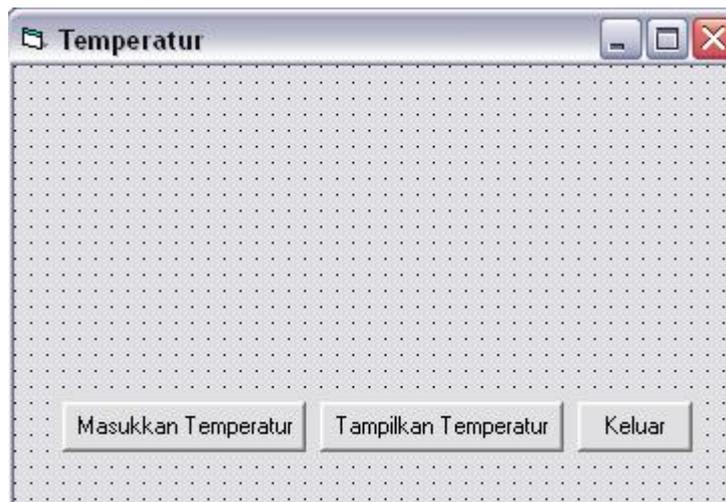
Langkah-langkahnya adalah sebagai berikut;

1. Pada menu *file*, klik perintah *New Project*, lalu klik *OK*.
2. Gunakan kontrol *CommandButton* untuk membuat tiga buah tombol perintah pada bagian bawah *form*.
3. Ubahlah properti berikut ini untuk objek tombol perintah dan *form*:

Tabel Perubahan Properti

Objek	Properti	Setting
Command1	Caption	"Masukkan temperatur"
	Name	cmdMasukTemp
Command2	Caption	"Tampilkan temperatur"
	Name	cmdTampilTemp
Command3	Caption	"Keluar"
	Name	cmdKeluar
Form1	Caption	"Temperatur"
	AutoReDraw	<i>True</i>

4. *Form* tampak sebagai berikut:



5. Pada menu Project, klik perintah Add Module, lalu klik Open untuk membuat sebuah modul standar untuk deklarasi array.
6. Ketikkan pernyataan berikut ini pada modul standar:

```
Option Base 1
Public Temperatur(7) As Variant
```

Pernyataan Option Base mengubah indeks elemen pertama array dari 0 menjadi 1 untuk semua array dalam program. Pernyataan kedua membuat sebuah array publik bernama Temperatur (yang bertipe variant) yang memiliki tujuh buah elemen. Karena array ini telah dideklarasikan secara publik, ia akan dapat digunakan pada seluruh bagian program.

7. Tutuplah jendela *Code* modul standar, lalu klik ganda tombol “Masukkan Temperatur”
Event procedure cmdMasukTemp_Click muncul pada jendela *code*.
8. Ketikkan pernyataan program berikut ini untuk meminta pemakai memasukkan temperatur dan me-load *input* ke dalam array:

```
Cls
Prompt$ = "Masukkan derajat temperatur"
For i% = 1 To 7
    Title$ = "Hari" & i%
    Temperatur(i%) = InputBox(Prompt$, Title$)
Next i%
```

Loop For...Next menggunakan variabel counter *i%* sebagai indeks array untuk me-load temperatur ke dalam elemen array 1 sampai 7. *Input* diterima dari fungsi *InputBox*, yang menggunakan variabel *Prompt\$* dan *Title\$* sebagai argumennya.

9. Bukalah kotak daftar Object pada jendela *Code*, lalu klik objek cmdTampilTemp. Ketikkan pernyataan berikut ini pada event procedure cmdTampilTemp_Click:

```
Print "Derajat Temperatur minggu ini:"
```

```

Print
For i% = 1 To 7
    Print "Hari "; i%, Temperatur(i%)
    Total! = Total! + Temperatur(i%)
Next i%
Print
Print "Rata-rata derajat temperatur: "; Total! / 7

```

Event procedure ini menggunakan metode print untuk menampilkan informasi yang disimpan dalam array Temperatur pada *form*. Ia menggunakan loop For...Next untuk memutar elemen dalam array, dan menghitung total dari semua temperatur menggunakan pernyataan:

```
Total! = Total! + Temperatur(i%)
```

Baris terakhir pada event procedure menampilkan temperatur rata-rata dari seminggu, hasil dari membagi total temperatur dengan jumlah hari.

10. Bukalah kotak daftar Objek pada jendela *Code*, lalu klik objek cmdKeluar. Ketikkan pernyataan berikut ini pada event procedure cmdKeluar_Click:

```
End
```

11. Jalankan proyek dengan menekan tombol Start.
12. Klik tombol Masukkan Temperatur, lalu ketikkan tujuh temperatur yang berbeda sewaktu diminta oleh fungsi *InputBox*.

Kotak dialog fungsi *InputBox* tampak seperti gambar di bawah ini:



13. Setelah Anda mengetikkan temperatur, klik tombol Tampilkan Temperatur.

Visual Basic menggunakan metode Print untuk menampilkan setiap temperatur pada *form* dan mencetak rata-rata di bagian bawah. Layar Anda akan tampak seperti gambar berikut ini:



14. Klik tombol Keluar untuk keluar dari program.

H. Membuat Array Dinamis

Seperti terlihat pada contoh di atas, array sangat bermanfaat untuk memanipulasi serangkaian angka, terutama jika Anda memprosesnya menggunakan loop For...Next. Tetapi bagaimana jika Anda tidak yakin seberapa besar array yang akan Anda gunakan sebelum menjalankan program?

Visual Basic menangani masalah ini dengan kontainer yang fleksibel bernama array dinamis. Array dinamis berubah ukurannya pada saat program berjalan, baik sewaktu pemakai menentukan ukuran array berdasarkan kondisi tertentu. Mengubah ukuran array dinamis membutuhkan beberapa langkah karena walaupun ukuran array tidak ditentukan sampai program berjalan. Anda perlu menyediakan tempat untuk array pada saat desain. Untuk membuat array dinamis ikuti langkah-langkah berikut ini:

- Tentukan nama dan tipe dari array pada saat desain, abaikan jumlah elemen dalam array. Sebagai contoh, untuk membuat sebuah array dinamis publik bernama Temperatur, ketikkanlah:

```
Public Temperatur() As Variant
```

- Tambahkan kode program untuk menentukan jumlah elemen yang harus berada dalam array pada saat program berjalan. Anda bisa meminta masukan dari pemakai menggunakan fungsi *InputBox*, atau Anda bisa menghitung ruang yang dibutuhkan oleh program menggunakan properti atau logika lain. Sebagai contoh, pernyataan berikut ini mendapatkan ukuran array dari pemakai dan menugaskannya ke dalam variabel Hari:

```
Hari = InputBox("Berapa hari?", "Membuat Array")
```

- Gunakanlah variabel pada pernyataan ReDim untuk mengubah dimensi array. Sebagai contoh, pernyataan berikut ini mengubah ukuran array Temperatur pada saat program berjalan menggunakan variabel Hari:

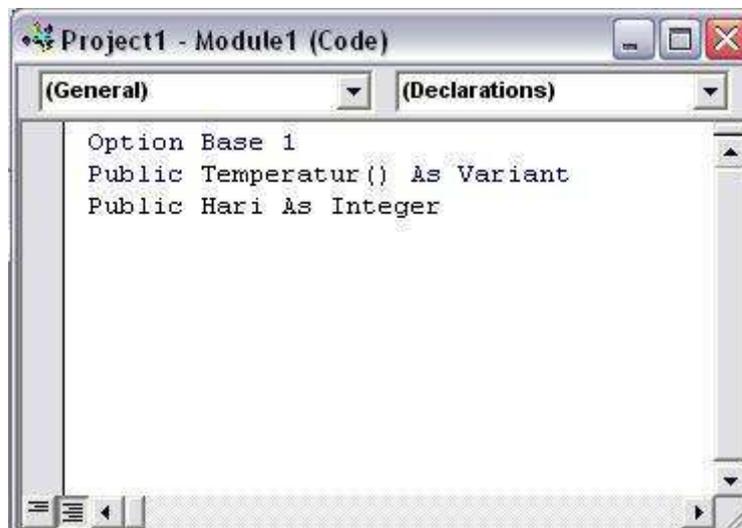
```
ReDim Temperatur(Hari)
```

- o Gunakanlah angka sebagai batas teratas pada loop For...Next untuk memproses elemen array jika perlu. Sebagai contoh, loop For...Next berikut ini menggunakan variabel Hari sebagai batas teratas dari loop.

```
For i% = 1 To Hari
    Temperatur (i%) = InputBox(Prompt$, Title$)
Next i%
```

Contoh Program Array Dinamis

1. Buatlah program baru seperti pada program Temperatur sebelumnya.
2. Klik Modul1 pada jendela Project, lalu klik tombol View Code untuk membukanya pada jendela Code.
3. Ketikkan kode program berikut, sehingga terlihat pada gambar berikut:



4. Public Temperatur() As Variant

Berbeda dengan contoh sebelumnya dimana pada deklarasi array diisikan angka 7 yang menyatakan banyaknya array yang dibuat. Penghapusan angka 7 ini membuat array menjadi array dinamis.

```
Public Hari As Integer
```

Variabel integer publik Hari akan digunakan untuk menerima *input* dari pemakai dan mengubah ukuran array dinamis pada saat program berjalan.

5. Tutuplah modul standar. Tutuplah Klik Form1 pada jendela Project, klik tombol View Object, lalu klik ganda tombol Masukkan Temperatur. Ubahlah event procedure cmdMasukTemp_Click agar tampak seperti kode berikut:

```
Cls
Hari = InputBox("Berapa hari?", "Buat Array")
If Hari > 0 Then ReDim Temperatur(Hari)
Prompt$ = "Masukkan derajat temperatur"
For i% = 1 To Hari
```

```
Title$ = "Hari" & i%
Temperatur(i%) = InputBox(Prompt$, Title$)
Next i%
```

Baris kedua dan ketiga meminta jumlah temperatur yang akan disimpan oleh pemakai, lalu menggunakan *input* untuk mengubah ukuran array dinamis. Pernyataan *If...Then* digunakan untuk memastikan bahwa jumlah hari lebih besar dari 0. Variabel *Hari* juga digunakan sebagai batas teratas untuk loop *For...Next*.

- 6. Bukalah kotak daftar Object, lalu klik objek *cmdTampilTemp*. Ubahlah event procedure *cmdTampilHasil_Click* agar tampak seperti kode di bawah ini:

```
Print "Derajat Temperatur:"
Print
For i% = 1 To Hari
    Print "Hari "; i%, Temperatur(i%)
    Total! = Total! + Temperatur(i%)
Next i%
Print
Print "Rata-rata derajat temperatur: "; Total! / Hari
```

- 7. Klik tombol *Start* untuk menjalankan proyek
- 8. Klik tombol *Masukkan Temperatur*, lalu ketikkan 5 apabila Anda diminta memasukkan jumlah hari yang ingin dicatat.
- 9. Masukkan kelima temperatur seperti yang diminta.
- 10. Apabila sudah selesai memasukkan temperatur, klik tombol *Tampilkan Temperatur*. Program menampilkan kelima temperatur pada *form* beserta rata-ratanya. Layar yang dihasilkan seperti berikut:



- 11. Klik tombol *End* pada toolbar untuk mengakhiri program.

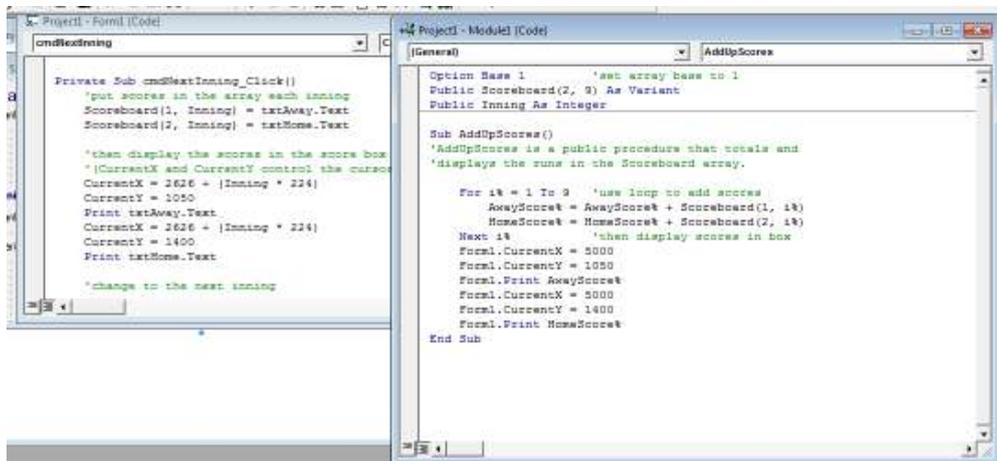
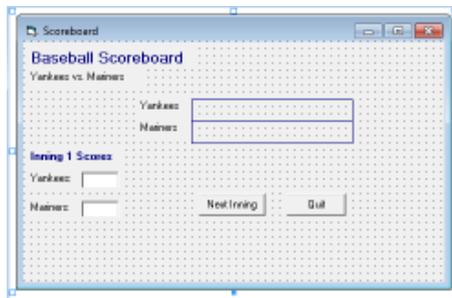
I. Menggunakan Array Multidimensiabel informasi.

Selain array satu dimensi yang menangani daftar informasi, Visual Basic mengizinkan Anda menciptakan array multidimensi yang menangani satu atau lebih tabel informasi. Bekerja dengan array multidimensi bisa menyulitkan sampai Anda dapat menguasainya.

Pada praktek berikut ini, Anda akan menggunakan sebuah array dua dimensi bernama ScoreBoard untuk mencatat nilai pada pertandingan baseball antara Seattle Mariners dan New York Yankees. Array yang akan Anda gunakan adalah array 2 kali 9 mirip dengan array yang Anda buat sebelumnya

Membuat papan nilai baseball dengan array dua dimensi

1. Klik tombol Open Project pada toolbar
2. Buatlah form ScoreBoard berikut



```
Project - Modul1 (Code)
General (Declarations)

Option Base 1 'set array base to 1
Public Scoreboard(1, 9) As Variant
Public Inning As Integer

Sub AddUpScores()
'AddUpScores is a public procedure that totals and
'displays the runs in the Scoreboard array.

  For i = 1 To 9 'use loop to add scores
    AwayScore = AwayScore + Scoreboard(1, i)
    HomeScore = HomeScore + Scoreboard(2, i)
  Next i 'then display scores in box
  Form1.CurrentX = 5000
  Form1.CurrentY = 1050
  Form1.Print AwayScore
  Form1.CurrentX = 5000
  Form1.CurrentY = 1400
  Form1.Print HomeScore
End Sub
```

3. Jalankan program.
4. Ketikkan 0 lalu tekan Tab, amati apa yang terjadi
5. Ketikkan 0, lalu klik tombol Next Inning, amati apa yang terjadi
6. Teruskan mengisi nilai pertandingan setiap Inning sampai Anda selesai sampai Inning ke-9