
MODUL II

VARIABEL DAN TIPE DATA

A. TUJUAN

- ✓ Mengetahui tipe data, jenis-jenis, dan penggunaannya.
- ✓ Memahami variabel dan cara pendeklarasiannya.
- ✓ Memahami konvensi penamaan dan *coding standard*.
- ✓ Memahami jenis konversi tipe data dan implementasinya.

B. PETUNJUK

- Awali setiap aktivitas dengan doa, semoga berkah dan mendapat kemudahan.
- Pahami tujuan, dasar teori, dan latihan-latihan praktikum dengan baik.
- Kerjakan tugas-tugas praktikum dengan baik, sabar, dan jujur.
- Tanyakan kepada asisten/dosen apabila ada hal-hal yang kurang jelas.

C. DASAR TEORI

○ Tipe Data

Tipe merupakan pola representasi suatu data di komputer dan berfungsi untuk mendefinisikan objek yang akan diprogram. Pada prinsipnya, tipe menentukan representasi internal data atau suatu nilai. Tipe data digunakan untuk menyimpan dan merepresentasikan data di dalam aplikasi.

Bahasa-bahasa pemrograman di .NET merupakan *strongly typed*, yang berarti bahwa setiap variabel dan ekspresi memiliki tipe yang harus diketahui pada saat kompilasi. Tipe data di .NET diklasifikasikan ke dalam dua bagian: tipe *value* dan tipe *reference*. Tipe-tipe primitif (kecuali string), enumerasi, dan struktur merupakan tipe value. Adapun kelas, string, modul standar, interface, array, dan delegate merupakan tipe reference.

Tipe primitif Visual Basic .NET terdiri dari integer (Byte, Short, Integer, Long), floating-point (Single, Double, Decimal), Boolean, dan Char. Tipe-tipe ini diidentifikasi melalui keyword, yang merupakan alias untuk tipe predefined di namespace `System`.

- **Variabel**

Variabel adalah suatu lokasi di memori, di mana interpretasinya bergantung pada kelas penyimpanan—yang ditentukan berdasarkan keyword dan konteks deklarasinya—dan tipenya. Pada dasarnya, setiap variabel memiliki nama (yang dikenal sebagai *identifier*), tipe, dan ruang lingkup (*scope*) atau visibilitas (*visibility*).

- **Coding Standard**

Dalam kaidah pemrograman, suatu kode program harus mudah dibaca dan dipelihara. Perlu diketahui, kemudahan suatu kode untuk dibaca bukan hanya oleh pembuatnya sendiri, tetapi juga bagi orang lain. Berikut ini adalah aturan-aturan dasar yang bisa membantu untuk menghasilkan kode program yang standar.

- **Notasi Penamaan**

Dalam penamaan objek atau variabel, disarankan menggunakan notasi yang telah ditetapkan.

Notasi Pascal: Notasi ini terdiri dari huruf besar di setiap awal kata, contohnya `MyVar`. Notasi Pascal disarankan untuk digunakan pada penamaan class, method, properti, event, enumerasi, field public, dan namespace. Khusus untuk penamaan interface, sebaiknya diawali huruf I, contohnya `INamaInterface`.

Notasi Camel: Notasi ini terdiri dari huruf kecil di awal kata dan diikuti huruf besar untuk kata berikutnya, contoh `myVar`, `getDataConn`, dan sebagainya. Notasi ini digunakan untuk parameter yang dilewatkan ke method, variabel-variabel lokal, dan anggota-anggota class dengan access modifier `Private` atau `Protected`.

Notasi Upper: Notasi ini terdiri dari huruf besar semua dan dipisahkan karakter garis bawah apabila ada beberapa kata, contohnya `MAX`, `MAX_DATA`, dan sebagainya. Notasi ini sering digunakan untuk penamaan konstanta.

- **Standar Penamaan**

Mengacu pada standar pembuatan program VB.NET, berikut ini adalah daftar standar penamaan yang disarankan. Diharapkan, dengan mengikuti standar yang ada, maka kode program akan lebih mudah dibaca serta dipahami—baik oleh pemrogram sendiri maupun orang lain.

Tipe Data	Awalan/Prefiks	Contoh
Boolean	b, bool, atau bln	bFound
Byte	byt	bytData
Char	chr	chrData
Date	dt	dtMax

Decimal	dec	decVal
Double	dbl	dblVal
Integer	int	intMax
Long	lng	lngVal
Object	o	oData
Short	srt	srtVal
Single	sng	sngVal
String	str	strVal

○ **Konversi**

Keberagaman tipe data di bahasa pemrograman berpotensi memicu terjadinya konversi data. Konversi bisa diartikan sebagai upaya pengubahan suatu entitas dari satu tipe data ke tipe data lainnya. Konversi ini bisa berupa *widening* atau *narrowing*.

Konversi *widening* adalah konversi dari satu tipe ke tipe lainnya yang dijamin dapat dilakukan—misalnya konversi dari tipe turunan ke tipe dasar. Di sisi lain, konversi *narrowing* tidak bisa dipastikan akan selalu berhasil dan berpotensi menimbulkan hilangnya informasi.

D. LATIHAN

a. **Deklarasi Variabel**

Deklarasi variabel dilakukan untuk menyimpan suatu penempatan di memori, memberinya nama, dan mendefinisikan tipe data yang dapat digunakan di variabel tersebut. Dengan demikian, deklarasi minimal mendefinisikan tipe data dan nama (*identifier*).

Pendeklarasian variabel global (field) sebaiknya menggunakan access modifier (Private, Protected, Public). Adapun untuk variabel lokal, gunakan keyword Dim (Dimension).

```
Module Module1
    ' Deklarasi field (variabel global) dengan tipe String
    Private strName As String

    Sub Main()
        ' Deklarasi variabel lokal
        Dim intVal As Integer

        Console.WriteLine(intVal)
        ' Output: 0

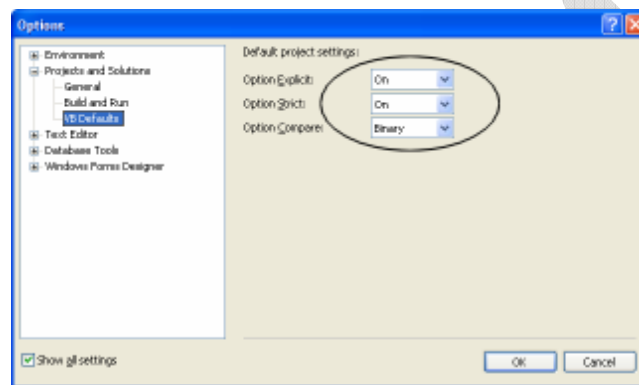
    End Sub
End Module
```

VB.NET juga mengizinkan deklarasi dan inisialisasi dilakukan secara bersamaan dalam satu baris.

```
' Deklarasi dan inisialisasi  
Private intMax As Integer = 9
```

b. Pernyataan Option Strict

Pada saat menulis kode program, sebaiknya tetapkan opsi kompilasi dengan menggunakan pernyataan `Option Explicit On`. Pernyataan ini dimaksudkan untuk menginstruksikan kompiler Visual Basic agar tidak mendukung deklarasi variabel secara implisit. Pernyataan ini bisa ditetapkan melalui menu **Tools > Options**.



Gambar 1 Mengatur opsi project

Pernyataan `Option Strict` menyatakan apakah operasi-operasi dan konversi-konversi objek diatur secara tegas (*strict*) atau tidak. Apabila `Option Strict On`, maka operasi komparasi, kombinasi, atau penambahan variabel mensyaratkan dilakukan dari tipe yang sama.

Pernyataan `Option` juga dapat didefinisikan secara manual di baris paling atas dalam kode program.

```
Option Explicit On  
Option Strict On  
  
Module Module1  
    Sub Main()  
    End Sub  
End Module
```

c. Konversi Tipe

Konversi tipe data dapat dilakukan dalam dua cara: **implisit**, di mana konversi dilakukan secara otomatis, dan **eksplisit**, di mana diperlukan sintaks khusus.

Contoh konversi implisit:

```
Dim intValue As Integer = 123
' Konversi implisit (konversi widening)
' dari Integer ke Long
Dim longValue As Long = intValue
Console.WriteLine(intValue & " = " & longValue)
' Output: 123
```

Contoh konversi eksplisit:

```
' Konversi narrowing (Long ke Integer)
Dim intValue2 As Integer = longValue
' Error

' Harus menggunakan method konversi
Dim intValue3 As Integer = CInt(longValue)
```

Sebagai alternatif, juga disediakan method `CType()` untuk konversi tipe data.

```
' Menggunakan method CType(ekspresi, tipe target)
Dim intValue4 As Integer = CType(longValue, Integer)
```

Method konversi lainnya antara lain: `CBool`, `CByte`, `CDate`, `Cdbl`, `CDec`, `CLng`, `CSng`, `CShort`, `CStr`, `Fix`, dan `Int`.

Perhatikan, perilaku konversi sangat dipengaruhi oleh pernyataan `Option Strict`.

Pada saat melakukan konversi tipe value ke reference, Visual Basic akan melakukan boxing (konversi implisit), misalnya konversi `Integer` ke `Object`.

E. TUGAS PRAKTIKUM

1. Konversi nilai variabel berikut ke tipe `Short` dan `Double`.

```
Dim intVal As Integer = 33
```

Jelaskan jenis konversinya?

2. Konversi nilai variabel berikut ke tipe `Integer`.

```
Dim str As String = "123"
```

Jelaskan jenis konversinya?

3. Konversi nilai variabel berikut ke `Object`.

```
Dim intVal As Integer = 33
```

Jelaskan jenis konversinya?