



# VARIABEL, KONSTANTA, TIPE DATA DAN OPERATOR

- ❑ Fitri Ayuning Tyas, S. Kom
- ❑ yas.0373@gmail.com
- ❑ 081804767700
- ❑ tyas-tamimy.com

# PENDAHULUAN



- ❑ Sebuah program tersusun dari kumpulan objek yang saling berkomunikasi satu dengan yang lain
- ❑ Data yang terbentuk dari jenis data struktur bertujuan untuk menyimpan satu nilai saja dan sering disebut dengan istilah **tipe data** sederhana
- ❑ Kemampuan objek dalam menyimpan data selama program berjalan dibedakan menjadi dua jenis yaitu objek yang dapat diubah nilainya selama program berjalan yang disebut dengan **variabel** dan objek yang tidak dapat diubah nilainya selama program berjalan yang disebut dengan istilah **konstanta**

# VARIABEL



- ❑ **Variabel** adalah salah satu pengenal (identifier) yang digunakan untuk menampung suatu nilai yang diperlukan di dalam program
- ❑ Dalam program, nilai akan disimpan dalam memori komputer pada alamat memori tertentu.
- ❑ Untuk mengakses ke nilai tersebut, dibutuhkan suatu nama yang direlasikan dengan nilai bersangkutan. Nama inilah yang disebut **variabel**.
- ❑ Nilai variabel dapat **diubah** sesuai kebutuhan sesuai kebutuhan
- ❑ Nama variabel harus bersifat **unik** di dalam suatu blok program

# Deklarasi Variabel dengan Tipe Data



- ❑ Dalam Visual Basic, variabel dideklarasikan menggunakan kata kunci `Dim` dan `As`

- ❑ Bentuk Umum:

```
Dim NamaVariabel As TipeData
```

- ❑ Contoh: 

```
'mendeklarasikan variabel a dengan tipe data Integer  
Dim a As Integer
```

```
'mendeklarasikan variabel b dengan tipe data Double  
Dim b As Double
```

```
'mendeklarasikan variabel nama dengan tipe data String  
Dim nama As String
```

```
'mendeklarasikan beberapa variabel dengan tipe data yang sama  
'tuliskan dalam satu baris kode  
'pisahkan setiap variabel dengan tanda koma  
Dim x, y, z As Integer
```

# Deklarasi Variabel



- ❑ Setiap nilai di dalam program akan disimpan di dalam alamat memori yang terpisah.
- ❑ Contoh, jika ada dua variabel  $a$  dan  $b$ , maka akan ada dua alamat memori yang dialokasikan untuk kebutuhan program.
- ❑ Gunakan operator sama dengan ( $=$ ) untuk menyimpan atau mengisi nilai ke dalam suatu variabel yang sudah dideklarasikan sebelumnya.
- ❑ Contoh:

```
'mengisi nilai ke dalam variabel  
a = 37  
b = 37.3  
nama = "Tyas Tamimy"
```

# Contoh Program Pendeklarasian dengan Tipe Data



Module Program\_Variabel

```
Sub Main()
```

```
Dim panjang As Integer  
Dim lebar As Integer  
Dim luas As Integer
```

Deklarasi Variabel

```
panjang = 7  
lebar = 3
```

Mengisi nilai ke dalam variabel

```
luas = panjang * lebar
```

Perhitungan dan menampung hasil  
ke dalam variabel luas

```
Console.WriteLine("PERSEGI PANJANG")  
Console.WriteLine("Panjang {0} = {1}", Chr(9), panjang)  
Console.WriteLine("Lebar {0} = {1}", Chr(9), lebar)  
Console.WriteLine("Luas {0} = {1}", Chr(9), luas)  
Console.ReadLine()
```

```
End Sub
```

```
End Module
```

# Deklarasi Variabel tanpa Tipe Data



- ❑ Visual Basic .NET mengizinkan deklarasi variabel tanpa tipe data. Tipe data dari variabel tersebut akan ditentukan otomatis oleh kompilator, tergantung nilai yang dimasukkan ke dalamnya.

- ❑ Contoh: 

```
Dim a = 7
Dim b = 3.73
Dim c = "T"c
Dim d = 3.73F
Dim e = "Tyas Tamimy"
```

atau

```
Dim a = 7, b = 3.73, c = "T"c, d = 3.73F, e = "Tyas Tamimy"
```



# Contoh Program

## Pendeklarasian tanpa Tipe Data

Module VariabelTanpaTipe

### □ Hasil

```
Sub Main()  
  'mendeklarasikan variabel tanpa tipe data  
  Dim a = 7  
  Dim b = 3.73  
  Dim c = "T"c  
  Dim d = 3.73F  
  Dim e = "Tyas Tamimy"  
  
  'variabel f belum diinisialisasi  
  Dim f  
  
  'mengisi nilai ke dalam variabel f  
  f = 373  
  
  'menampilkan nilai  
  Console.WriteLine("Nilai variabel:")  
  Console.WriteLine("Nilai a = {0}", a)  
  Console.WriteLine("Nilai b = {0}", b)  
  Console.WriteLine("Nilai c = {0}", c)  
  Console.WriteLine("Nilai d = {0}", d)  
  Console.WriteLine("Nilai e = {0}", e)  
  Console.WriteLine("Nilai f = {0}", e)  
  Console.ReadLine()  
  
End Sub  
End Module
```

```
Nilai variabel:  
Nilai a = 7  
Nilai b = 3,73  
Nilai c = T  
Nilai d = 3,73  
Nilai e = Tyas Tamimy  
Nilai f = Tyas Tamimy
```

Sisipkan kode pada slide berikut,  
gunakan metode GetType() —> untuk mengambil tipe data variabel tertentu



# Contoh Program

## Pendeklarasian tanpa Tipe Data

```
'menampilkan nilai
Console.WriteLine("Nilai variabel:")
Console.WriteLine("Nilai a = {0}", a)
Console.WriteLine("Nilai b = {0}", b)
Console.WriteLine("Nilai c = {0}", c)
Console.WriteLine("Nilai d = {0}", d)
Console.WriteLine("Nilai e = {0}", e)
Console.WriteLine("Nilai f = {0}", e)
Console.WriteLine()

'menampilkan tipe data menggunakan metode GetType()
Console.WriteLine("Tipe Variabel: ")
Console.WriteLine("Nilai a = {0}", a.GetType())
Console.WriteLine("Nilai b = {0}", b.GetType())
Console.WriteLine("Nilai c = {0}", c.GetType())
Console.WriteLine("Nilai d = {0}", d.GetType())
Console.WriteLine("Nilai e = {0}", e.GetType())
Console.WriteLine("Nilai f = {0}", e.GetType())

Console.ReadLine()
```

### □ Hasil

```
Nilai variabel:
Nilai a = 7
Nilai b = 3,73
Nilai c = I
Nilai d = 3,73
Nilai e = Iyas Tamimy
Nilai f = Iyas Tamimy

Tipe Variabel:
Nilai a = System.Int32
Nilai b = System.Double
Nilai c = System.Char
Nilai d = System.Single
Nilai e = System.String
Nilai f = System.String
```

End Sub

End Module

# Contoh Program Membaca Data dari *Keyboard*



Module BacaDariKeyboard

```
Sub Main()  
  Dim nama As String  
  
  Console.Write("Masukan nama Anda: ")  
  'membaca data dari keyboard  
  nama = Console.ReadLine()  
  
  'menampilkan data  
  Console.WriteLine("Hallo {0}, apa kabar?", nama)  
  
  Console.ReadLine()  
End Sub  
  
End Module
```

- Hasil

```
Masukan nama Anda: Tyas Tamimy  
Hallo Tyas Tamimy, apa kabar?
```

# Konversi Tipe Data



- ❑ Pada Visual Basic .NET, proses pembacaan data dari layar console akan selalu dianggap bertipe teks (*string*)
- ❑ Untuk mengkonversi ke data bertipe numerik gunakan metode `TryParse()` atau `Parse()` atau kelas `Convert`

# Contoh Program Membaca Data dari *Keyboard* dan penggunaan

## TryParse ()

Module KonversiString

```
Sub Main()  
    Dim x As String  
    Dim y As Char  
  
    Console.Write("Masukkan karakter: ")  
    x = Console.ReadLine()  
  
    'konversi dari Tipe data string ke Char  
    If (Char.TryParse(x, y)) Then  
        Console.WriteLine("Anda memasukan karakter {0}", y)  
    Else  
        Console.WriteLine("Konversi ke Tipe Char gagal")  
    End If  
  
    Console.ReadLine()  
  
End Sub  
  
End Module
```

## □ Hasil

```
Masukkan karakter: I  
Anda memasukan karakter I
```

```
Masukkan karakter: Tyas  
Konversi ke Tipe Char gagal
```



# KONSTANTA



- ❑ Konstanta adalah suatu pengenal yang berguna untuk menyimpan tetapan-tetapan (nilai konstan) yang dibutuhkan oleh program, seperti nilai Phi, kecepatan cahaya, dan sebagainya
- ❑ Konstanta mirip dengan variabel, bedanya nilai konstanta tidak dapat diubah

❑ Bentuk Umum: `Const NamaKonstanta As TipeData = Nilai`

❑ Contoh:

```
'pendefinisian konstanta
Const Phi As Double = 3.14
Const NAMASOFTWARE As String = "Visual Basic .NET"
```

- ❑ Dalam Visual Basic .NET, konstanta secara otomatis bersifat Shared, Static, dan ReadOnly sehingga kita tidak dapat mendeklarasikan konstanta menggunakan salah satu kata kunci tersebut.

# Contoh Program dengan Penerapan Konstanta



Module PendefinisianKonstanta

```
Sub Main()
```

```
'pendefinisian konstanta
```

```
Const Phi As Double = 3.14
```

```
Dim x As String
```

```
Dim r, luas As Double
```

```
'membaca data bilangan bulat dari keyboard
```

```
Console.WriteLine("Masukkan jari-jari lingkaran: ")
```

```
x = Console.ReadLine()
```

```
'konversi dari tipe string ke tipe double menggunakan kelas convert
```

```
r = Convert.ToDouble(x)
```

```
'penggunaan konstanta Phi
```

```
luas = Phi * r * r
```

```
Console.WriteLine("Luas Lingkaran = {0}", luas)
```

```
Console.ReadLine()
```

```
End Sub
```

```
End Module
```

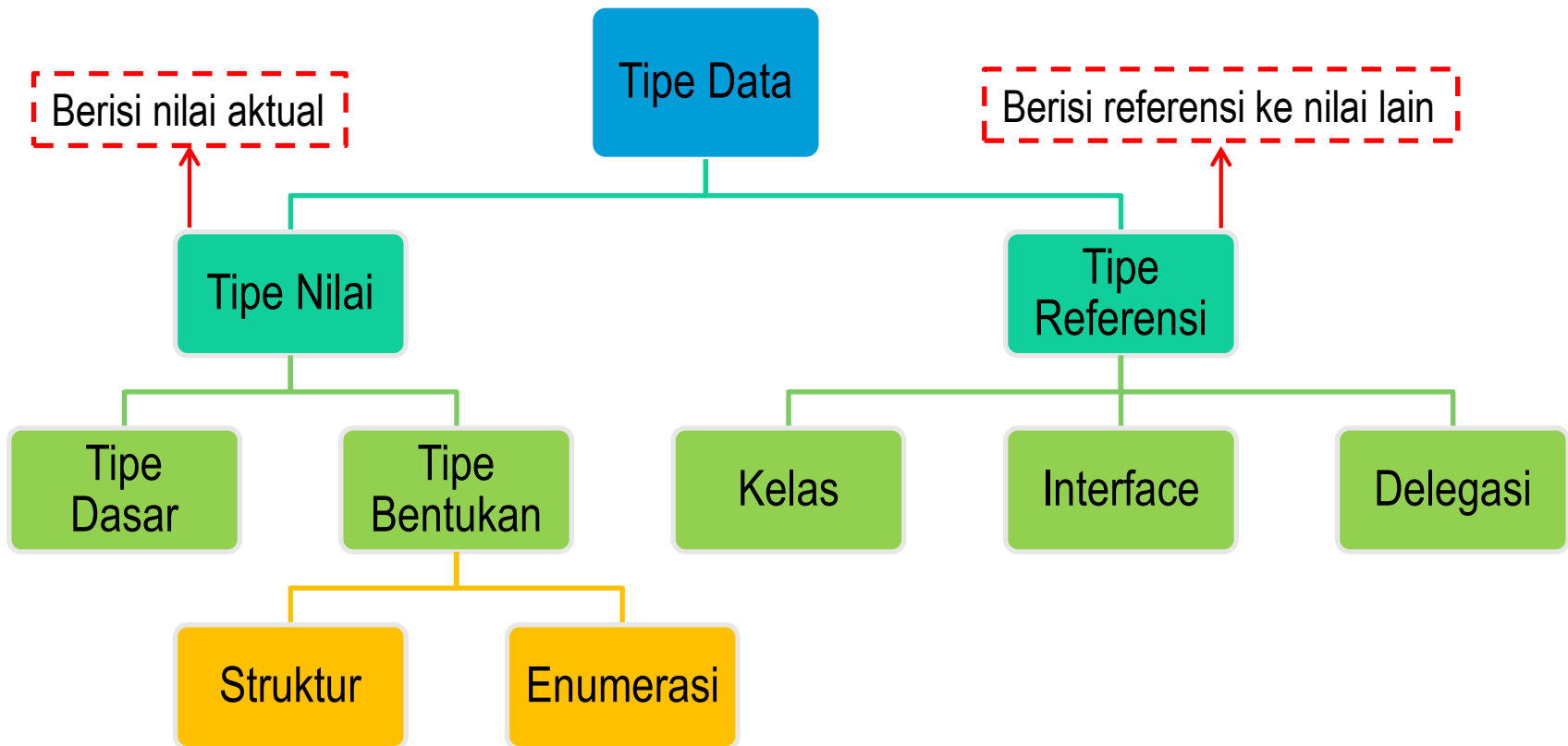
□ Hasil

```
Masukkan jari-jari lingkaran: 5  
Luas Lingkaran = 78,5
```

# TIPE DATA



## Klasifikasi Tipe Data dalam Visual Basic .NET



# Tipe Data



- ❑ **Common Type System (CTS)** adalah Tipe data yang didefinisikan di dalam .NET Framework. Tipe data yang didefinisikan di dalam Visual Basic .NET sebenarnya merupakan alias atau nama lain dari tipe .NET
- ❑ **Tipe Dasar** adalah jenis data yang sudah didefinisikan oleh kompilator Visual Basic .NET, tergolong tipe nilai karena bertipe struktur, bukan kelas.

# Daftar Tipe Dasar dalam Visual Basic .NET



Tipe	Tipe .NET (CTS)	Keterangan
<b>Boolean</b>	System.Boolean	Tipe logika (True atau False)
<b>Char</b>	System.Char	Tipe karakter
<b>Date</b>	System.DateTime	Tipe tanggal dan waktu
<b>Decimal</b>	System.Decimal	Tipe numeric untuk uang ( $1.0 \times 10^{-28}$ s.d. $7.9 \times 10^{28}$ )
<b>Double</b>	System.Double	Bilangan riil (64-bit) ( $5.0 \times 10^{-324}$ s.d. $1.7 \times 10^{308}$ )
<b>Single</b>	System.Single	Bilangan riil (32-bit) ( $1.5 \times 10^{-45}$ s.d. $3.4 \times 10^{38}$ )
<b>Byte</b>	System.Byte	Bilangan bulat tak bertanda (8-bit) (0 s.d. 255)
<b>Sbyte</b>	System.SByte	Bilangan bulat bertanda (8-bit) (-128 s.d. 127)

# Daftar Tipe Dasar dalam Visual Basic .NET



Tipe	Tipe .NET (CTS)	Keterangan
<b>Integer</b>	System.Int32	Bilangan Bulat (32-bit) (-2.147.483.648 s.d. 2.147.483.647)
<b>Long</b>	System.Int64	Bilangan bulat dengan rentang panjang (64-bit) (-9.223.372.036.854.775.808 s.d. 9.223.372.036.854.775.807)
<b>Short</b>	System.Int16	Bilangan bulat dengan rentang pendek (16-bit) (-32.768 s.d. 32.767)
<b>UInteger</b>	System.UInt32	Tipe Integer tak bertanda (32-bit) (0 s.d. 4.294.967.295)
<b>ULong</b>	System.UInt64	Tipe Long tak bertanda (64-bit) (0 s.d. 18.446.744.073.709.551.615)
<b>UShort</b>	System.Int16	Tipe Short tak bertanda (16-bit) (0 s.d. 65.535)

# Penulisan Tipe Data



- Penambahan akhiran pada tipe data:
  - untuk tipe `Short` adalah `s` atau `S`
  - untuk tipe `UShort` adalah `us` atau `US`

```
Dim x As Short = 88S  
Dim y As UShort = 88US
```

- untuk tipe `Long` adalah `l` atau `L`
- untuk tipe `Ulong` adalah `ul` atau `UL`
- untuk tipe `UInteger` adalah `ui` atau `UI`

```
Dim a As Integer = 99  
Dim b As UInteger = 99UI  
Dim c As Long = 99L  
Dim d As Ulong = 99UL
```

# Penulisan Tipe Data



- ❑ Penambahan akhiran pada tipe data:
  - ❑ untuk tipe `Single` adalah `f` atau `F`
  - ❑ (optional) untuk tipe `Double` adalah `r` atau `R`

```
Dim m As Single = 10.25F
```

```
Dim y As Double = 37.00 'atau ditulis: 37.00R
```

- ❑ (optional) untuk tipe `Decimal` adalah `d` atau `D`

# Contoh Program dengan Penerapan Tipe Data



Module Lingkaran

□ Hasil

```
Sub Main()  
    'konstanta bertipe single  
    Const Phi As Single = 3.14152956F  
  
    'deklarasi variabel  
    Dim r As Single = 0.0F  
    Dim luas As Double = 0.0  
    Dim keliling As Double = 0.0  
    Dim s As String = Nothing 'null string  
  
    Console.WriteLine("Masukkan jari-jari: ")  
    s = Console.ReadLine  
  
    r = Convert.ToSingle(s) 'mengkonversi tipe string ke single  
    luas = Phi * r * r  
    keliling = 2 * Phi * r  
  
    Console.WriteLine("Luas lingkaran {0} = {1}", Chr(7), luas)  
    Console.WriteLine("Keliling lingkaran {0} = {1}", Chr(7), keliling)  
  
    Console.ReadLine()  
  
End Sub
```

```
Masukkan jari-jari: 7  
Luas lingkaran = 153,934948444366  
Keliling lingkaran = 43,9814138412476
```

End Module

# TUGAS 1



- ❑ Buatlah sebuah program sederhana menggunakan *Console Application* VB .NET yang menerapkan:
  1. Pendeklarasian variabel dengan tipe data, baik variabel global maupun variabel lokal.
  2. Pendeklarasian variabel **tanpa** tipe data. Tampilkan tipe data dari variabel-variabel yang dideklarasikan menggunakan metode `GetType ()`
  3. Pembacaan data dari Keyboard
  4. Konversi tipe data `String` ke tipe data lainnya
  5. Pendeklarasian Konstanta
  6. Penggunaan komentar

# KETENTUAN TUGAS 1



- ❑ Produk Tugas berupa Screenshot Kode Program dan hasilnya dalam format (.pdf) dengan nama file:

**NIM\_Nama\_TUGAS-1\_PD**

- ❑ Kirim tugas ke email [yas.0373@gmail.com](mailto:yas.0373@gmail.com) dengan Subjek:  
**[STMIK-MPB\_TUGAS-1\_PD\_NIM]**

**Batas pengumpulan tugas sehari sebelum perkuliahan berikutnya**

# OPERATOR



- ❑ Operator adalah tanda (dan kata) khusus yang digunakan untuk melakukan operasi-operasi tertentu yang dibutuhkan oleh program. Ex. Perhitungan, perbandingan nilai, manipulasi bit dsb.
- ❑ Operator-operator dalam Visual Basic .NET
  - ❑ Operator penugasan
  - ❑ Operator aritmetika
  - ❑ Operator relasional
  - ❑ Operator logika
  - ❑ dll

# Operator Penugasan



- ❑ Visual Basic .NET menggunakan operator = (sama dengan) untuk melakukan penugasan (pengisian nilai) ke dalam suatu variabel.

No	Operator	Fungsi	Ekspresi
1	=	Assignment	NamaVariabel = Nilai

# Operator Aritmetika



- Adalah Operator yang digunakan untuk melakukan fungsi aritmetika

No	Operator	Fungsi	Ekspresi
1	+	Penjumlahan	$n = 7 + 3$
2	-	Pengurangan	$n = 7 - 3$
3	*	Perkalian	$n = 7 * 3$
4	/	Pembagian	$n = 7 / 3$
5	\	Pembagian bilangan bulat tanpa memperdulikan sisa baginya	$n = 7 \setminus 3$
6	Mod	Modulus, sisa pembagian	$n = 7 \text{ Mod } 3$
7	^	Perpangkatan	$n = 7 ^ 3$

# Operator Relasional



- Adalah Operator yang digunakan untuk menyatakan relasi atau perbandingan antara dua operand. Hasil dari operasi perbandingan selalu bernilai logika: `True` atau `False`

No	Operator	Fungsi	Ekspresi
1	=	Sama dengan	<code>2 = 2</code> (True)
2	>	Lebih besar	<code>4 &gt; 6</code> (False)
3	<	Lebih kecil	<code>6 &lt; 2</code> (False)
4	>=	Lebih besar sama dengan	<code>8 &gt;= 8</code> (True)
5	<=	Lebih kecil sama dengan	<code>7 &lt;= 9</code> (True)
6	<>	Tidak sama dengan	<code>7 &lt;&gt; 9</code> (True)

# Operator Logika



- Adalah operator yang digunakan untuk membandingkan dua buah nilai logika, hasil operasinya juga berupa logika. Asumsikan a bernilai 30 dan b bernilai 70

No	Operator	Fungsi	Ekspresi
1	And	Logika AND	$(a = b) \text{ And } (a < b)$ bernilai False
2	Or	Logika OR	$(a <> b) \text{ Or } (a > b)$ bernilai True
3	Xor	Logika XOR	$(a = b) \text{ Xor } (a < b)$ bernilai False
4	Not	Logika NOT	Not $(a > b)$ bernilai True

# Logika AND



- ❑ Operasi logika AND akan bernilai `True` hanya jika kedua operand bernilai `True`. Apabila ada salah satu operand bernilai `False` maka hasilnya `False`.

<code>a</code>	<code>b</code>	<code>a And b</code>
<code>True</code>	<code>True</code>	<code>True</code>
<code>True</code>	<code>False</code>	<code>False</code>
<code>False</code>	<code>True</code>	<code>False</code>
<code>False</code>	<code>False</code>	<code>False</code>

# Logika OR



- ❑ Operasi logika OR akan bernilai `True` jika salah satu operand bernilai `True`. Apabila kedua operand bernilai `False` maka hasilnya `False`.

<code>a</code>	<code>b</code>	<code>a Or b</code>
<code>True</code>	<code>True</code>	<code>True</code>
<code>True</code>	<code>False</code>	<code>True</code>
<code>False</code>	<code>True</code>	<code>True</code>
<code>False</code>	<code>False</code>	<code>False</code>

# Logika XOR



- ❑ Operasi logika XOR akan bernilai `True` jika salah satu operand bernilai `True` dan operand lainnya bernilai `False`. Apabila kedua operand bernilai `True` maka hasilnya bernilai `False`, begitu juga apabila kedua operand bernilai `False` maka hasilnya bernilai `False`.

a	b	a Xor b
True	True	False
True	False	True
False	True	True
False	False	False