

# MUDAH MENGUASAI MICROSOFT VISUAL BASIC SCRIPT

**Fadhly Permata**  
fadh325@plasa.com

## **Lisensi Dokumen:**

Copyright © 2007 Fadhly Permata

Seluruh dokumen dari **fadh.4t.com** dapat digunakan, dimodifikasi dan disebarluaskan secara bebas untuk tujuan bukan komersial (non-profit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin tertulis terlebih dahulu dari **fadh.4t.com**

---

## KONVENSI KODE VBSCRIPT

### **10.1. Apa itu Konvensi Kode?**

Konvensi kode adalah suatu usulan/saran yang mungkin dapat membantu dalam menuliskan kode menggunakan Microsoft Visual Basic Scripting Edition. Konvensi kode meliputi:

- Konvensi untuk penamaan object, variabel, dan prosedur
- Konvensi komentar
- Konvensi format teks

Alasan utama dalam penggunaan konvensi kode adalah standarisasi gaya penulisan skrip sehingga Anda atau orang lain yang membacanya dapat dengan mudah untuk mengerti tujuan dari skrip yang Anda buat. Penggunaan kode konvensi yang baik dapat menghasilkan skrip yang baik, menarik, source program yang jelas dan konsisten dengan konvensi bahasa lain.

### **10.2. Konvensi Penamaan Konstanta**

Pada VBScript versi terdahulu tidak terdapat mekanisme untuk menentukan konvensi konstanta. Jika menggunakan konstanta, diimplementasikan seperti variabel dengan menggunakan

huruf besar semua. Konstanta yang menggunakan kata lebih dari satu dipisahkan dengan karakter underscore ( \_ ). Sebagai contoh:

```
USER_LIST_MAX  
NEW_LINE
```

Selama tidak melanggar aturan penamaan, Anda dapat menggunakan skema penamaan alternatif lainnya, pada versi ini Anda dapat membuat konstanta dengan menggunakan statemen **Const**. Dengan menggunakan konvensi penamaan konstanta dengan diberi prefix "con". Sebagai contoh:

```
conKonstantaAnda
```

### 10.3. Konvensi Penamaan Variabel

Dengan tujuan memudahkan dalam membaca kembali dan menjaga konsistensi kode, gunakan prefix berikut ini dengan menggunakan deskripsi untuk variabel pada kode VBScript.

Subtipe	Prefix	Contoh
Boolean	bln	blnFound
Byte	byt	bytRasterData
Date/Time	dtm	dtmStart
Double	dbl	dblTolerance
Error	err	errOrderNum
Integer	int	intQuantity
Long	lng	lngDistance
Object	obj	objCurrent
Single	sng	sngAverage
String	str	strFirstName

### 10.4. Ruang Lingkup Variabel

Variabel pada VBScript akan selalu diberikan ruang lingkup (*scope*) sekecil mungkin. Variabel VBScript dapat memiliki ruang lingkup seperti berikut:

Ruang Lingkup	Letak Pendeklarasian	Jarak Lingkupan
Procedure-level	Prosedur Event, Function, atau Sub	Jarak lingkupan hanya pada prosedur yang mendeklarasikan saja.
Script-level	Diluar dari semua prosedur	Jarak lingkupan pada skrip, dan semua prosedur dapat menggunakannya

#### 10.4.1. Prefix Ruang Lingkup Variabel

Dalam pengembangan skrip, tidak perlu menggunakan prefix ruang lingkup dalam memberikan penamaan pada sebuah variabel. Jika tetap ingin menggunakan prefix ikuti konvensi berikut:

Ruang Lingkup	Prefix	Contoh
Procedure-level	Tidak ada	dblVelocity
Script-level	s	sdblVelocity

#### 10.5. Deskripsi Variabel dan Nama Prosedur

Dalam pembuatan variabel atau prosedur sebaiknya menggunakan *case* huruf campuran dan menggunakan nama dari tujuan variabel atau prosedur itu dibuat. Sebagai tambahan, sebaiknya penamaan diawali dengan kata kerja, misalnya `InitNamaArray` atau `TutupDialog`.

Untuk penggunaan jangka panjang, disarankan menggunakan singkatan standar untuk membantu menjaga panjang nama variabel yang layak (*valid*). Umumnya nama variabel yang menggunakan lebih dari 32 karakter akan lebih sulit untuk dibaca. Bila menggunakan singkatan, maka dapat menjaga konsistensi dalam skrip. Sebagai contoh, `Cnt` untuk kata `Count` di dalam skrip dapat digunakan untuk menghindari kesulitan dalam pembacaan.

#### 10.6. Konvensi Penamaan Objek

Berikut ini adalah daftar konvensi untuk penamaan suatu objek yang dapat digunakan saat membuat program dengan VBScript.

Type Objek	Prefix	Contoh
3D Panel	pnl	pnlGroup
Animated Button	ani	aniMailBox
Check Box	chk	chkReadOnly
Combo Box, Drop-down List Box	cbo	cboIndonesia
Command Button	cmd	cmdExit
Frame	fra	fraLanguage
Horizontal Scroll Bar	hsb	hsbVolume
Image	img	imgIcon
Label	lbl	lblHelpMessage
Line	lin	linVertical
List Box	lst	lstPolicyCodes
Spin	spn	spnPages
Text Box	txt	txtLastName
Vertical Scroll Bar	vsb	vsbRate
Slider	sld	sldScale

### 10.7. Konvensi Komentar Kode

Semua prosedur sebaiknya diberikan komentar atau deskripsi yang menggambarkan apa yang dikerjakan dari prosedur tersebut. Deskripsi ini tidak diuraikan secara detail sebab seiring bergantinya waktu dan perubahan skrip dapat menjadikan salah dalam memberikan komentar.

Suatu argumen sebaiknya dijelaskan tujuannya bila suatu prosedur sulit untuk dibaca. Nilai kembalian dari fungsi dan variabel diubah oleh suatu prosedur, terutama yang menggunakan referensi argumen, sebaiknya diberikan komentar pada setiap awal prosedur.

Komentar awal prosedur sebaiknya ditambahkan dalam setiap prosedur. Sebagai contoh lihat sesi "Memformat Kode" di bawah ini.

Judul Komentar	Isi Komentar
Tujuan	Apa yang dapat dilakukan oleh suatu prosedur (bukan bagaimana prosedur bekerja).
Asumsi	Menampilkan semua list variabel external, kontrol, atau elemen lain yang digunakan pada prosedur.
Efek	Menampilkan list variabel external, kontrol, atau elemen lain yang mengalami perubahan dalam prosedur.
Input	Penjelasan dari tiap argumen yang tidak jelas. Setiap argumen sebaiknya dipisahkan dalam baris yang berbeda.
Nilai Kembalian	Penjelasan dari nilai kembalian ( <i>return-value</i> ).

Dalam melakukan konvensi komentar kode, gunakan point-point berikut:

- Setiap variabel sebaiknya menggunakan komentar yang berbeda baris.
- Setiap nama variabel, kontrol, dan prosedur sebaiknya menggunakan nama yang jelas dengan tujuannya, keterangan digunakan hanya untuk sekedar informasi detail.
- Pada awal skrip, sebaiknya memberikan penjelasan yang menguraikan informasi dari skrip tersebut, seperti object-objek yang digunakan, prosedur, algoritma, kotak dialog, dan ketergantungan sistem (*system dependencies*) yang digunakan. Terkadang potongan *pseudocode* dapat menggambarkan tujuan.

## 10.8. Memformat Kode

```
*****
' Tujuan: Mencari letak posisi pertama user tertentu pada
' array userList.
'
' Input: strUserList(): daftar user yang akan dicari.
' strTargetUser: nama user yang akan dicari.
'
' Returns: Indek dari array yang memiliki value sama dengan
' strTargetUser.
' Jika tidak ditemukan nilai kembaliannya adalah -1.
*****

Function intFindUser (strUserList(), strTargetUser)
Dim i ' Loop counter.
Dim blnFound ' Target found flag

intFindUser = -1
i = 0 ' Inisialisasi jumlah perulangan

Do While i <= Ubound(strUserList) and Not blnFound
If strUserList(i) = strTargetUser Then
blnFound = True ' Set flag menjadi True
intFindUser = i ' Set return value ke jumlah perulangan
End If

i = i + 1 ' tambahkan jumlah perulangan
Loop
End Function
```