

Mengenal Visual Basic Editor

Di mana macro dibuat? Tidak di worksheet, bukan di workbook, namun di area khusus yang disebut dengan istilah Visual Basic Editor. Di dalam Editor inilah, Anda merangkai script pemrograman *Visual Basic for Application* agar bisa menciptakan macro untuk tugas yang spesifik.

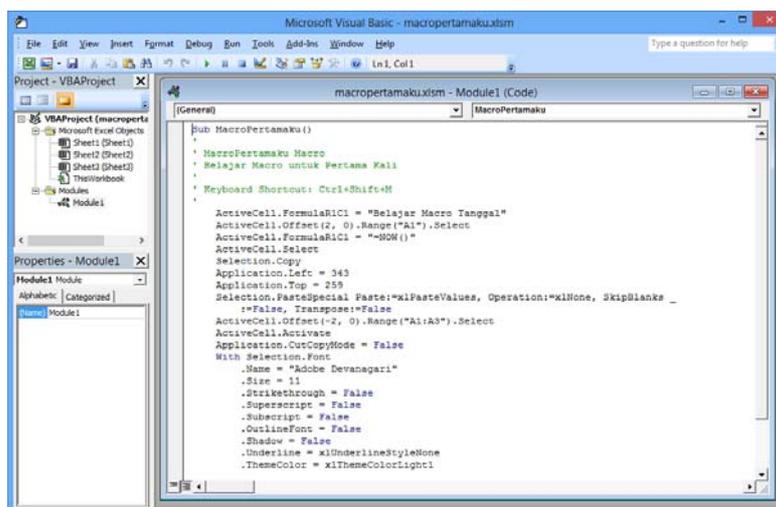
Walaupun terpisah dari MS Excel, namun Visual Basic Editor hanya bisa diluncurkan apabila terlebih dulu Anda menjalankan MS Excel. Itu artinya, Visual Basic Editor bukanlah aplikasi stand-alone yang bisa Anda jalankan dengan menekan tombol **Start > All Programs** pada MS Windows.

Meluncurkan Visual Basic Editor

Di bab pertama sebelumnya, Anda sudah mengenal langkah-langkah dasar meluncurkan Visual Basic Editor. Selain menggunakan ikon **Visual Basic** yang ada dalam tab **Developer**, Anda juga bisa menjalankan Visual Basic Editor dengan menekan tombol **Alt+F11**.

Mengenal Bagian-Bagian di dalam Visual Basic Editor

Visual Basic Editor dibagi dalam beberapa panel dengan fungsi dan tujuan spesifik. Marilah kita mengenali tiap-tiap bagian itu:

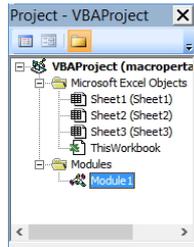


Tampilan Visual Basic Editor yang utuh

Project Explorer

Panel Project Explorer secara default diletakkan di pojok kiri atas. Fungsinya untuk melakukan navigasi terhadap seluruh objek yang ada di dalam proyek pembuatan macro sebuah workbook. Objek-objek yang dikenali di dalam Project Explorer antara lain: Microsoft Excel Objects, Forms, Modules, dan Class Modules.

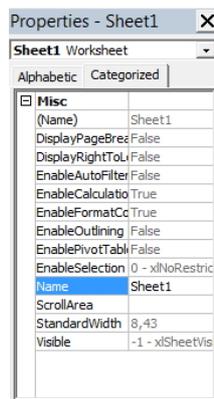
Kalau panel ini tidak terlihat, maka Anda bisa mengaktifkannya dengan menekan menu **View > Project Explorer**. Atau, percepat dengan menekan tombol **Ctrl+R** atau klik ikon  yang ada di dalam toolbar.



Panel Project Explorer untuk mengeksplorasi objek-objek di dalam proyek macro

Panel Properties

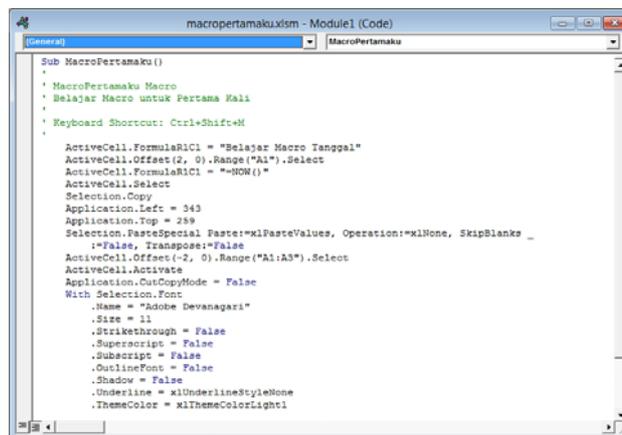
Panel Properties berfungsi untuk menampilkan berbagai macam properti yang melekat pada sebuah objek. Apabila panel ini belum terlihat, aktifkanlah dengan menekan menu **View > Properties Window** atau cukup dengan menekan tombol **F4**. Tampilan di dalam panel ini akan berbeda-beda, tergantung objek yang ingin diubah properties-nya (seperti warna, jenis font, dan lain sebagainya).



Panel Properties akan memiliki tampilan beragam, tergantung objek yang aktif saat itu

Panel Code

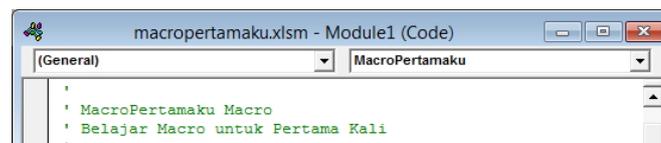
Bisa jadi, pusat perhatian Anda ketika bekerja dengan Visual Basic for Application terfokus pada panel ini. Inilah panel terbesar yang ada di dalam Visual Basic Editor. Fungsi panel Code adalah untuk melihat, merancang, dan melakukan fungsi editing pada script VBA agar macro bekerja seperti yang diharapkan.



```
Sub MacroPertamaku ()  
    ' MacroPertamaku Macro  
    ' Belajar Macro untuk Pertama Kali  
    ' Keyboard Shortcut: Ctrl+Shift+M  
    '   
    ActiveCell.FormulaR1C1 = "Belajar Macro Tanggal"  
    ActiveCell.Offset(2, 0).Range("A1").Select  
    ActiveCell.FormulaR1C1 = "=NOW()  
    ActiveCell.Select  
    Selection.Copy  
    Application.Left = 343  
    Application.Top = 259  
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _  
    :=False, Transpose:=False  
    ActiveCell.Offset(-2, 0).Range("A1:A3").Select  
    ActiveCell.Activate  
    Application.CutCopyMode = False  
    With Selection.Font  
        .Name = "MSoboe Devanagari"  
        .Size = 11  
        .Strikethrough = False  
        .Superscript = False  
        .Subscript = False  
        .OutlineFont = False  
        .Shadow = False  
        .Underline = xlUnderlineStyleNone  
        .ThemeColor = xlThemeColorLight1  
    End With  
End Sub
```

Panel Code adalah panel dengan ukuran paling besar dan mungkin juga yang berperan paling penting dalam pembuatan VBA

Di bagian atas pada panel Code ini terdapat dua menu kombo, sebelah kiri dan kanan. Yang kiri disebut **Object Selector** dan yang kanan diberi nama **Event Selector**.

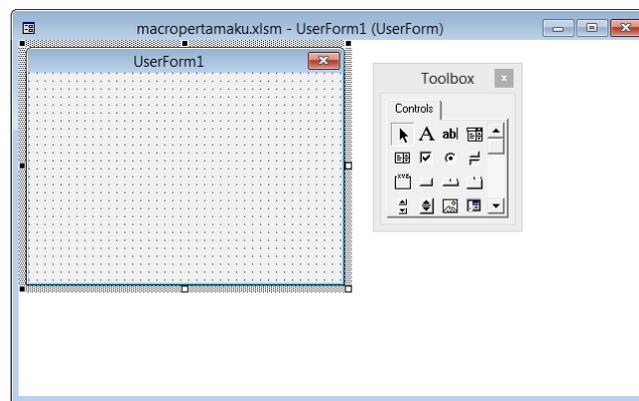


Menu kombo sebelah kiri-atas disebut Object Selector dan kanan-atas disebut Event Selector

Object Selector digunakan apabila ingin melihat script macro yang melibatkan objek atau koleksi objek di dalam workbook. Sedangkan Event Selector digunakan untuk memilih event pada sebuah objek.

Panel Object

Kalau Anda ingin merancang macro yang indah dan fungsional secara visual, maka Anda membutuhkan panel Object. Jika belum terlihat di dalam layar monitor, munculkanlah panel ini dengan mengklik menu **View > Object** atau dengan menekan tombol **Shift+F7**. Anda bisa memanfaatkan panel ini ketika ingin membuat UserForm, misalnya (klik menu **Insert > UserForm**).



Tampilan panel Object

Panel Object Browser

Bisa dikatakan, panel Object Browser ini adalah "Windows Explorer"-nya Visual Basic Editor karena bisa digunakan untuk menampilkan atau melakukan pencarian terhadap semua objek, koleksi objek, properti-properti, method, atau event yang dibuat dalam script VBA.

Cukup tekan tombol **F2** untuk menampilkan panel ini. Yang perlu Anda lakukan hanyalah mengetik kata kunci di dalam kotak teks yang telah disediakan. Sebagai contoh, Anda bisa mengetik **font** untuk mencari Library, Class, dan lain sebagainya yang berkaitan dengan font.



Pencarian di dalam panel Object Browser

Bagaimana Menulis Kode VBA di dalam Panel Code?

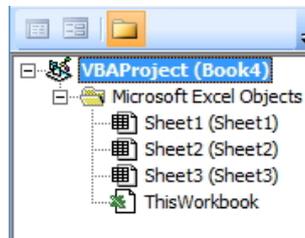
Bicara tentang macro maka tak akan pernah dilepaskan dari script VBA. Secara umum, script atau kode-kode VBA itu bisa ditulis di dalam panel Code lewat dua metode, yaitu:

- Menulis kode langsung secara manual.
- Menggunakan Macro Recorder untuk selanjutnya diubah menjadi kode-kode VBA.

Untuk metode kedua, kita telah melakukannya di bab pertama buku ini. Sedangkan jika memilih metode pertama, maka salah satu syaratnya adalah mampu menulis kode-kode VBA dengan baik dan benar. Metode pertama memang paling sulit. Namun metode pertama lebih menjanjikan hasil yang lebih variatif dibandingkan metode kedua.

Berikut salah satu langkah-langkah untuk menulis kode secara langsung dan manual untuk menciptakan macro sederhana:

1. Buatlah workbook baru dan masuklah ke dalam **Visual Basic Editor** (tekan saja tombol **Alt+F11**).
2. Klik pada nama workbook yang tercatat di dalam panel **Project**.



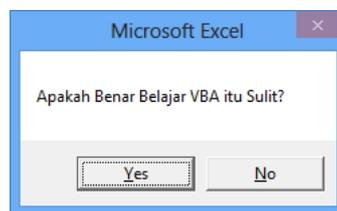
Klik pada nama workbook di mana macro akan dibuat

3. Buatlah module baru dengan mengklik menu **Insert > Module**.
4. Kalau sudah, ketik script di bawah ini di dalam panel Code:

```
Sub TanyaJawab()  
    Pesan = "Apakah Benar Belajar VBA itu Sulit?"
```

```
Jawaban = MsgBox(Pesan, vbYesNo)
If Jawaban = vbNo Then MsgBox "Anda Memang Hebat!"
If Jawaban = vbYes Then MsgBox "Ayo Belajar Terus!"
End Sub
```

5. Tekan tombol **F5** untuk menjalankan script di atas. Nanti, Anda akan melihat kotak dialog dengan pertanyaan di bawah ini:



Kotak dialog lengkap dengan dua alternatif jawaban

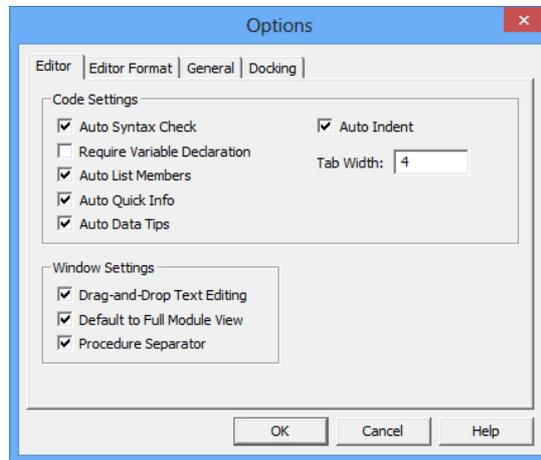
6. Jika menjawab **Yes** atau **No**, maka sudah disiapkan jawaban yang tepat seperti script di atas.

Baik, sampai sejauh ini Anda sudah berhasil menciptakan dua buah macro. Pertama, Anda membuat macro lewat fungsi perekaman. Kedua, Anda menulis script VBA untuk mendapatkan hasil seperti pada gambar di atas.

Mengatur Lingkungan Kerja Visual Basic Editor

Sebelum kita masuk lebih dalam lagi dalam pembahasan Visual Basic for Application (VBA) untuk perancangan macro, akan lebih baik jika terlebih dulu kita membuat pengaturan-pengaturan lingkungan kerja. Tujuannya, agar ketika melakukan *coding* (penulisan script), kita merasa lebih nyaman.

Untuk mengatur lingkungan kerja VBE, Anda bisa mengklik menu **Tools > Options**. Tunggu beberapa saat dan akan muncul jendela seperti gambar di bawah ini. Jendela tersebut terbagi dalam beberapa tab, yaitu Editor, Editor Format, General, dan Docking.



Jendela Options untuk mengatur lingkungan kerja VBE

Memfaatkan Tab Editor

Pertama-tama, pastilah Anda akan melihat tab Editor di dalam jendela Options di atas. Tab ini berfungsi untuk mengatur bagaimana VBE membantu kita menulis script. Berikut beberapa pilihan penting yang ada di dalam tab tersebut:

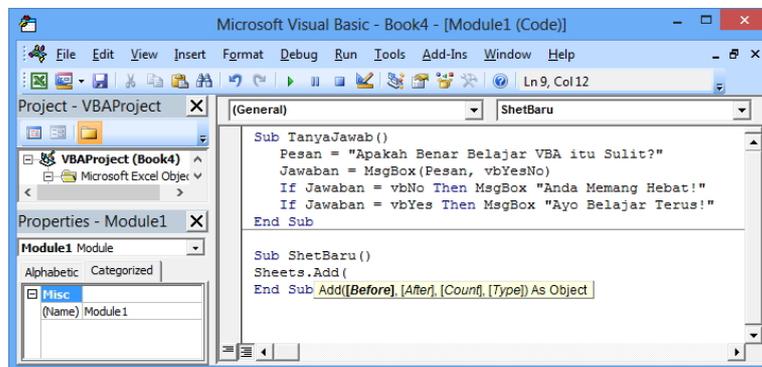
AUTO SYNTAX CHECK OPTION

Dalam kondisi aktif, maka Anda mengizinkan VBE menampilkan pesan kesalahan (*syntax error*) dalam bentuk kotak dialog ketika terjadi kesalahan dalam penulisan kode VBA. Kalau Anda merasa keberadaan kotak dialog yang akan muncul terus menerus ketika terjadi kesalahan malah mengganggu, maka lebih baik nonaktifkan opsi ini. Lagipula, apabila VBE menemukan adanya kesalahan di dalam kode, maka kode tersebut akan ditandai dengan warna yang berbeda.

REQUIRE VARIABLE DECLARATION UNIT

Kalau Anda mengaktifkan opsi ini, maka setiap kali membuat module baru, VBE akan menyisipkan satu baris kode berikut ini:

```
Option Explicit
```

Auto Quick Info membantu Anda menulis script secara cepat dan tanpa kekhawatiran terjadinya salah ketik

AUTO DATA TIPS OPTION

Jika dalam kondisi aktif (tercentang), maka Visual Basic Editor akan menampilkan nilai dalam sebuah variable saat kursor diletakkan di atas variable itu. Proses ini akan terjadi ketika Anda sedang melakukan debugging terhadap kode.

AUTO INDENT

Berfungsi untuk membuat agar baris-baris kode menjorok ke arah kanan (dalam) seperti baris-baris sebelumnya ketika Anda menulis script dan menekan tombol Enter. Fitur ini lebih baik diaktifkan agar script tampak lebih rapi.

Kalau Anda ingin melakukan indentasi (membuat baris menjorok ke dalam) secara manual, gunakanlah tombol **Tab**. Sedangkan untuk “menarik” (outdent) kembali indentasi itu ke arah kiri, tekan tombol **Shift+Tab**.

DRAG-AND-DROP TEXT EDITING

Kalau aktif, maka Anda bisa memindahkan teks di dalam script ke baris atau tempat lain dengan cara klik-drag.

DEFAULT TO FULL MODULE VIEW

Dalam kondisi aktif, procedure-procedure yang dibuat di dalam jendela Code akan ditampilkan secara utuh. Namun kalau tidak aktif, maka procedure itu akan ditampilkan bagian per bagian dalam satu waktu yang sama.

PROCEDURE SEPARATOR

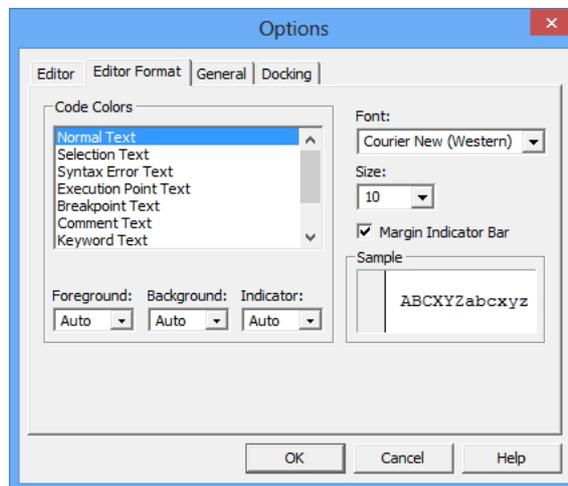
Berfungsi untuk menampilkan garis pemisah/separator di bawah setiap procedure sehingga Anda bisa melihat masing-masing procedure secara utuh walaupun procedure itu disajikan dalam satu jendela yang besar.

Memanfaatkan Tab Editor Format

Tab **Editor Format** digunakan untuk membantu Anda mengatur jenis, ukuran, dan warna font. Secara default, font yang digunakan untuk penulisan script adalah Courier New (Western). Anda bisa mengubah font itu dengan yang lain sesuai dengan keinginan Anda.

Berikut beberapa pilihan yang ada di dalam tab Editor Format tersebut:

- **Code Color:** Anda bisa mengatur warna font untuk script VBA (Foreground), warna latar belakang untuk script (Background), dan indikator untuk script (Indicator). Sebagai contoh, jika Anda ingin mengatur teks yang dibuat untuk penulisan komentar, pilihlah **Keyword Text** dan ubahlah warna Foreground, Background, dan Indicator-nya.
- **Font:** berfungsi untuk menentukan atau memilih font yang digunakan dalam penulisan script VBA.
- **Size:** berfungsi untuk menentukan atau memilih ukuran font saat penulisan script VBA.
- **Margin Indicator Bar :** jika aktif (tercentang) berfungsi untuk menampilkan indikator area margin secara vertikal di dalam jendela Code. Fungsinya untuk membantu Anda melihat indikator visual ketika proses debugging terhadap code dilakukan.



Tab Editor Format dipakai untuk mengatur bagaimana teks akan terlihat di dalam jendela Code

Bekerja Menggunakan Object

Jika Anda seorang akuntan, administrasi, atau staf kantor yang memanfaatkan MS Excel untuk tugas harian, maka besar kemungkinan Anda tidak akan berkenalan dengan satu istilah ini, yaitu: Object.

Namun, saat Anda bekerja menggunakan *Visual Basic for Application* (VBA), Anda akan melihat MS Excel sebagai objek utuh dan di dalamnya terdapat objek-objek lain yang lebih kecil. Kalau sudah begini, kita akan bekerja dengan objek yang di dalamnya terdapat objek lainnya, atau objek yang terstruktur (*hierarchy*).

Nah, MS Excel sendiri sering disebut dengan istilah **Application Object**.

Mengenal Hierarki pada Sebuah Objek

Telah dikatakan kalau MS Excel adalah sebuah objek besar, bernama Application Object. Di dalam object ini, terdapat objek-objek lainnya yang lebih kecil sehingga membentuk struktur hierarki. Ada banyak objek-objek kecil di dalam sebuah objek yang lebih besar.

Sebagai contoh, Application Object mengandung objek-objek yang lebih kecil. Berikut beberapa objek tersebut (objek yang paling sering digunakan):

- AddIn
- Window
- Workbook
- WorksheetFunction

Nah, di dalam objek-objek di atas kita bisa temukan lagi objek-objek yang lebih kecil. Sebagai contoh, Workbook Object di atas terdiri dari beberapa objek sebagai berikut:

- AddIn
- Window
- Workbook
 - Chart
 - Name
 - VBProject
 - Window
 - Worksheet
- WorksheetFunction

Nanti, di dalam objek Worksheet di atas, kita juga akan temukan objek-objek lainnya. Contohnya sebagai berikut:

- AddIn
- Window
- Workbook
 - Chart
 - Name
 - VBProject
 - Window

- Worksheet
 - Comment
 - Hyperlink
 - Name
 - PageSetup
- WorksheetFunction

Mengenal Collections

Collections, atau nanti akan ditulis sebagai “koleksi”, adalah kumpulan objek yang memiliki tipe yang sama. Salah satu cara mengenali sebuah koleksi adalah penulisan huruf “s” (plural dalam Bahasa Inggris) di belakang penamaan tersebut. Berikut beberapa koleksi yang biasa dipakai:

- **Workbooks:** collection (koleksi-koleksi) yang berisi objek-objek Workbook yang saat itu sedang aktif.
- **Worksheet:** collection yang berisi objek-objek worksheet yang ada di dalam sebuah objek Workbook.
- **Charts:** collection dari objek Chart yang ada di dalam sebuah objek Workbook.
- **Sheets:** collection dari objek Worksheet yang ada di dalam sebuah objek Workbook.

Inilah Caranya Menulis Objects di dalam VBA

Ketika Anda ingin menggunakan sebuah object, atau collection, maka Anda harus mengidentifikasi keduanya itu—object atau collection—dengan menuliskannya secara tepat.

Dalam penulisannya, objek dan collection dipisahkan menggunakan tanda titik “.”. Sebagai contoh, jika Anda ingin memilih Sheet1 dalam sebuah workbook, maka cukup tuliskan:

```
Sheets("Sheet1").Select
```

Jika ingin merujuk pada workbook (baca: nama file XLS) tertentu, maka nama workbook itu harus disebut di dalam collections Workbooks. Misalnya seperti ini

```
Workbooks("Datapenjualan.xls").Sheets("Sheet1").Select
```

Ingat, beri tanda kutip ganda untuk merujuk pada objek. Jika tidak menggunakan tanda kutip ganda maka tulisan tersebut, misalnya Sheet1 (tanpa tanda kutip), akan dianggap sebagai variable.

SAAT BEKERJA MENGGUNAKAN OBJEK APPLICATION

Application object adalah MS Excel itu sendiri, induk dari semua objek. Oleh karena itu, jika ingin bekerja menggunakan Application Object, awalilah dengan menulis: **Application**.

Semua objek yang ada di MS Excel ada di bawah Application ini. Nah untuk menulis objek-objek bawahan itu, maka pisahkan satu objek yang lebih tinggi dan lebih rendah tersebut menggunakan operator titik.

Sebagai contoh, Anda bisa menuliskan:

```
Application.Workbooks("Datapenjualan.xlsx")
```

Kalau Anda hanya ingin bekerja menggunakan salah satu sheet di dalam Workbook "Datapenjualan.xlsx" itu, maka Anda bisa menulis:

```
Application.Workbooks("Datapenjualan.xlsx").Worksheets("Sheet1")
```

Masih lanjut lagi, jika Anda ingin bekerja dengan nilai yang ada di dalam sel A1, maka tulis seperti ini:

```
Application.Workbooks("Datapenjualan.xlsx").Worksheets("Sheet1").Range("A1").Value
```

CATATAN:

Ketika mengetik script di atas, buatlah dalam satu baris!

Ketika kita menulis objek bernama Range yang langsung merujuk pada sel A1, maka kita telah melakukan proses yang disebut *fully qualified reference*, artinya rujukan itu pasti dan jelas.

MEMPERPENDEK SCRIPT

Jika bekerja menggunakan objek Application, salah satu kelemahan mendasar adalah, script itu menjadi terlalu panjang dari kanan ke kiri. Oleh karena itu, untuk banyak kasus, Anda bisa mengesampingkan Application. Sebagai contoh, script panjang di atas bisa diperpendek menjadi:

```
Workbooks("Datapenjualan.xlsx").Worksheet("Sheet1").Range("A1").Value
```

Workbooks object pun bisa dikesampingkan kalau file workbook yang aktif saat itu bernama "Datapenjualan.xlsx" sehingga script di atas menjadi lebih pendek lagi menjadi:

```
Worksheet("Sheet1").Range("A1").Value
```

Lantas, apakah Anda yakin kalau Sheet1 ada di dalam workbook tersebut dan dalam kondisi aktif? Kalau demikian, potong saja script di atas menjadi:

```
Range("A1").Value
```

Bekerja Menggunakan Property

Property, atau biasa ditulis Properti, adalah karakteristik yang dimiliki atau melekat pada sebuah objek. Bagaikan objek mobil, Anda bisa memilih properti warna yang diisi dengan nilai "merah", "putih",

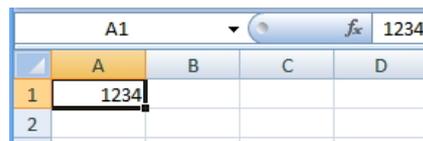
“hijau”, dan seterusnya. Properti ini akan membuat object yang telah kita tunjuk di atas menjadi lebih fungsional.

Sebagai contoh, jika Anda menunjuk objek worksheet “Sheet1” seperti ini, maka apa yang bisa Anda lakukan selanjutnya? Tidak ada, bukan?

Workbooks (“Datapenjualan.xlsx”).Worksheet (“Sheet1”)

Maka itu, kita pilih gunakan properti. Dalam contoh di atas, properti tersebut adalah Value. Untuk menguji bagaimana properti di atas bekerja, maka lakukanlah langkah-langkah sebagai berikut:

1. Pindahlah ke Worksheet terlebih dulu dan tulis angka 1234 ke dalam sel A1.



The image shows a screenshot of an Excel spreadsheet. The active cell is A1, and its value is 1234. The formula bar at the top shows the value 1234. The spreadsheet has columns A, B, C, and D, and rows 1 and 2. Cell A1 is highlighted with a black border.

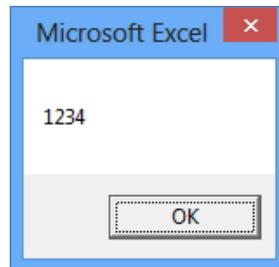
Tuliskan data di dalam sel A1

2. Beralihlah ke dalam Visual Basic Editor.
3. Tulis script seperti berikut ini di dalam panel Code:

```
Sub TampilkanData()  
    Tampilan = Worksheets("Sheet1").Range("A1").Value  
    MsgBox Tampilan  
End Sub
```

4. Setelah itu, tekanlah tombol F5.

Nanti, Anda akan melihat nilai: 1234 di dalam kotak dialog yang muncul di layar monitor. Angka tersebut, yaitu 1234, didapat dengan mengambil properti dari object bernama Range.



Angka di dalam kotak dialog

Mengubah Property Sebuah Objects

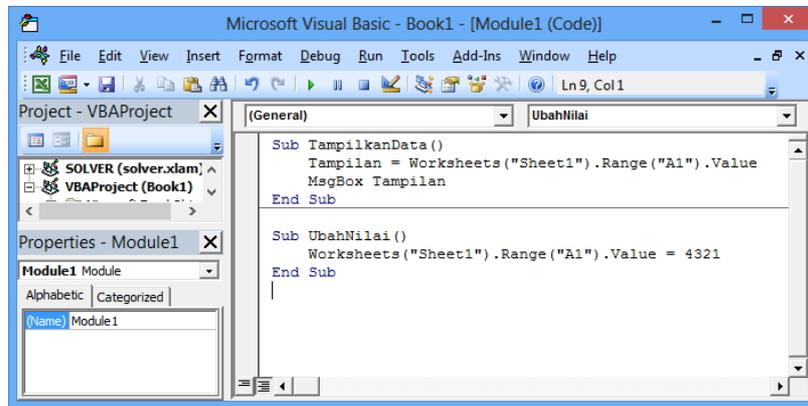
Kalau nilai di dalam sebuah property bisa ditampilkan, maka itu artinya nilai tersebut juga bisa diubah, bukan? Memang demikian adanya. Nilai sebuah property yang melekat di dalam object bisa diubah sesuai keinginan.

Ada beberapa cara untuk mengubah nilai property sebuah object. Berikut cara-cara yang dikenal:

MENULIS LANGSUNG DI DALAM SCRIPT

Untuk mengubah nilai property di dalam sebuah object, Anda bisa menulis nilai yang baru itu secara langsung pada script. Cobalah lanjutkan script di atas dengan menulis beberapa baris script VBA di bawah ini:

```
Sub UbahNilai()  
    Worksheets("Sheet1").Range("A1").Value = 4321  
End Sub
```



Script lanjutan untuk mengubah nilai di dalam object Range("A1")

Yang terjadi kemudian, di dalam sel A1 akan tertulis angka yang baru, yaitu: 4321.

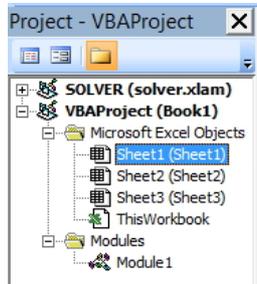
	A1		
	A	B	C
1	4321		
2			

Angka di dalam sel A1 yang telah berubah

MENGUBAH NILAI PROPERTY MENGGUNAKAN PANEL PROPERTIES

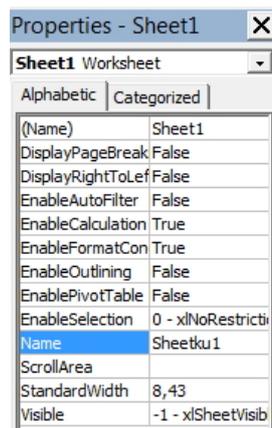
Ada beberapa nilai property yang juga bisa diubah tanpa harus melewati penulisan script secara langsung. Cara pengubahannya, yaitu dengan memanfaatkan panel **Properties**. Sebagai contoh, lakukan langkah-langkah sebagai berikut:

1. Klik pada **Sheet1(Sheet1)** yang ada di dalam panel **Project**.



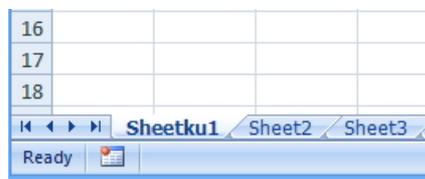
Memilih Sheet1 di dalam panel Project

2. Di dalam panel **Properties**, ubah value untuk **Name** menjadi: **Sheetku1**.



Mengubah value Name menjadi Sheetku1

3. Sekarang, masuklah ke dalam worksheet MS Excel dan lihatlah nama sheet yang sudah berganti itu.

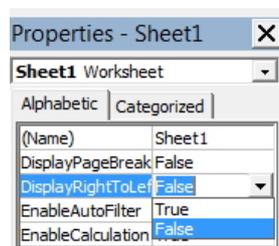


Nama Sheet1 yang sudah berubah menjadi Sheetku1

Umumnya, yang bisa diganti menggunakan panel Properties antara lain: Name, Height, Width, Left, Top, dan lainnya.

Ketika mengganti value sebuah Properti menggunakan panel Properties, adakalanya Anda melakukan salah satu atau dari variasi-variasi berikut ini:

- **Menulis secara langsung:** yang sudah kita praktekan di atas.
- **Memilih opsi:** ada kalanya, value sebuah properti sudah ditentukan oleh Visual Basic, misalnya operator Boolean seperti **Enabled** yang hanya menawarkan salah satu dari pilihan berikut: **True** atau **False**.



Mengubah nilai dengan cara memilih salah satu opsi yang telah disediakan

Bekerja Menggunakan Methods

Jika property adalah: apa yang melekat di dalam sebuah object, maka Methods berbicara tentang: apa yang bisa dilakukan terhadap object itu. Karena berbicara tentang “apa yang bisa dilakukan terhadap object itu”, maka method bisa mempengaruhi nilai sebuah property.

Sebagai contoh, jika Anda menulis script seperti ini:

```
Sub HapusData()  
    Range("A1").ClearContents  
End Sub
```

Maka itu artinya, sel A1 akan dibersihkan dari data apa pun yang ada di dalamnya.

Jika sebuah property bisa memiliki value, maka methods pun bisa memiliki arguments. Nah, arguments akan mengatur apa yang akan dilakukan oleh method itu terhadap objek. Untuk menulis arguments, Anda perlu memisahkan arguments itu dengan method memanfaatkan tanda spasi.

Mari kita bedah script di bawah ini:

```
Sub KopiData()  
    Worksheets("Sheet1").Activate  
    Range("A1").Copy Range("B1")  
End Sub
```

Ada dua buah method di atas, yaitu Activate yang berfungsi untuk memilih object Worksheets("Sheet1") dan Copy. Kita menggunakan method Copy untuk mengkopi data yang ada di dalam object Range("A1"), yaitu data di dalam sel A1. Lantas, method Copy ini memiliki argument, yaitu tujuan di mana data yang telah dikopi itu akan diletakkan. Dalam hal ini, argument-nya adalah Range("B1").