

BAB 1

PENGENALAN PADA VBA EXCEL

Objektif :

- Mengetahui Dasar-dasar Makro
- Mengetahui Lingkungan Visual Basic Editor
- Mengetahui Jenis – jenis data, variable dan Array
- Mengetahui Operator pada Visual Basic Editor
- Dapat Membuat Range dan Worksheet
- Dapat Membuat Objek Message Box dan Input Box

Sejak pengembangan Excel 5.0, Microsoft memperkenalkan bahasa pemrograman yang dinamakan Visual Basic for Application (VBA). VBA adalah bahasa pemrograman yang intuitive menggunakan perancangan object – oriented . Memiliki persamaan yang besar dengan Visual Basic (VB). Jika memakai VBA, kita akan memahami bagaimana Visual Basic berkerja. VBA sangat berbeda dengan bahasa Macro , dala setiap lembar kerja elektronik disediakan macro yang berfungsi untuk menjalankan perintah – perintah rutin yang sering diulang –ulang sedangkan VBA dipergunakan untuk men gontrol objek –objek yang terdapat didalamnya. Dengan VBA juga dapat mengontrol objek –objek aplikasi lainnya seperti Word, Power Point, Access.

Jadi Visual Basic for Application (VBA) adalah bahasa pemrograman yang cukup populer dan mudah untuk dipelajari. Dengan Visual Basic for Application (VBA), kita dapat mengembangkan prosedur kecil (macro) yang sering digunakan menjadi lebih mudah dan membuat suatu latihan dalam waktu lebih singkat. Visual Basic for Application tidak hanya terdapat pada Microsoft Excel saja, tetapi terdapat pula VBA untuk word, VBA untuk project, VBA untuk Access dan semuanya itu berbeda jauh satu sama lain. VBA juga merupakan program yang sangat canggih, dimana dapat mengembangkan macro sederhana (VBA Procedures) dengan singkat dan menarik .

1.1 Membuat Makro

Macro adalah serangkaian perintah dan fungsi yang disimpan di dalam module Microsoft Visual Basic Editor untuk menjalankan tugas secara otomatis. Menggunakan macro berarti menambahkan kode VBA pada worksheet.

Fitur macro berada di dalam sebuah tab yaitu tab Developer. Jika tab tersebut belum tampil di dalam ribbon, berikut cara untuk menampilkannya pada Microsoft Excel 2010:

1. Buka program MS.Excel 2010
2. Klik tab **File** lalu pilih **Options**. Klik **Customize Ribbon**, di dalam kotak Main Tabs beri tanda cek pada pilihan **Developer**.
3. Klik **OK**. Tab Developer tampil di dalam ribbon.

Pemberian nama pada makro:

- Maksimum 46 huruf
- Diawali dengan huruf namun dibelakang karakter pertama boleh menggunakan angka Tidak boleh menggunakan spasi dan titik.
- Karakter yang digunakan sebagai short key tidak boleh berupa angka atau karakter khusus.


Pada kotak teks dalam kotak dialog record macro terdapat 3 buah pilihan diantaranya adalah :

- ***This Workbook***, akan menyimpan macro yang akan kita buat didalam workbook yang aktif. Tetapi makro tersebut masih berfungsi apabila membuka sebuah workbook baru didalam satu jendela Excel.
- ***New Workbook***, akan menciptakan sebuah lembar kerja baru dan menyimpan macro yang akan kita buat didalam workbook yang baru. Macro ini pun masih bekerja pada workbook yang kita buka sebelumnya.
- ***Personal Macro Workbook***, akan menyimpan macro yang akan kita buat didalam workbook yang aktif dan berfungsi pada seluruh jendela Excel yang buka. Dalam menggunakan pilihan ini penulis menyarankan supaya berhati-hati, karena apabila terjadi kesalahan akan berakibat serius.

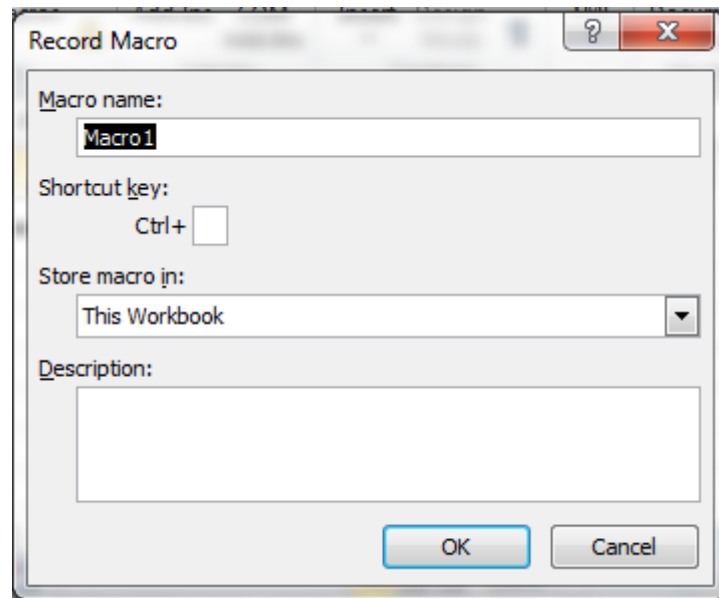
Latihan1 : Kita akan membuat cell yang memiliki ketentuan sebagai berikut

Border : Outline

Langkah – Langkah merekam Macro :

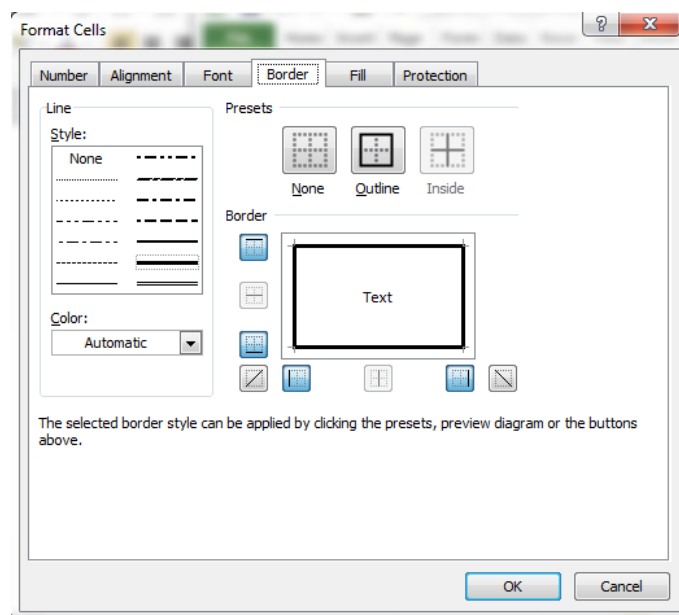
1. Letakkan sel aktif pada salah satu sel yang akan diformat misal A1.
2. Pilih tab Developer, di dalam group Code klik tombol **Record Macro**  .

3. Maka akan muncul kotak dialog seperti pada gambar 1.1

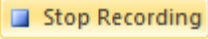


Gambar 1.1 : Kotak Dialog Record Macro

4. Isikan nama pada kotak Macro Name dengan nama FormatAsResult.
5. Pada Kotak Shortcut Key isikan huruf F kapital dengan menekan tombol Shift F.
6. Klik OK, proses perekaman dimulai.
7. Lakukan langkah pemformatan dengan klik kanan pada sel aktif yaitu A1 lalu pilih **Format Cells**
8. Maka akan muncul kotak dialog seperti pada gambar 1.2 kemudian dari tab Borders, klik tombol outline dan style seperti dibawah ini, klik OK



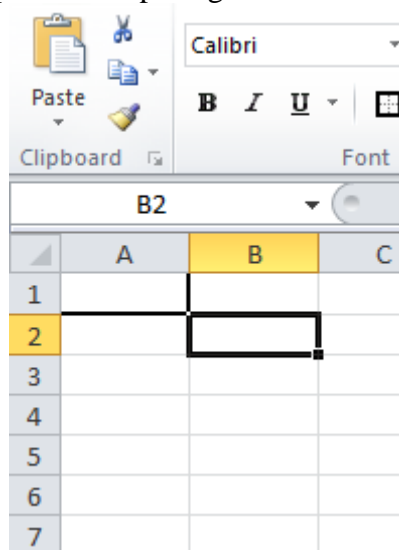
Gambar 1.2 : Definisi New Style

9. Kembali ke tab Developer, klik tombol  untuk mengakhiri perekaman macro.

10. Dapat dilihat dalam modul yang terdapat pada VBA sintaksnya dengan menekan tombol Alt + F11 yaitu:

```
Sub FormatAsResult()  
'  
' FormatAsResult Macro  
'  
' Keyboard Shortcut: Ctrl+Shift+F
```

11. Hasil Pada Excel dapat terlihat pada gambar 1.3



Gambar 1.3 : Kotak Dialog Hasil Makro pada Excel


Latihan2 : Buatlah Judul kolom dengan ketentuan

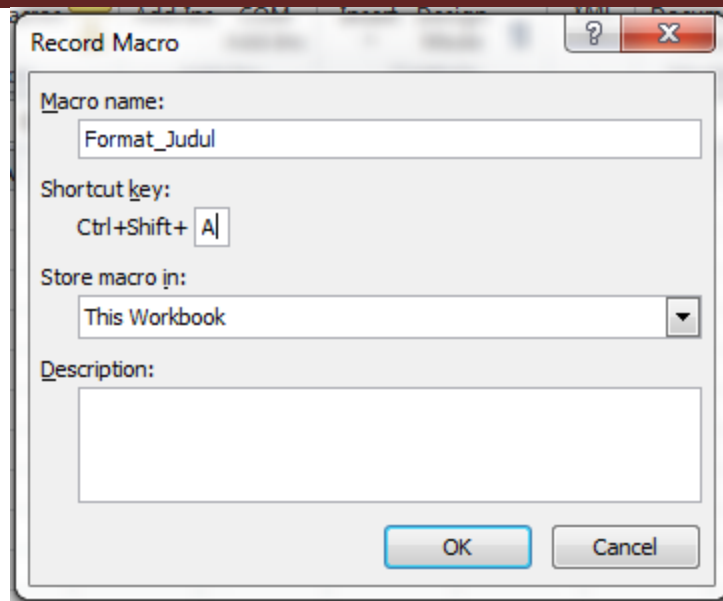
Font: Times New Roman, Bold, Biru

Tab Aligment, Vertical Center, Horizontal Center, Wrap Text

Border dan Outline.

Langkah – Langkah merekam makro :

1. Letakkan sel aktif pada salah satu sel yang akan diformat misal A2.
2. Pilih tab Developer, di dalam group Code klik tombol **Record Macro** 
3. Pada saat itu akan muncul kotak dialog seperti gambar 1.4



Gambar 1.4: Kotak Dialog Record Macro Format_Judul

4. Isikan Macro Name dengan Format_Judul
5. Pada Kotak shortcut key isikan huruf A kapital.
6. Klik OK, proses perekaman dimulai.
7. Lakukan langkah pemformatan dengan klik kanan pada sel aktif lalu pilih **Format Cells**
8. Maka akan muncul kotak dialog seperti pada gambar 1.2 kemudian dari tab Borders, klik tombol outline dan style seperti dibawah ini, klik OK
9. Setelah selesai, klik tombol stop Recording.

Langkah – langkah menjalankan makro sebagai berikut :

1. Bloklah sel- sel yang ingin diformat sebagai Judul Kolom.

	A	B	C	D	E	F	G
							Nama
1	Nama	Kategori	Nama Bar	Nama Sup	Harga Satu	Tgl Penjual	Pembeli

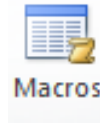
Gambar 1.5 Judul – Judul Sebelum diformat

2. Gunakan shortcut key Ctrl+Shift+A untuk mengaktifkan mkro.
3. Hasilnya terlihat seperti pada gambar 1.6

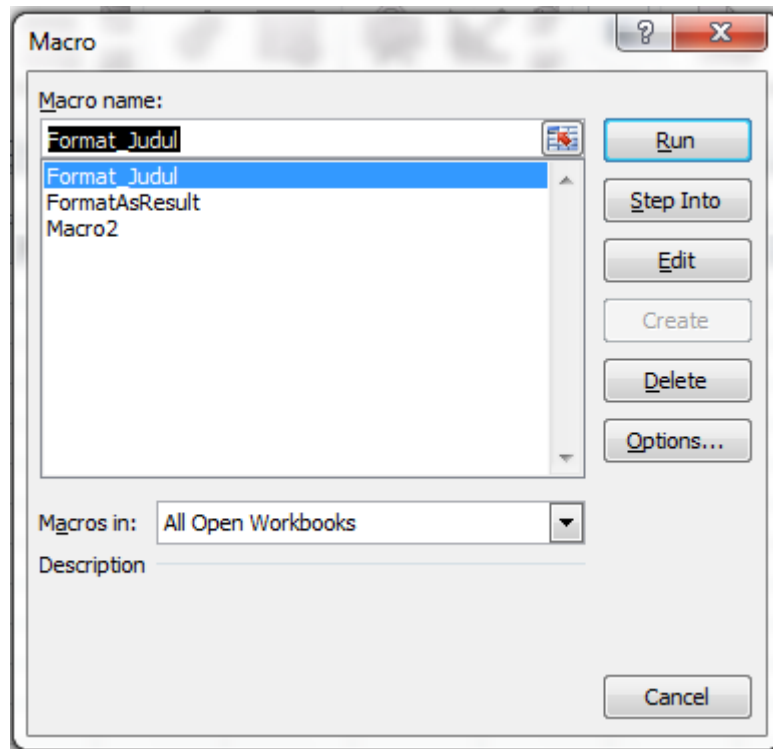
	A	B	C	D	E	F	G
1	Nama	Kategori	Nama Barang	Nama Suplier	Harga Satuan	Tgl Penjual	Nama Pembeli

Gambar 1.6 Judul – Judul setelah diformat

Menjalankan Makro tanpa Shortcut



1. Klik tab Developer, pada grup Code klik
2. Maka akan muncul kotak dialog seperti pada gambar 1.7
3. Pilih Run.




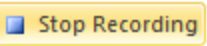
Gambar 1.7 Kotak Dialog Macro.

Latihan 3: Menghitung penjualan

1. Ketikkanlah data yang terdapat pada gambar 1.8

	A	B	C	D	E
1	DATA PEMBAYARAN TOKO KARTUN				
2	PER JANUARI 2010				
3					
4	NO	NAMA	FEE	BONUS	TOTAL FEE
5	1	DESPICABLE ME	200000	2%	
6	2	SUPERMAN	300000	3%	
7	3	SON GO KU	200000	2%	
8	4	GATOT KACA	500000	5%	
9	5	AQUAMAN	100000	1%	
10	6	DAPHNE	300000	3%	
11	7	SHAGGY	200000	2%	
12	8	JHONY QUEST	100000	1%	
13	9	MEGAMIND	300000	3%	
14	10	BATMAN	200000	2%	
15					

Gambar 1.8 Contoh data pembayaran

- Pilih tab Developer, di dalam group Code klik tombol **Record Macro** .
- Tulis nama macro Hitung
- Pada kotak Shortcut Key isi dengan huruf A kapital.
- Apabila setelah selesai mengatur semua pada kotak dialog **Record Macro**, kemudian klik tombol **OK** pada kotak dialog tersebut.
- Setelah itu akan muncul kotak dialog .
- Ketik formula sebagai berikut : $=C5+(C5*D5)$ pada sel E5
- Tekan tombol Enter dan akhiri perkaman dengan cara mengklik tombol **Stop** pada kotak dialog **Stop Recording**

Langkah memformat sel :

- Pilihlah salah satu sel pada lembar kerja. Contohnya sel G7
- Aktifkan jendela Record Macro
- Pada jendela Record Macro isilah kotak teks Macro name dengan **Format_Cell** dan shortcut key dengan huruf B, kemudian OK.
- Klik **Format**, **Cells** untuk menampilkan kotak dialog **Format Cells**.
- Pada tab **Number** pilihlah Currency untuk memformat nilai angka menjadi format mata uang.
- Selanjutnya tab **Alignment**, pada ComboBox Horizontal pilihlah Center.
- Aktifkan tab **Font**, pada Listbox Font pilihlah Font Times New Roman dan pada Combobox Color pilihlah warna merah.
- Aktifkan tab **patterns** dan pilihlah warna kuning.

9. Selanjutnya klik OK pada kotak dialog Format Cells dan hentikan proses perekaman dengan cara mengklik tombol **Stop** pada kotak dialog **stop Recording**.

10. Klik tombol **Undo** (atau **Ctrl+Z**) untuk membatalkan perubahan didalam worksheet.

Menjalankan macro Hitung:


1. Pastikan berada pada sel E6, lakukan perintah Shortcut (Ctrl+Shift+A) pada area tersebut.
2. Kemudian lakukan perintah shortcut pada sel E7 sampai E14.

Menjalankan macro memformat sel :

1. Seleksi sel E5 sampai E4, kemudian lakukan perintah shortcut (Ctrl+Shift+B) untuk memformat sel yang terseleksi. Apabila hasilnya sama dengan gambar 1.10, maka proses pembuatan macro yang kedua sukses.

	A	B	C	D	E
1	DATA PEMBAYARAN TOKO KARTUN				
2	PER JANUARI 2010				
3					
4	NO	NAMA	FEE	BONUS	TOTAL FEE
5	1	DESPICABLE ME	200000	2%	Rp204,000.00
6	2	SUPERMAN	300000	3%	Rp309,000.00
7	3	SON GO KU	200000	2%	Rp204,000.00
8	4	GATOT KACA	500000	5%	Rp525,000.00
9	5	AQUAMAN	100000	1%	Rp101,000.00
10	6	DAPHNE	300000	3%	Rp309,000.00
11	7	SHAGGY	200000	2%	Rp204,000.00
12	8	JHONY QUEST	100000	1%	Rp101,000.00
13	9	MEGAMIND	300000	3%	Rp309,000.00
14	10	BATMAN	200000	2%	Rp204,000.00
15					

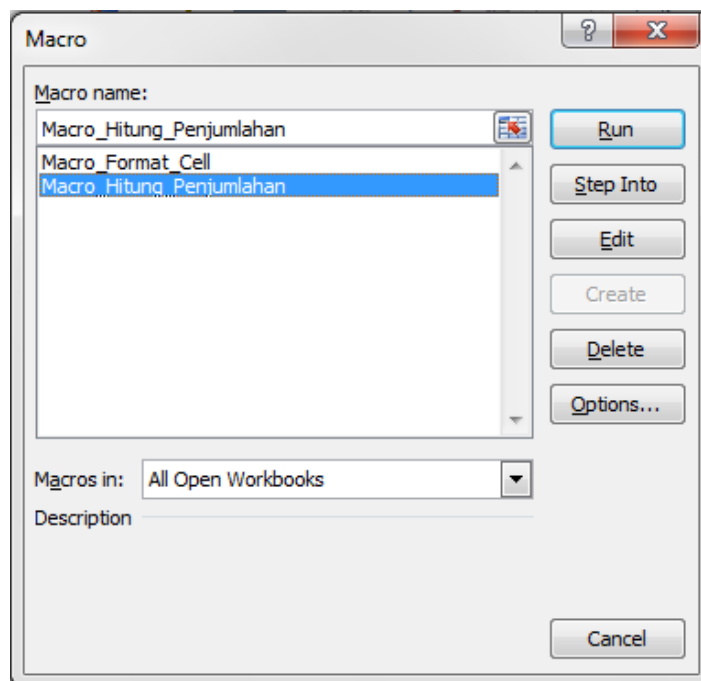
Gambar 1.10 Hasil pemformatan menggunakan shortcut

2. Tidak Menggunakan Shortcut , masih menggunakan data tabel sebelumnya.
 - 1) Pilih tab Developer, di dalam group Code klik tombol **Record Macro**  ataupun dapat menggunakan shortcut Alt+F8 untuk menampilkan kotak dialog Macro.

	B	C	D	E
1	DATA PEMBAYARAN TOKO KARTUN			
2	PER JANUARI 2010			
3				
4	NO	NAMA	FEE	BONUS
5	1	DESPICABLE ME	200000	2%
6	2	SUPERMAN	300000	3%
7	3	SON GO KU	200000	2%
8	4	GATOT KACA	500000	5%
9	5	AQUAMAN	100000	1%
10	6	DAPHNE	300000	3%
11	7	SHAGGY	200000	2%
12	8	JHONY QUEST	100000	1%
13	9	MEGAMIND	300000	3%
14	10	BATMAN	200000	2%

Gambar 1.11 Data pembayaran.

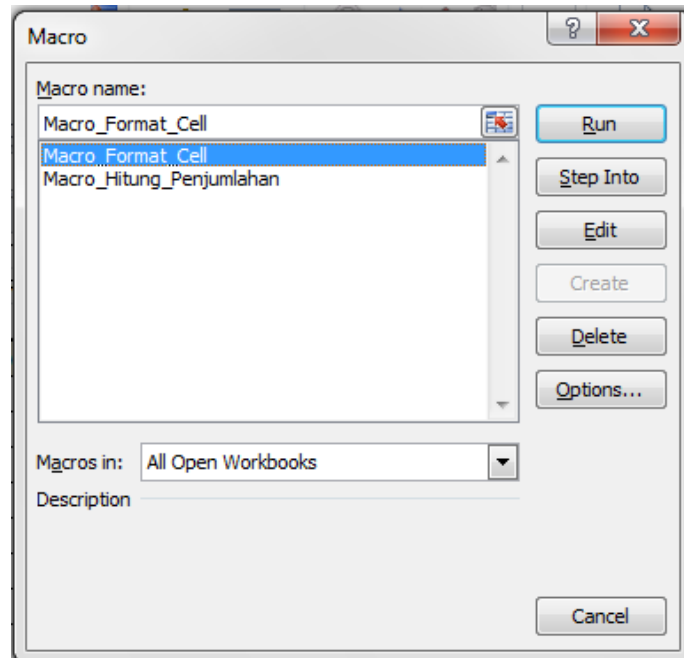
- 2) Di dalam kotak dialog Macro terdapat dua perintah macro yang telah dibuat, pilihlah makro_Hitung.



Gambar 1.12 Menjalankan Macro_Hitung_Penjumlahan

- 3) Sebelum menjalankan dengan klik Run pastikan kembali bahwa sel yang pilih adalah sel G5.


- 4) Apabila sel yang dipilih sudah benar, klik tombol Run.
- 5) Ulangi langkah 1 sampai ke 4 sampai seluruh kolom Total terisi semua.
- 6) Kemudian jalankan Macro_Format_Cell dengan cara yang sama, hanya kotak dialog

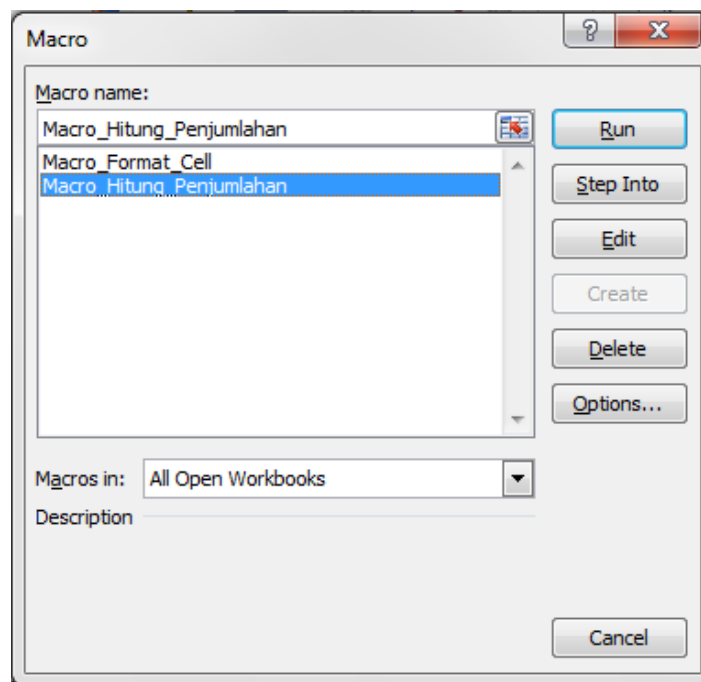


Gambar 1.13 Menjalankan Macro_Format_Cell

Mengedit Listing Macro

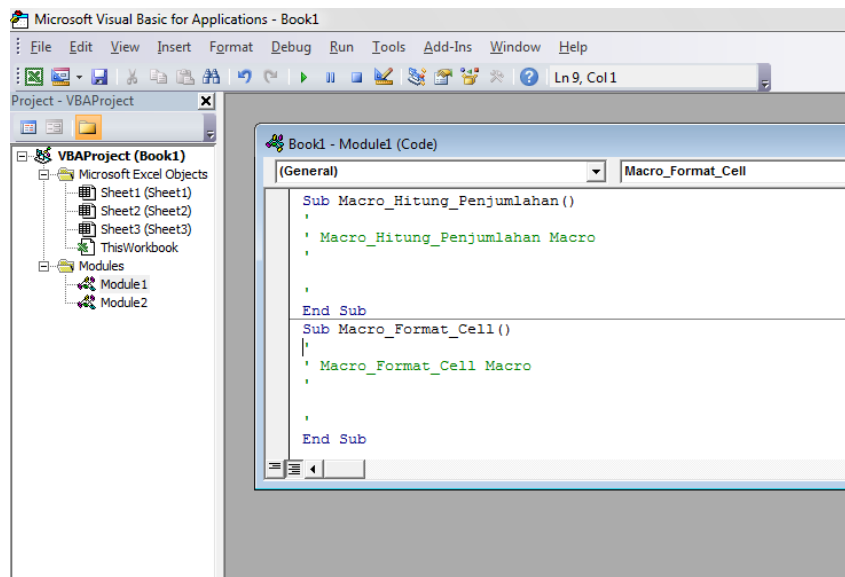
Macro yang kita buat tersimpan didalam sebuah modul, didalam modul tersebut kita dapat memperbaiki semua langkah – langkah yang dilakukan sebelumnya, mungkin dianggap salah atau tidak perlu, langkah- langkah pengeditan macro dibawah ini:

1. Pilih tab Developer, di dalam group Code klik tombol **Record Macro**  atau menggunakan shortcut Alt+F8 untuk menampilkan kotak dialog macro.
2. Didalam Kotak dialog Macro pilihlah Macro_Hitung_Penjumlahan , kemudian klik tombol Edit.



Gambar 1.14 Mengedit Macro_Hitung_Penjumlahan

3. Setelah mengklik tombol Edit pada kotak dialog macro , maka secara otomatis Jendela Microsoft Visual Basic akan tampil.



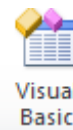
Gambar 1. 15 Jendela Visual Basic Editor.

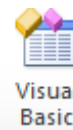
1.2 VBE (Visual Basic Editor)

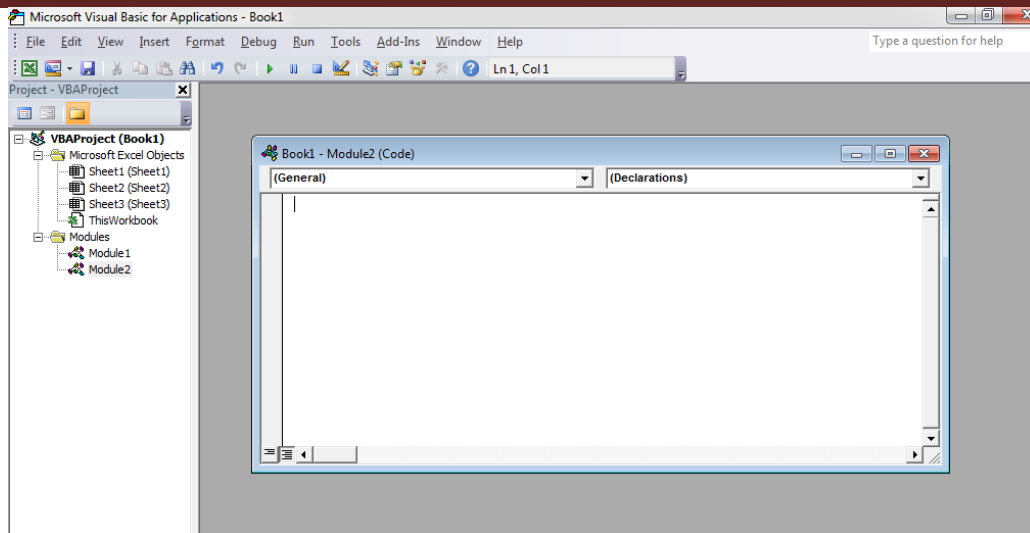
Jika kita telah terbiasa menggunakan Microsoft Excel, maka kita familiar dengan tampilan Spreadsheet dan tidak begitu familiar dengan VBE. Prosedure VBA (macro) dapat dikembangkan didalam Visual Basic Editor (VBE) menjadi bagian dari workbook dimana VBA dikembangkan dan ketika workbook disimpan, semua komponen VBA(makro, modul, Userform) disimpan pada waktu yang bersamaan. Visual Basic Editor (VBE) merupakan aplikasi yang user Friendly.

Untuk menuliskan suatu modul VBA kita perlu membuka sebuah lembar VBA atau Visual Basic Editor. Langkah – langkah nya:

1. Bukalah jendela Microsoft Office 2003.



2. Klik tab Developer, pada group Code klik icon  Atau dengan Shortcut **Alt+F11**.



Gambar 1.16 Jendela VBE

3. Dari VBE kita dapat pindah ke Excel dengan mengklik pada icon Excel yang terletak diatas kiri

Pada tampilan layar kita akan menjumpai susunan menu yang berbeda dari lembar kerja Excel, karena menu toolbar yang tersedia ini bukan lagi untuk diaplikasikan pada lembar kerja melainkan untuk fasilitas yang ada pada VBA itu sendiri. Komponen pada layar VBA editor diantaranya terdiri atas :

Toolbar Utama



Menampilkan Lembar Kerja Excel



Menyimpan File



Memotong Objek Terpilih



Menyalin Teks atau objek terpilih



Merekatkan Objek dari Clipboard ke lokasi Crusor



Mencari



Membatalkan perintah terakhir



Membatalkan pembatalan terakhir



Menjalankan sub/user form



Break (menghentikan program yang sedang dieksekusi)



Reset (kembali ke kondisi semula)



Design Mode



Project Explorer



Properties



Object Browser



Toolbox



Menampilkan bantuan untuk VBA



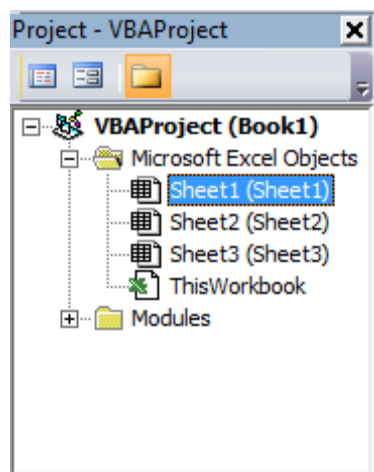
Insert user form digunakan untuk menambahkan Userform, module, class module, procedure.

Project Explorer

Biasanya terletak dibagian kiri atas bidang layar, adalah sebuah window yang berisi komponen – komponen dari suatu project.

Komponen tersebut dapat berupa:

- Sheet –sheet yang ada (beserta nama sheetnya)
- Microsoft Excel Objects



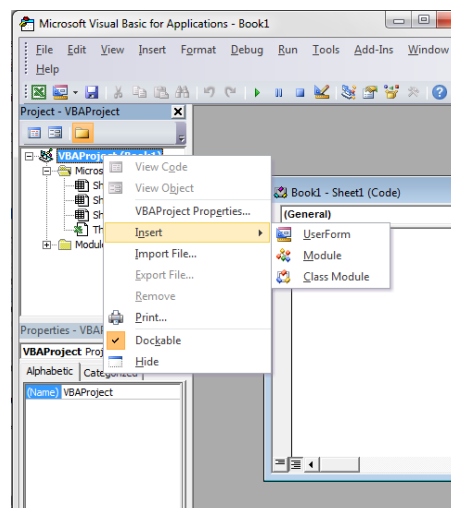
Gambar 1.18 Panel Project Explorer pada layar VBA editor

Jendela VBA project menunjukkan kepada semua workbook yang sedang terbuka.

Sedangkan pada saat bekerja, kita dapat menambahkan komponen berikut ini:

- Forms
- Modules
- Class Modules

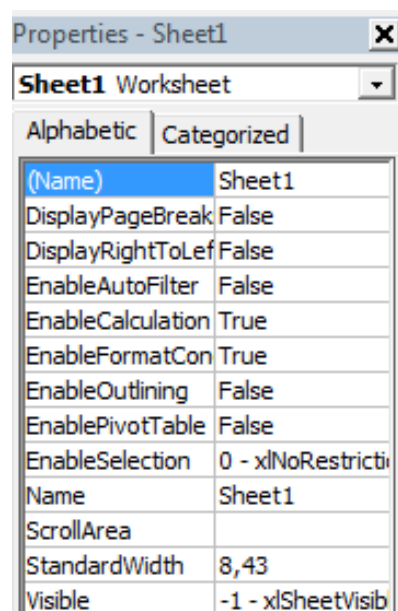
Dengan cara menggunakan menu Insert



Gambar 1.19 menambahkan Modul atau Form

Properties Windows

Jendela properties akan menunjukkan properties dari elemen yang kita pilih pada jendela VBA project. Properti ini dapat didefinisikan saat didesain atau kita dapat memodifikasi properti setelah didefinisikan. Elemen – elemen yang terdapat pada properties windows bergantung pada apa yang terpilih saat itu di jendela project.



Gambar 1.20 Jendela Properties dari Sheet 1 yang aktif

Properties yang ditampilkan dikelompokkan pada dua kategori, yang pertama sesuai susunan alfabetis sedang yang lain dengan susuna sesuai kategorinya. Kita dapat memodifikasi properties manapun dari elemen yang terdapat di dalam VBA project. Daftar

properties akan berbeda pada setiap jenis komponen yang pilih (This Workbook, worksheet, modul , atau form).

Membuat VBE pertama dalam Microsoft Excel

Latihan1

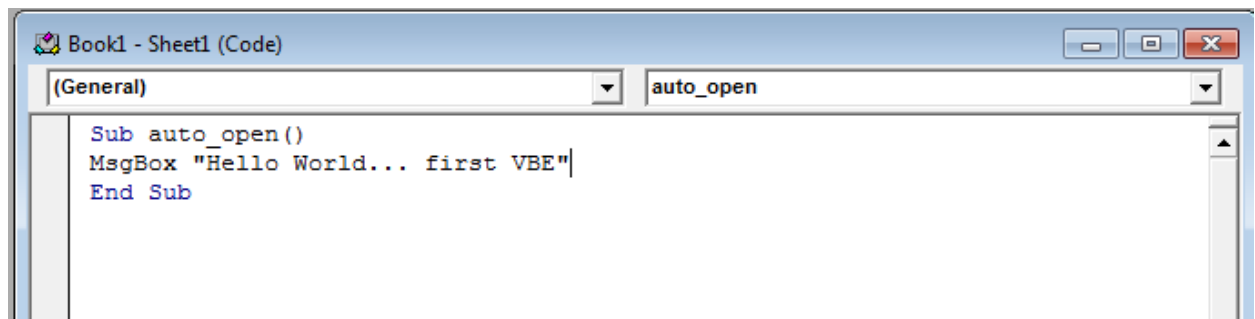
Langkah memulai VBE :

1. Aktifkan jendela Visual Absic Editor dengan klik tab Developer, pada group Code klik




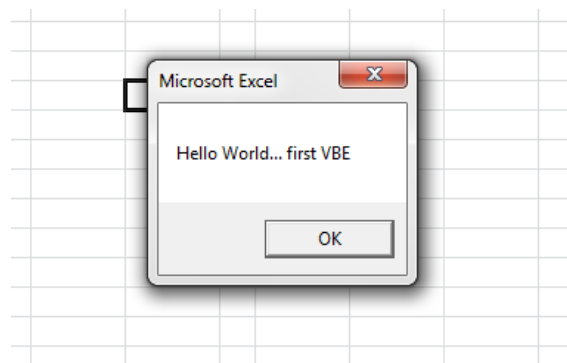
icon **Visual Basic** Atau dengan Shortcut **Alt+F11**.

2. Didalam jendela VBA project klik g pada sheet1. Jendela kode masih dalam keadaan kosong, sebab belum membuat prosedur yang berhubungan dengan Sheet 1
3. Tuliskan Kode program pada jendela kode seperti gambar 1.21



Gambar 1.21 kode program pertama pada VBE

4. Setelah selesai menuliskan maka jalankan VBE dengan klik icon  pada toolbar atau dengan menekan tombol F5 pada keyboard.
5. Maka secara otomatis menuju sheet 1 dan menampilkan message box.



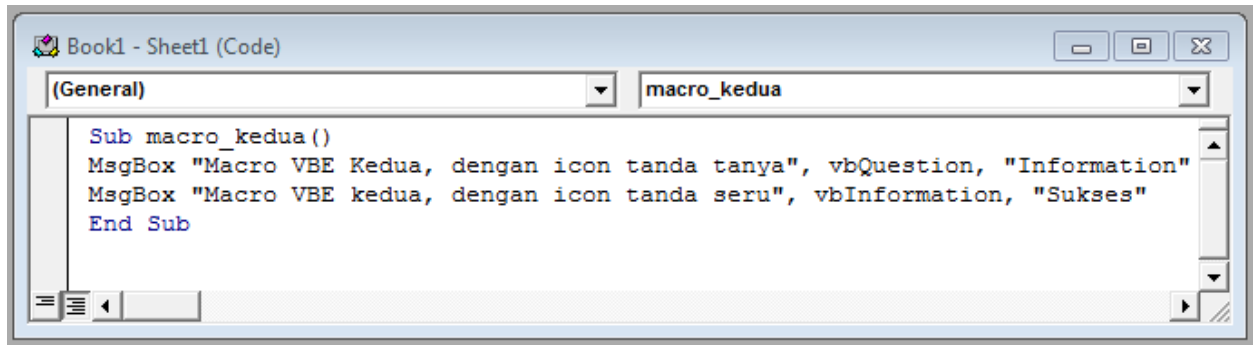
Gambar 1.22 Mesage box

6. Klik tombol OK yang terdapat pada message box, jendela VBE akan tampil kembali.
7. Jika akan mulai kembali VBE maka buka jendela VBE.

Latihan 2 : Buatlah makro VBE dan jalankan dengan perintah F8 dan t icon “?” serta t icon “!”

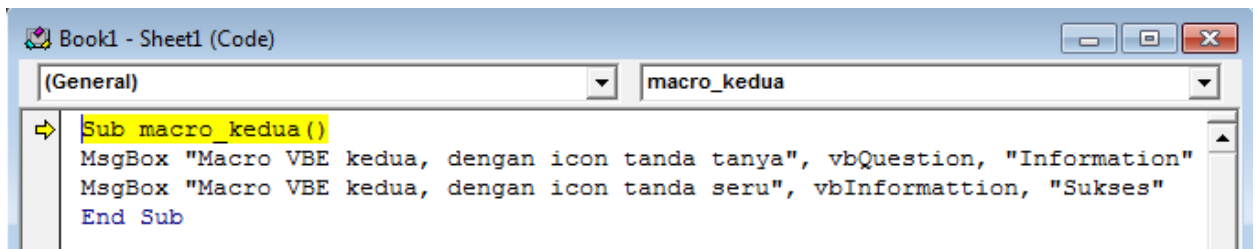
Langkah – langkahnya :

1. Buka jendela VBE, dan ketikkan langsung pada jendela VBE kode berikut ini



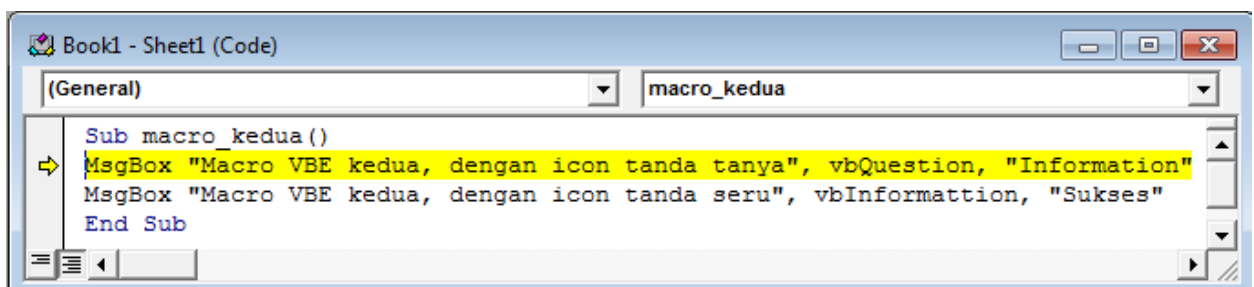
Gambar 1.23 Listing Program Icon

2. Setelah selesai menuliskan kode program, tekan tombol F8 pada keyboard , maka secara otomatis pada baris pertama akan berwarna kuning dan terdapat icon panah disebelah kiri jendela.



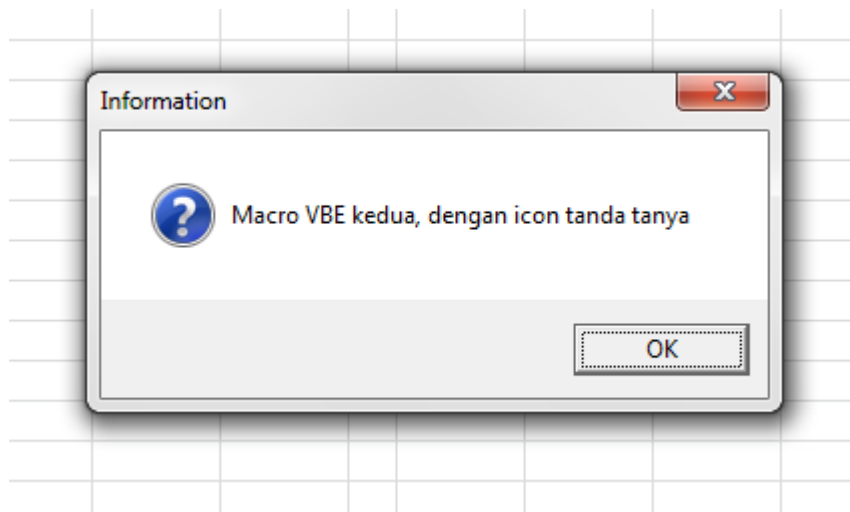
Gambar 1.24 Menekan F8(1)

3. Selanjutnya tekan tombol F8 kembali. Baris pesan yang pertama akan diseleksi dengan warna kuning, tetapi Msgbox yang pertama belum dijalankan.



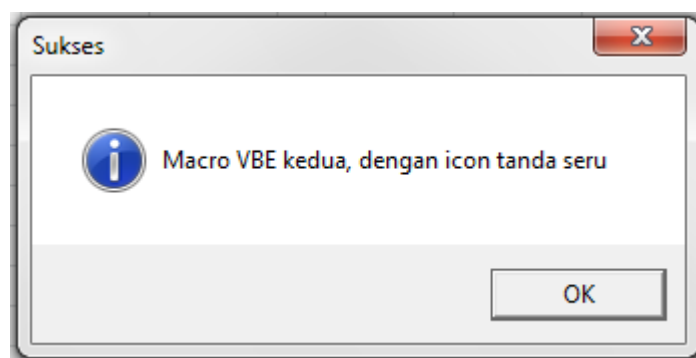
Gambar 1.25 Menekan F8(2)

4. Tekan F8, baris kedua kan diseleksi. Selanjutnya perintah MsgBox yang pertama akan dijalankan pada jendela Excel dengan menggunakan icon t tanya (?) pada kotak pesan.



Gambar 1.26 Kotak Pesan Macro_Kedua (1)

5. Tekan F8 kembali, baris terakhir akan diseleksi dan menjalankan perintah MsgBox yang kedua dengan menggunakan t seru.



Gambar 1.27 Kotak Pesan Macro_Kedua (2)

6. Selanjutnya sebagai langkah terakhir tekan kembali tombol F8 pada keyboard.
- Menjalankan Perintah macro dengan menggunakan tombol **F8** bertujuan untuk mengetahui setiap langkah perintah yang dibuat dan bisa juga untuk mengetahui letak suatu kesalahan sebuah perintah macro.

Objek Properties yang paling sering digunakan pada excel

Tabel 1.1 Object Properties

Nama Property	Type dan Artinya	Terdapat pada Object
ActiveCell	Object: sel yang aktif	Application, Window
ActiveChart	Object: Grafik yang aktif	Application, Window, Workbook

ActiveSheet	Object: lembar kerja yang aktif	Application, Window, Workbook
Count	Integer: jumlah koleksi dari Object	Semua koleksi obkect
Formula	String: Rumus yang berlaku pada suatu sel	Range
Index	Integer: Jumlah object dalam koleksi	Worksheet, Dialogsheet. Module
Name	String: nama suatu objek	Application, Workbook, Others
Path	String: disk drivemaupun path alamat letak suatu file/ object	Addln, Application, Workbook
Saved	Boolean: pilihan True atau False (ya atau tidak disave)	Workbook
Selection	Object: obyek yang terpilih saat ini	Application, Window
StatusBar	String: pesan yang tertulis pada status bar	Application
ThisWorkBook	Object: perintah yang akan menimpa workbook dalam procedure ini	Application
Type	Integer: suatu angka yang menyatakan type suatu object	Window, WorkSheet, Chart
Visible	Boolean: Ya atau tidak untuk menentukan suatu objek ditampilkan atau tidak	Range, dan lainnya
Value	Nilai sebenarnya yang ditampilkan disel	Range

1.3 Tipe Data, Variabel dan Array

Dalam VBA kita akan mempelajari variable dan constant, maka sebelum itu kita perlu mengenal berbagai jenis tipe data didalam VBA.

Tipe Data

Suatu variable dapat memberikan tipe data yang dibatasi pada data yang akan digunakan. Dimana akan membuat efesien pada listing kode. Jika tidak memakai tipe data maka semua variable adalah varian.

Tabel 1.2 Jenis Data pada Excel

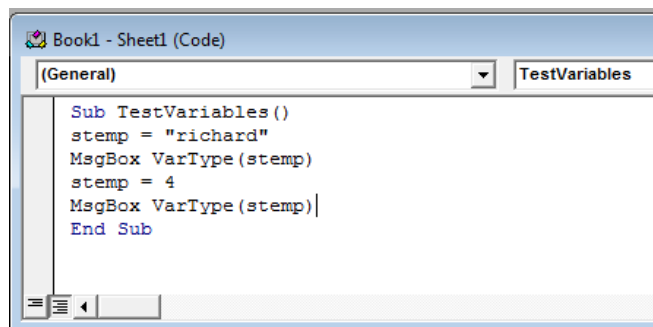
Type Data	Ukuran Dalam Byte	Uraian
Array	Sesuai dengan tipe dan jumlah bilangan	Setiap elemen range array sama dengan tipe dasar. Elemen dalam Array tidak terbatas.

	yang terdapat dalam elemen array	
Boolean	2 Byte(16 Bits)	Berisi logika True dan False.
Currency	8 Byte(64 Bits)	Bilangan Dengan nilai antara – 922,337,203,685,477,5808 hingga
Date	8 Byte(64 Bits)	Menyimpan kombinasi informasi tanggal dan waktu. Tanggal antara 1 Januari 0100 hingga 31 Desember 9999. Waktu antara 00:00:00 hingga 23:59:59
Double	8 Byte (64 bits)	Bilangan negatif antara: $-1.7976931348623 \times 10^{-324}$ hingga $-4.940656458623 \times 10^{308}$ Bilangan positif antara: $4.940656458623 \times 10^{-324}$ hingga $1.7976931348623 \times 10^{308}$
Integer	2 Byte (16 bits)	Semua bilangan antara -32.768 hingga 32.767
Long	4 Byte(32 bits)	Semua bilangan antara -2.147.483.648 hingga 2.147.483.647
Object	4 Byte(32 bits)	Digunakan untuk mengakses objek apa saja yang diperlukan oleh VBA.
Single	4 Byte(32 bits)	Bilangan negatif antara -3.402823×10^{38} hingga $-1.401298 \times 10^{-45}$ Bilangan positif antara 1.401298×10^{-45} Hingga 3.402823×10^{38} .
String	1 Byte tiap karakter	Untuk menyimpan teks.
Variant	16 Byte ditambah 1 byte setiap karakter	Tipe Variant ini menyimpan tipe data apa saja.

Variant

Variant dapat mengambil seluruh data, baik itu berupa text, angka, tanggal atau semua informasi. Bahkan dapat mengambil array. Suatu varian variable dapat berubah dengan bebas pada saat dijalankan, Kita juga dapat menggunakan fungsi **VarType** untuk mengetahui jenis tipe data dengan variant:

Latihan 1: Ketikkan kode berikut ini



Gambar 1.28 contoh Variant

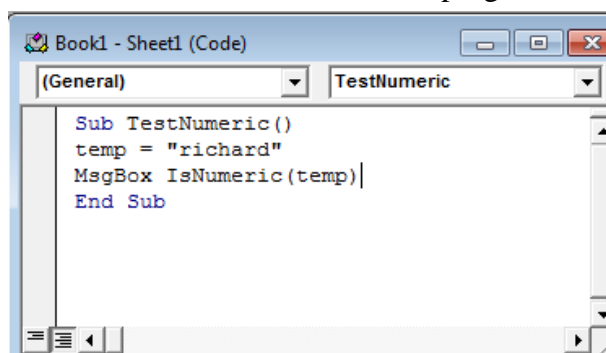
Setelah Program dijalankan maka Message Box yang pertama tampil adalah 8, menunjukkan string. Kemudian akan muncul 2 yang menunjukkan integer

Tabel 1.3 VarType

ReturnValue	Type
0	Empty
1	Null
2	Integer
3	Long
4	Single
5	Double
6	Currency
7	Date/Time
8	String

VBA selalu menggunakan variant yang efisien untuk pengambilan data. Seperti latihan 1 dimana secara otomatis data store berubah menunjukkan data selanjutnya.

Latihan 2: Buka WorkSheet baru dan ketikan kode program berikut ini



Gambar 1.29 Kode program kedua

Apakah Hasil dari program tersebut? Jelaskan kenapa!

Date/ Times

Variable variant juga terdiri dari date/times. Merupakan angka Floating point- yang berupa integer menampilkan tanggal mulai dari 31 desember 1899 dan untuk desimal menampilkan jam, menit dan detik dalam 24 jam. Contoh : 37786.75 menunjukkan 14-Juni-2003 pukul 18:00.

Empty Value

Suatu variant yang tidak memiliki variable bernilai Kosong. Ini dapat diuji dengan menggunakan fungsi IsEmpty.

```
MsgBox IsEmpty(temp)
```

Null Value

Variant dapat terdiri dari Null. Nilai Null digunakan untuk mengidentifikasi data yang tidak dikenal atau data hilang. Variable tidak bernilai null, jadi jika tidak menggunakan Null pada aplikasi, maka kita tidak perlu khawatir dengan nilai Null. Null itu bukan berarti 0. untuk memudahkan pemakaian Null maka pada kode program gunakan perintah **IsNull**. Untuk metode yang lain, statement Is Null, mungkin tidak memberikan hasil yang tepat.

Latihan 3: Buka Lembar kerja VBE yang baru, kemudian ketikkan Kode program berikut

```
Sub TestNull()  
temp=Null  
Msgbox IsNull(temp)  
End Sub
```

Apa Hasil dari Program Berikut? (**Hasil True**)

Tipe Data VBA

Ada beberapa jumlah tipe data yang dapat digunakan dalam VBA

Tabel 1.4 Tipe data (Variabel) pada VBA

Nama	Keterangan	Type-Declaration Character	Range
Integer	2-byte integer	%	–32,768 sampai 32,767
Long	4-byte integer	&	–2,147,483,648 sampai 2,147,438,647
Single	4-byte floating point number	!	–3.402823E38 sampai 1.401298E-45 (negative values) 1.401298E-45 sampai 3.402823E38 (positive values)
Double	8-byte floating point number	#	–1.79769313486232D308 sampai –4.94065645841247D-324 (negative values) 4.94065645841247D-324 sampai

			1.79769313486232D308 (positive values)
Currency	8-byte number with fixed decimal point	@	–922337203685477.5808 to 922337203685477.5807
Fixed Length String	String of characters— fixed length	\$	0 to approximately 65,500 characters
Variable Length String	String of characters— variable length	\$	0 to approximately 2 billion characters
Variant	Date/Time, floating point number or string	None	Date Values: January 1, 0000 to December 31, 9999; numeric values: same range as double; string values: same range as string

Type Numerics

Jika kita akan memakai angka, dan mendekalrasikan variable dengan integer atau long tergantung dari kapasitas. Operator matematika memiliki memori yang cepat tergantung dari panjang ukuran tipe.

Jika kita menggunakan angka maka kita pakai Single, Double tau Currency. Currency (Decimal Point) dengan kapasitas 4 digit untuk desimal point kanan dan 15 digit untuk desimal point kiri. Floating Point(single atau Double) memiliki kapasitas yang besar

Type String

Jika Variable terdiri dari text, maka kita mendeklarasikan dengan **String**:

```
Dim temp as String
```

Variabel

Variabel adalah sebuah nama yang diberikan pada suatu area dalam memory komputer untuk menyimpan data yang akan diproses dalam suatu operasi.

Nama Variabel harus didefinisikan terlebih dahulu sebelum dapat digunakan. Masing – masing variabel mempunyai jenis spesifikasi yang membedakan berapa banyak memori yang digunakan dalam mengoperasikan suatu data variabel. Bentuk umum: **Dim** nama_variabe [AsType]

Cara penamaan variabel:

- Variabel harus diawali dengan Letter (Alfabhet)
- Setelah Letter pertama, dapat terdiri atas kombinasiapa saja dari letter, bilangan, dan t baca garis bawah(_)
- Nama variabel tdak boleh diawali dengan angka

- Tidak boleh mengandung karakter – karakter khusus, seperti *,-,+,#,@ spasi dan sebagainya.
- Nama Variabel tidak boleh melebihi 255 karakter.
- Nama variabel harus unik (tidak kembar) dalam suatu scope dimana tidak didefinisikan
- Nama variabel tidak boleh sama dengan perintah – perintah VBA misalnya Dim

AsType : Bersifat opsional yang mendefinisikan tipe data pada variabel yang dideklarasikan

Contoh : `Dim MyInteger as Integer`

Variabel dapat dideklarasikan menjadi tiga macam :

a. Variabel Local

Variabel Local menggunakan **Dim**, **Static** atau **ReDim**(digunakan untuk array) digunakan untuk mendeklarasikan variabel dalam prosedur. Beberapa prosedur menggunakan **temp**, sebab variabel lokal dalam program berdiri sendiri. Statement **Dim** digunakan selama prosedur itu dijalankan. Statement **Static** digunakan selama aplikasi itu dipakai.

Contoh : `Dim TempVal`

`Static TempVal`

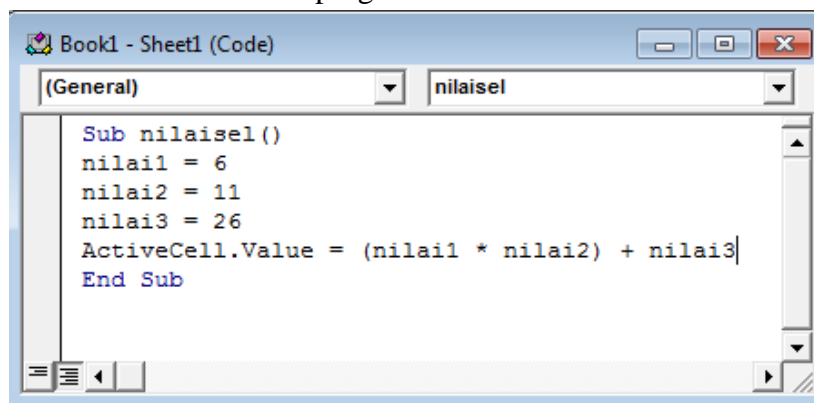
Penggunaan **ReDim** harus didefinisikan dengan Array. `Dim A(3)` membuat array sederhana dengan 4 elemen(0-3). **ReDim A(10)** berarti membuat 11 elemen array. **ReDim** juga digunakan untuk untuk menampung elemen array.

b. Variabel Modul

Variabel Modul mendeklarasikan modul yang umum. Dipakai untuk semua prosedur yang memakai modul

c. Variabel Global

Latihan 1: Buka VBE dan ketikkan kode program berikut :



Gambar 1.30 Listing Program Variabel

Apa hasil dari program setelah dijalankan?

Array

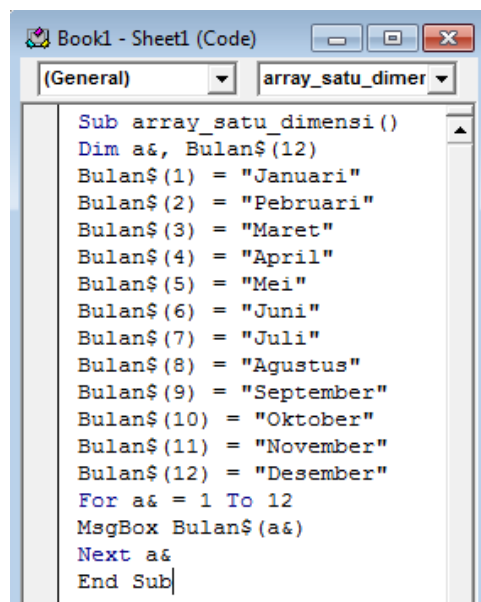
Mendeklarasikan sebuah array sama halnya dengan kita mendeklarasikan sebuah variabel reguler. Kita juga dapat menetapkan tipe array tersebut dan menetapkan nilai array didalam t kurung yang mengikuti nama array. Didalam VBA ada dua kategoridari *Array* :

Fixed-size Array , Ukuran Array selalu ditetapkan

Array dapat ditampilkan lebih dari satu dimensi

- Array satu dimensi

Latihan 1: Bukalah jendela VBA dan ketikkan kode program berikut ini:



```

Sub array_satu_dimensi()
Dim a&, Bulan$(12)
Bulan$(1) = "Januari"
Bulan$(2) = "Pebruari"
Bulan$(3) = "Maret"
Bulan$(4) = "April"
Bulan$(5) = "Mei"
Bulan$(6) = "Juni"
Bulan$(7) = "Juli"
Bulan$(8) = "Agustus"
Bulan$(9) = "September"
Bulan$(10) = "Oktober"
Bulan$(11) = "November"
Bulan$(12) = "Desember"
For a& = 1 To 12
MsgBox Bulan$(a&)
Next a&
End Sub

```

Gambar 1.31 Pemakaian array

Jalankan kode program berikut. Apa hasil dari program berikut?

- Array Multi dimensi
- Mengetahui ukuran suatu Array

Ubound berfungsi untuk menemukan batas tertinggi dari suatu dimensi array. Sedangkan **Lbound** berfungsi untuk menentukan batas yang lebih rendah dari suatu dimensi array.

Array Dinamis- ukuran array dapat ditentukan / diubah pada saat dijalankan.

1.4 Operator

Operator menampilkan fungsi matematik, fungsi perbandingan atau logika antara dua angka atau numerik yang terdapat pada program. Contoh sederhana operator adalah

plus (+) atau minus (-). Operator membatasi perhitungan. Terdapat macam – macam kategori operator seperti tabel dibawah ini :

Tabel 1.5 Opertor dalam VBA

Arithmetic	Comparison	Logical
Exponentiation (^)	Equality (=)	Not
Negation (-)	Inequality (<>)	And
Multiplication and division (*, /)	Less than (<)	Or
Integer division (\)	Greater than (>)	Xor
Modulo arithmetic (Mod)	Less than or equal to (<=)	Eqv
Addition and subtraction (+, -)	Greater than or equ al to (>=)	Imp
String concatenation (&)	Like / Is	

Operator matematik

Operator matematik yang dgunakan yaitu plus (+), minus (–), multiply (*), dan divide (/).

- **Operator (*)**

Operator ini akan mengalikan 2 angka

`MsgBox 6 * 3`

Hasilnya adalah 18

Angka yang dikalikan dapat berupa numerik dan hasil dari perkalian dapat berupa integer, long, single, double, currency. Jika Operand bernilai Null maka hasilnya akan null

- **Operator (+)**

Operator ini akan menjumlahkan 2 angka atau ekspresi bersamaan

`MsgBox 4 + 2`

Maka hasilnya adalah 6

Operator ini dapat menjumlahkan angka atau menggabungkan string. Angka yang dikalikan dapat berupa numerik dan hasil dari perkalian dapat berupa integer, long, single, double, currency. Jika Operand bernilai Null maka hasilnya akan null.

Aturan dalam menjumlahkan dan menggabungkan:

- Dua operand berupa numerik.
- Untuk penggabungan kedua oerand berupa string.
- Penjumlahan dapat juga jika salah satu operand numerik dan yang lainnya adalah variant (not null).
- Penggabungan jika salah satu operand berupa string dan yang lainnya varian(not null).

- Operator (-)

Mengurangkan satu angka dengan angka yang lain atau menunjukkan nilai negative.

Contoh dibawah ini akan menghasilkan nilai 2:

MsgBox 6-4

Angka yang dikurangkan dapat berupa numerik dan hasil dari perkalian dapat berupa integer, long, single, double, currency. Jika Operand bernilai Null maka hasilnya akan null.

- Operator (/)

Membagi dua angka dan memberikan hasil dalam bentuk Floating point. Angka dapat berupa numerik dan hasil dari perkalian dapat berupa integer, long, single, double, currency. Jika Operand bernilai Null maka hasilnya akan null.

- Operator (\)

Membagi dua angka dan memberikan hasil dalam bentuk integer. Angka yang dipakai berupa numerik. Hasil dari tipe data adalah integer atau long. Jika salah satu operand Null maka hasilnya akan Null.

- Operator (^)

Akan mengangkat angka. Operand dapat berupa ekspresi numeric, Operand dapat berupa negative jika nilai eksponen berupa integer.

- Operator Mod

Merupakan hasil akhir atau sisa pembagian.

Operator pembandingan

Operator pembandingan membandingkan dua ekspresi.

MsgBox 3 >1

Hasilnya adalah True sebab 3 lebih besar dari 1

Operator pembandingan memberikan hasil berupa Boolean yaitu True atau False juga kan bernilai Null jika operand mengandung nilai null.

Tabel 1.6 Operator Pembandingan

Operator	Meaning
<	Less than
<=	Less than or equal to

>	Greater than
>=	Greater than or equal to
=	Equal to
<>	Not equal to

Operator Concatenation (pengabungan)

Menggabungkan dua operand

```
MsgBox "Richard " & "Shepherd"
```

Memberikan hasil “Richard Shepherd.”

Dapat juga menggabungkan angka dengan string, tetapi hasil yang akan keluar adalah string.

Hasil dibawah ini akan memberikan hasil “12 Twelve” :

```
Msgbox 12 & " Twelve"
```

Hasil untuk program dibawah ini adalah 34, berupa string bukan angka

```
Msgbox 3 & 4
```

Operator Logika

Operator logika terdiri dari:

- Operator And

Hasil dari operator ini akan bernilai benar jika nilai keduanya bernilai benar

- Operator Not

Operator Not menampilkan logika Not untuk ekspresi angka. Jika nilai bit 1 dibandingkan dengan 0 maka hasilnya adalah 1 begitu juga sebaliknya.

```
MsgBox Not (2 = 3)
```

Hasil dari program sekarang adalah True sebab 2 tidak sama dengan 3.

- Operator Or

Akan bernilai Benar jika salah satu atau keduanya bernilai salah.

- Operator Xor

1.5 Range dan Worksheet

Range

Range adalah properti yang paling penting dalam object range. Perintah Range sering sekali digunakan baik itu sebagai fungsi untuk menyeleksi, berpindah antara satu sel dengan sel yang lain atau fungsi yang lainnya. Agar lebih mudah dipahami bagaimana kegunaan dari properti Range lakukan latihan berikut:

Latihan 1:

1. Bukalah sebuah buku kerja baru, kemudian buatlah sebuah modul baru pada lembar Visual basic sebagai berikut:

```
Sub testrange() Range("A1",
"D3").Select Range(ActiveCell,
"G2").Select Range("B2:C4").Select
Range("B3:G5").Name = "RangeSatu"
Range("RangeSatu").Select
Range("B2").Select
ActiveCell.Range("B2").Select
Range("RangeSatu").Range("A1").Select
End Sub
```

2. Letakkan crusor pada prosedur tersebut. Tekan tombol F8 tiga kali untuk menjalankan perintah baris

```
Range("A1", "D3").Select
```

	A	B	C	D
1				
2				
3				
4				

3. Aktifkan lembar kerja Excel dengan mengklik icon View Microsoft Excel, sel yang saat ini terpilih adalah range “A1 hingga D3”. Sel yang aktif adalah “A1” sebagai batas pojok kiri atas dari range
4. Tekan F8 lagi untuk menjalankan baris . Oleh karena saat ini sel aktifnya masih “A1”, maka sel tersebut tetap sebagai batas range kiri atas, sedang batas kanan bawahnya adalah “G2”

```
Range(ActiveCell, "G2").Select
```

	A	B	C	D	E	F	G
1							
2							
3							

5. Tekan F8 lagi untuk menjalankan baris . Oleh karena saat ini sel aktifnya masih “A1”, maka sel tersebut tetap sebagai batas range kiri atas, sedang batas kanan bawahnya adalah “G2”

```
Range (ActiveCell, "G2") .Select
```

	A	B	C	D
1				
2				
3				
4				
5				

6. Tekan F8 dua kali untuk menjalankan baris

```
Range ("B3:G5") .Name = "RangeSatu"
```

```
Range ("RangeSatu") .Select
```

Maksud dari baris pertama adalah memberi nama range “B3” sampai “G5” dengan “RangeSatu”. Sedangkan perintah baris yang kedua un utk mengaktifkan “RangeSatu” tersebut.

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							

7. Tekan F8 kembali untuk menjalankan baris:

```
Range ("B2") .Select
```

Yaitu perintah untuk memilih sel B2

	A	B
1		
2		
3		
4		
5		

8. Tekan F8 untuk menjalankan baris:

```
ActiveCell.Range ("B2") .Select
```

Range yang dipilih adalah sel “C3”. Sel “C3” adalah kolom kedua dan baris kedua dari ActiveCell.

9. Tekan F8 untuk menjalankan baris

```
Range ("RangeSatu") .Range ("A1") .Select
```

“RangeSatu” memiliki area “B3” sampai “G5. Jadi range “A1” dari “RangeSatu” adalah “B3”.

10. Tekan F8 sekali lagi untuk menghentikan makro

Worksheet

Setiap buku kerja (sheet) adalah worksheet object, sedangkan worksheet object adalah bagian dari worksheet – worksheet yang ada dalam sebuah workbook. Untuk menunjukkan pada suatu worksheet tertentu, kita dapat menggunakan method worksheet. Bentuk Penulisan:

Object.Worksheet (index)

Object adalah sebutan pada objek workbook yang mengandung worksheet yang dimaksud. *Index* dapat berupa nomer worksheet yang ingin ditunjukkan angka 1 akan dianggap sebagai worksheet pertama, 2 dianggap worksheet kedua, demikian seterusnya.

Worksheet nama dari worksheet yang dimaksud. Nama ini adalah nama yang tertera pada tab worksheet.

1.6 Message Box dan Input Box

MessageBox dan InputBox adalah perintah untuk menampilkan sebuah kotak pesan dan kotak input pada VBA. Kadang – kadang perintah MessageBox dan InputBox diperlukan untuk mengambil sebuah informasi atau menampilkan informasi kepada pemakai. Syntax dalam penulisan perintah MessageBox :

MsgBox(pesan[, tombol atau icon][, judul kotak pesan])

Syntax dalam penulisan perintah InputBox:

InputBox(pesan[, judul kotak pesan][, default])

Dalam penulisan perintah MessageBox kita dapat menambahkan icon atau tombol pada kotak pesan yang dibuat.

Constant	Nilai	Fungsi
vbOkOnly	0	Menampilkan tombol OK saja
vbOkCancel	1	Menampilkan tombol OK dan Cancel
vbAbortRetryIgnore	2	Menampilkan tombol Abort, Retry, dan Ignore
vbYesNoCancel	3	Menampilkan tombol Yes, No, Cancel
vbYesNo	4	Menampilkan tombol Yes dan No
vbRetryCancel	5	Menampilkan tombol Retry dan Cancel
vbCritical	16	Menampilkan icon Pesan Kritis
vbQuestion	32	Menampilkan icon Peringatan Query

vbExclamation	48	Menampilkan icon Pesan
vbInformation	64	Menampilkan icon pesan Informasi
vbMsgBoxRight	524288	Memindahkan posisi teks judul pesan
vbMsgBoxRtlReading	1048576	Memutar posisi teks judul dan tombol silang pada kotak pesan

Membuat Perintah MessageBox

Latihan: Kerjakan kode program dibawah ini untuk setiap program akan berbeda sesuai dengan nama variabel.

```
Private Sub Workbook_Open()
    MsgBox "Hello"
End Sub
```

```
Sub MyMessage()
    MsgBox "This macro is created by Vina"
    MsgBox "The icon is different", vbInformation
    MsgBox "The top title is different", vbExclamation, "Vina Tip's"
End Sub
```

```
Sub MessageBox1()
    MsgBox "Kotak pesan pertama!"
End sub
```

```
Sub MessageBox2()
    Dim strMessage As String
    strMessage = "Kalau pesan yang gunakan terlalu panjang"
    & vbCrLf & "gunakan jenis kotak pesan ini!"
    MsgBox strMessage
End sub
```

```
Sub MessageBox3()
    Dim strAnswer As VbMsgBoxResult

    strAnswer = MsgBox("Apakah akan mewarnai kolom yang aktif?", _
vbQuestion + vbYesNo, "Kotak pesan dengan tombol Yes/No!")

    if strAnswer = vbYes then
        selection.interior.colorindex = 3
    end if
end sub
```

```
Sub TwoLines()
    MsgBox "Line 1" & vbCrLf & "Line 2"
End Sub
```


Membuat Perintah InputBox

Latihan: Kerjakan kode program dibawah ini untuk setiap program akan berbeda sesuai dengan nama variabel.

```
Sub GetInput()  
    Dim MyInput  
    MyInput = InputBox("Enter your name")  
    MsgBox ("Hello ") & MyInput  
End Sub  
  
Sub InsertRow()  
    Dim Rng  
    Rng = InputBox("Enter number of rows required.")  
    Range(ActiveCell.Offset(0, 0), ActiveCell.Offset(Rng - 1,  
    0)).Select  
    Selection.EntireRow.Insert  
End Sub  
  
Sub InputBox()  
    Dim strResponse As String  
    strResponse = InputBox("Silahkan masukkan nama  
    !",_"Nama ")  
  
    If strResponse = "" then  
        MsgBox " harus memasukkan nama !"  
        Exit sub  
    End if  
    Range("A1").value = strResponse
```

Jalankan Kode Program diatas satu persatu dan Tampilkan Hasilnya.



