

BAB 2

STATEMENT-STATEMENT PADA VBA EXCEL

Objektif :

- Mengetahui Statement Bersyarat pada VB Excel
- Mengetahui Jenis – Jenis Statement Bersyarat
- Mengetahui Statement Pengulangan
- Mengetahui Jenis – Jenis Statement Bersyarat

Ketika kita membuat program, penting untuk mengetahui bagaimana penggunaan keputusan (decisions) dan perulangan. Perulangan adalah proses pengambilan beberapa instruksi sampai kondisi itu berakhir.

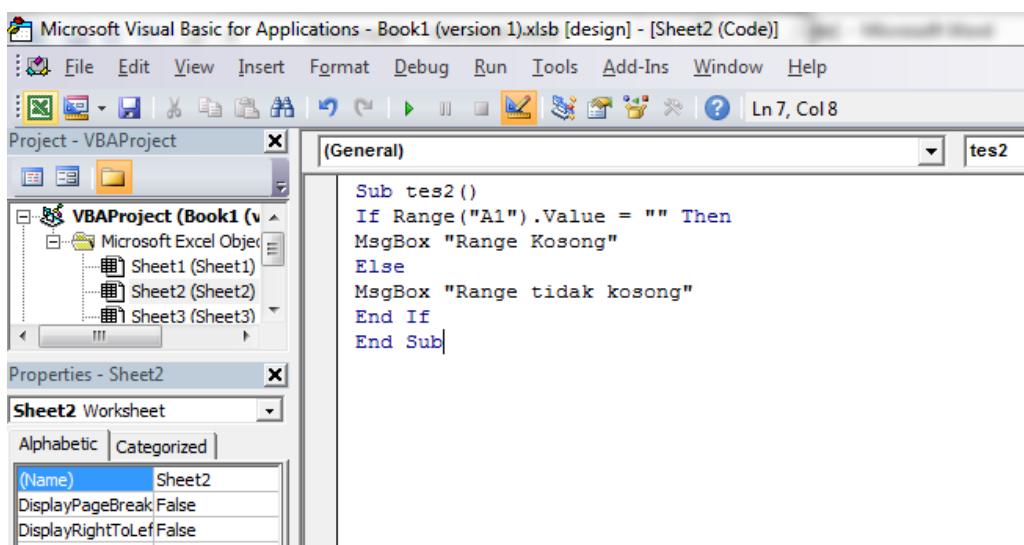
Program memakai statement keputusan berdasarkan parameter yang memudahkan program untuk menjalankannya. Perulangan terjadi setiap kita melakukan aktifitas. Ketika kita makan, akan melakukan kegiatan yang sama

2.1 Pernyataan Bersyarat

Ada berbagai format untuk membuat perintah **If**. Contohnya syntax dalam satu baris, syntax ini sering digunakan ketika hanya satu kondisi yang akan dieksekusi. Pemakaian syntax dalam satu baris tidak perlu menggunakan perintah penutup **End if**.

If kondisi Then statement1 [Else statement2]

Latihan1 : Bukalah Excel, kemudian tampilkan jendela VBE setelah itu ketik program dibawah ini.



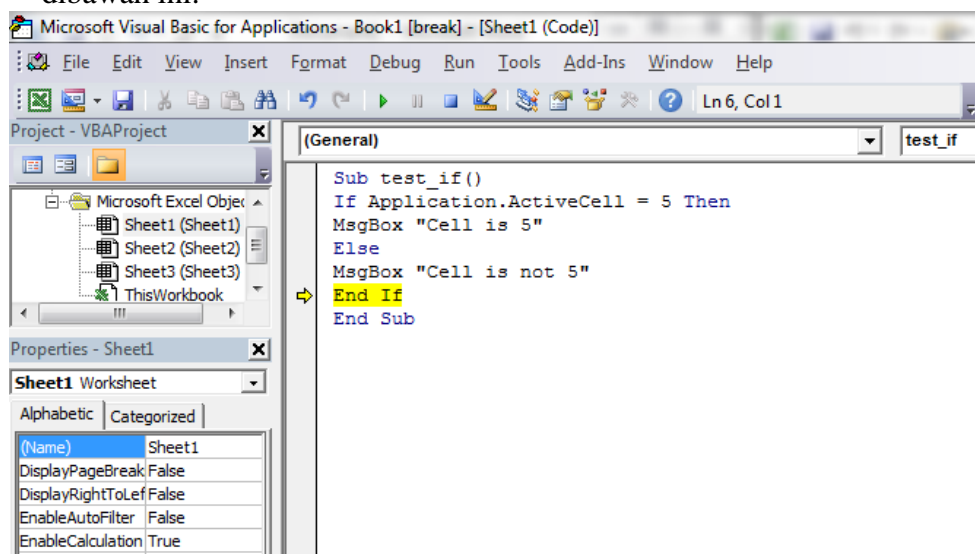
Gambar 2.1 Kode Statement kondisi

Jalankan program diatas dan apa hasilnya?

Apabila kita mempunyai banyak kondisi maka kita menggunakan syntax multi- linen untuk memudahkan kita membaca suatu kondisi. Bentuk statementnya sebagai berikut:

```
If kondisi1 Then  
    Statement1  
[ElseIf kondisi2 Then  
    statement2]  
[Else  
    statement3]  
EndIf
```

Latihan2 : Bukalah Excel, kemudian tampilkan jendela VBE setelah itu ketk program dibawah ini.



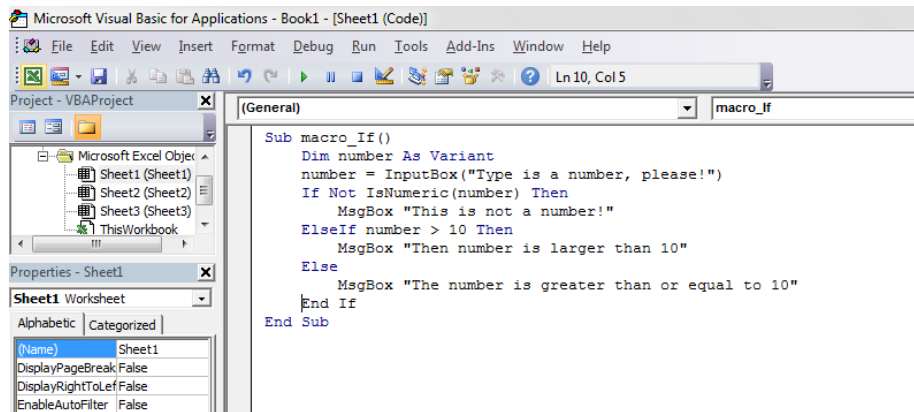
Gambar 2.2 Kode Statement kondisi

Jalankan Program tersebut dengan F8 atau melalui toolbar! Apa hasilnya?

Latihan3 :

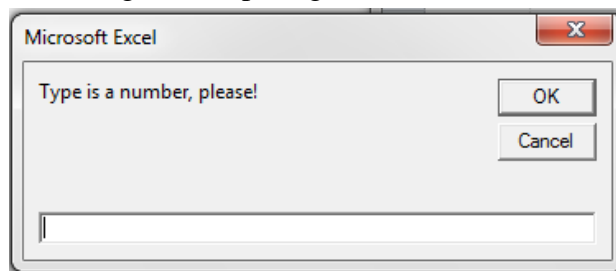
1. Bukalah Excel, kemudian tampilkan jendela VBE.
2. Kemudian beri nama dengan macro_if.

3. Setelah itu ketikkan kode program berikut ini.



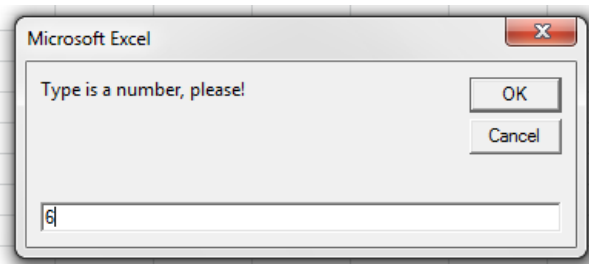
Gambar 2.3 Kode Stattement kondisi

4. Jalankan Program tersebut dengan mengklik toolbar atau F8
5. Akan muncul message box seperti gambar dibawah ini.



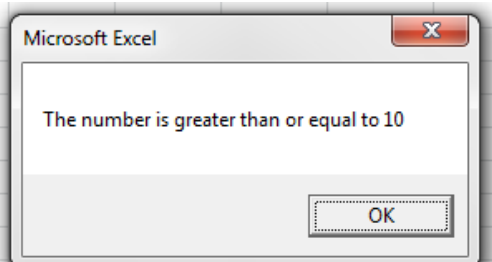
Gambar 2.4 Input Box pertama

6. Ketikkan angka pada input box misalnya "6", klik OK.



Gambar 2.5 Input Box Kedua

7. Kemudian akan muncul kembali message box yang menyatakan "The number is greater than or equal to 10"



Gambar 2.6 Message Box hasil input

8. Klik OK, maka akan kembali ke jendela VBE

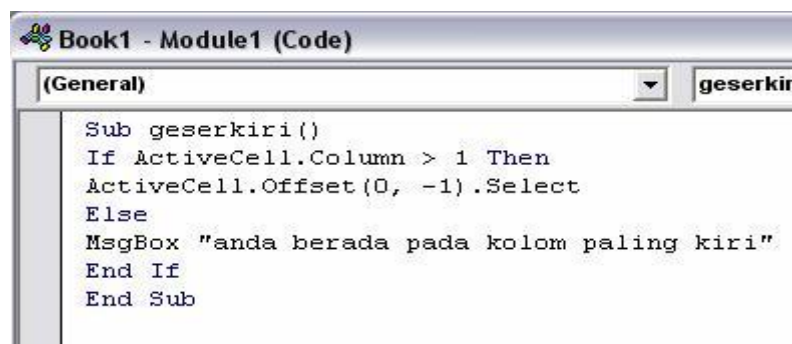
Operator Kondisional yang dipakai untuk statement kondisi:

Tabel 2.1 Operator kondisional

Operator	Keterangan
=	Dua angka bernilai sama. Kondisi ini juga dapat dipakai untuk fungsi string.
<	Nilai yang pertama lebih kecil dari nilai yang kedua.
>	Nilai yang pertama lebih besar dari nilai kedua
<=	Nilai pertama lebih kecil sama dengan nilai kedua.
>=	Nilai pertama lebih besar sama dengan nilai kedua
<>	Nilai pertama tidak sama dengan nilai kedua

Latihan4:

1. Buatlah sebuah modul baru di VBA
2. Ketikkan kode makro berikut ini



Gambar 2.7 Kode If kondisi

3. Aktifkan lembar kerja Excel kembali, kemudian letakkan pointer sel pada B1. Jalankan makro, maka sel aktif berpindah ke A1. Akan tetapi setelah sel aktif berada di A1, maka apabila program dijalankan menampilkan pesan, seperti gambar



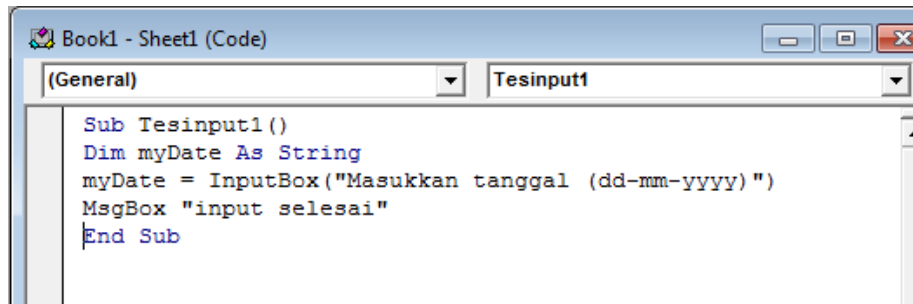
Gambar 2.8 Message box hasil kondisi

4. Klik OK untuk menutup kotak pesan itu.

Branch atau percabangan dapat digunakan untuk melakukan error checking terhadap input yang dimasukkan. Hal ini sangat penting karena dengan melakukan checking tersebut, program akan berjalan sesuai dengan yang diinginkan.

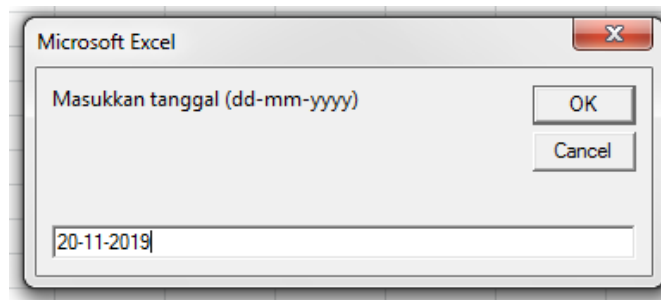
Latihan5:

1. Buatlah makro seperti dibawah ini.



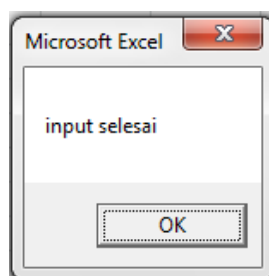
Gambar 2.8 MessageBox untuk if

2. Berpindahlah ke layar Excel, lalu jalankan makro tersebut. Jika makro yang ditulis benar, di layar akan ditampilkan kotak dialog seperti pada gambar berikut.



Gambar 2.9 Kotak Dialog tanggal

3. Isikan pada input box tersebut dengan data tanggal menggunakan format dd-mm-yy, kemudian klik tombol OK. Setelah itu akan tampil MsgBox dengan teks "Input Selesai".



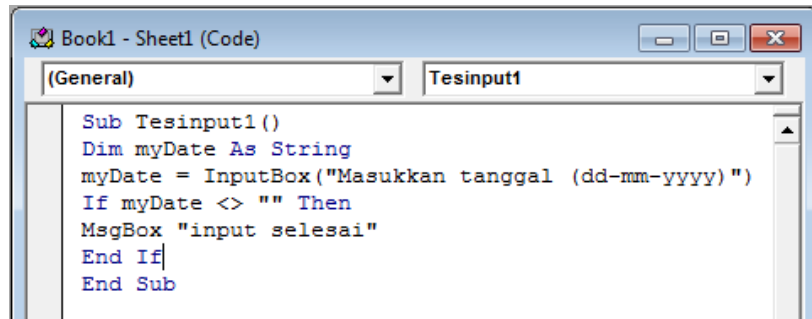
Gambar 2.10 Message Box tanggal

4. Klik OK untuk menutupnya

Pada input Box terdapat dua buah tombol, yaitu OK dan Cancel. Apabila tombol cancel diklik, MsgBox akan ditampilkan juga, padahal seharusnya dengan klik tombol cancel untuk membatalkan perintah. Oleh karena itu, pada makro perlu ditambahkan kondisi jika tombol

Cancel tidak diklik, maka akan menampilkan msgbox. Klik tombol Cancel sama saja dengan membiarkan input box = ''.

1. Tambahkan pada makro TestInput diatas sebagai berikut:

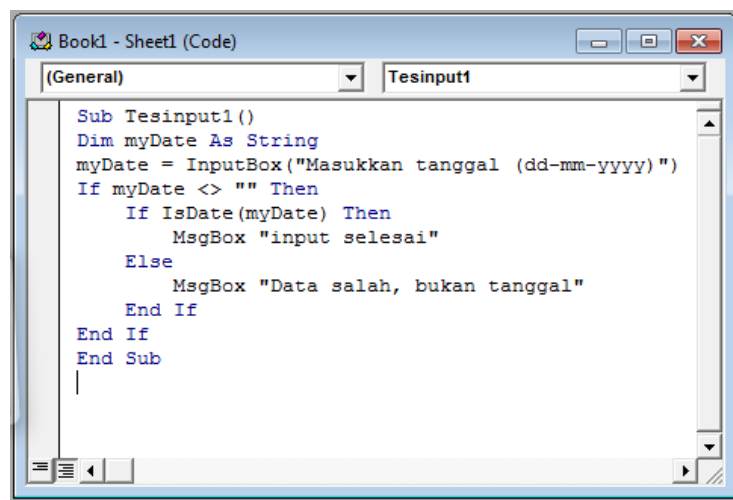


Gambar 2.11 Gambar Kode perubahan program input

2. Jalankan makro diatas, tekan tombol Cancel, maka makro tersebut tidak akan menampilkan msgbox lagi.

Sekarang jalankan kembali makro tersebut, kemudian isikan pada kotak input sembarang data teks, kemudian klik OK. Makro akan tetap menampilkan msgbox meskipun data yang dimasukkan tidak sesuai dengan kriteria yang disyaratkan.

Tambahkan kembali beberapa baris sehingga makronya akan sebagai berikut:



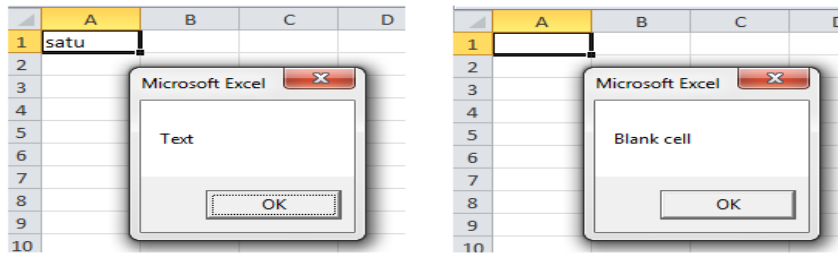
Gambar 2.12 Penambahan kode program

Jalankan makro tersebut dan masukkan teks pada input data lalu klik OK. Jalankan kembali sehingga akan ditampilkan kotak dialog seperti berikut.



Gambar 2.13 Kotak Dialog jika data yang diinput salah

Latihan 6: Buatlah makro yang menampilkan hasil seperti berikut :



Gambar 2.14 Kotak dialog output

Langkah- langkah nya:

1. Buatlah sebuah modul baru di VBA
2. Ketikkan kode program dibawah ini

```
Sub ContentChk()
    If Application.IsText(ActiveCell) = True
        Then MsgBox "Text"
    Else
        If ActiveCell = "" Then
            MsgBox "Blank cell"
        Else
            End If
        If ActiveCell.HasFormula Then
            MsgBox "formula"
        Else
            End If
        If IsDate(ActiveCell.Value) = True Then
            MsgBox "date"
        Else
            End If
        End If
    End Sub
```

3. Kembali ke worksheet pada excel dan letakkan pointer ke range yang memiliki nilai text atau kosong
4. Jalankan program VBA dan kemudian akan muncul message box yang menjelaskan bahwa cell yang aktif itu berupa text, kosong atau formula
5. Setelah selesai keluar dari excel dan simpan program .

Dari latihan makro yang sudah kita lakukan dapat diketahui bahwa struktur If tidak efisien jika dipakai untuk menyelesaikan masalah. Oleh karena itu jika memiliki banyak pilihan seperti diatas, jangan pergunkan pernyataan *If... Then- Else – End If* , tetapi gunakan struktur *Select Case ... End Select*. Syntax dari perintah **Select Case** adalah:

Select Case *ungkapan utama*

Case ungkapan1

Statement1

Case ungkapan2

Statement2

Case ungkapanN

StatementN

[Case Else

statementX]

End Select

Latihan1:

1. Buatlah modul baru pada VBA
2. Ketikkan program dibawah ini

```
Sub Select_case( )  
Select Case Range("A1").  
    Value Case is > 90  
        MsgBox "Nilai ujian A"  
    Case is > 80  
        MsgBox "Nilai ujian B"  
    Case is > 70  
        MsgBox "Nilai ujian C"
```



```

Case is > 65
    MsgBox "Nilai ujian D"
Case Else
    MsgBox " Tidak Lulus"
End Select
End Sub

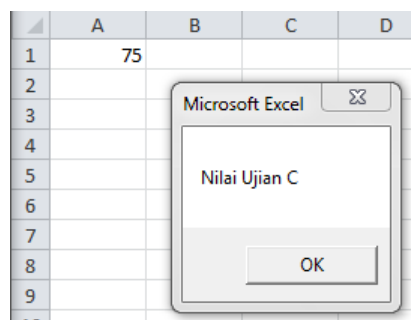
```

3. Buka WorkSheet Excel dan ketikkan nilai 75 di sel A1

	A	B
1	75	
2		
3		
4		
5		
6		

Gambar 2.15 Nilai pada worksheet

4. Kembali lagi ke Jendela VBA dan jalankan program
5. Jika program berhasil maka akan muncul message box seperti gambar dibawah ini.

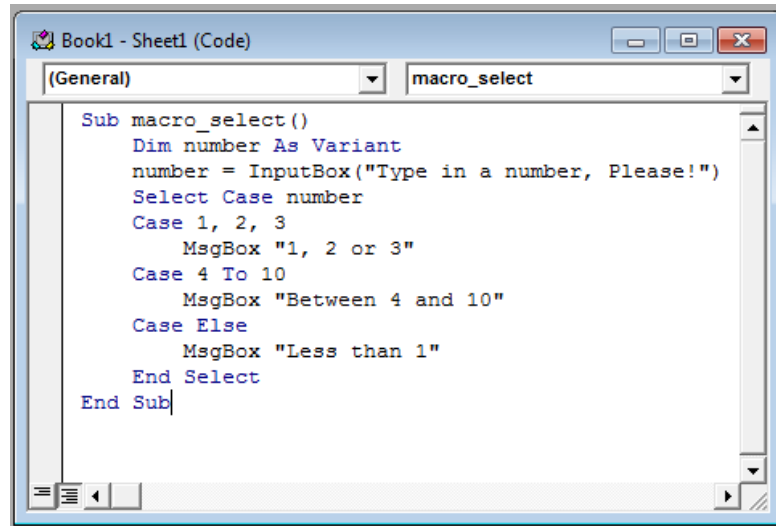


Gambar 2.16 Hasil dari program Nilai

6. Lakukan hal yang sama untuk nilai yang lain.
7. Setelah selesai tutup program, sehingga kita dapat memulai program yang lain.

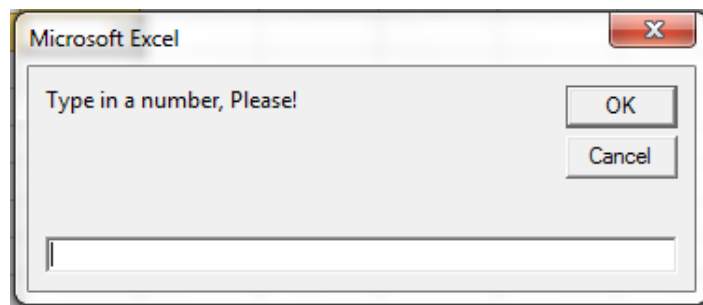
Latihan 2:

1. Buatlah Modul baru pada VBA.
2. Ketikkan Program select case dibawah ini.



Gambar 2.17 Program Select Case 2

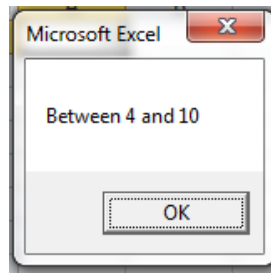
3. Jalankan program VBA dengan F5 atau F8
4. Setelah program dijalankan maka akan tampil Input Box pada Work Sheet seperti gambar dibawah ini.



Gambar 2.18 Input Box program

Excel 5. Masukkan nilai 5 pada input Box

6. Klik OK maka kemudian akan tampil Message Box yang menjelaskan bahwa 5 antara 4 dan 10.



Gambar 2.19 Hasil dari inputan

7. Jika selesai maka keluar dari program VBA dan Excel.

2.2 Struktur Pengulangan

Struktur pengulangan dapat digunakan untuk melaksanakan kode secara berulang. Pengulangan yang berulang kali dalam melaksanakan satu baris atau blok kode diterapkan terlebih dahulu banyaknya pengulangan. Struktur pengulangan terdiri dari :

1. Perintah *For... Next*

Perulangan yang sederhana adalah perintah *For* dan *Next*. Sebuah pengulangan menggunakan variabel konter yang meningkatkan atau menurunkan nilai selama perintah pengulangan dijalankan

Syntax dari perintah **For... Next** yaitu:

For konter = [nilai awal] to [nilai akhir]

Statement

Next [konter]

Latihan : Buatlah program untuk makro- makro dibawah ini, kemudian jalankan makro berikut.

```
Sub macro_loop1()  
    Dim i As Integer  
    For i = 1 To 10  
        If i > 5 Then Exit For  
        Debug.Print i  
    Next i  
End Sub  
  
Sub macro_loop2()  
    Dim i As Double  
    For i = -0.3 To 0.3 Step 0.1  
        Debug.Print i  
    Next i  
End Sub
```

```
Next i
End Sub

Sub For_Next ()
    For i = 1 To 10
        Cells(i, 1) = i
    Next i
End Sub
```

Perintah For...Next menggunakan Step

Syntax dari perintah **For...Next** menggunakan **Step** yaitu:

For counter = [nilai awal] to [nilai akhir] step [jumlah lompatan]
Statement
Next [counter]

```
Sub For_Next_Step1 ()
    For i = 1 To 10 Step 2
        Cells(i, 1) = 1
    Next i
End Sub
```

Apa hasil dari program berikut? Tampilkan hasilnya!

Catatan: Variabel counter adalah variabel yang menyatakan jumlah pengulangan yang bertipe numerik.

2. Perintah For...Each...Next

Syntax dalam For Each Next agak sedikit berbeda dari For Next, karena variabel diberikan setelah For each. Contoh dari for.. each.. Loop

```
Sub ShowName ()
    Dim oWSheet As Worksheet
    For Each oWSheet In Worksheets
        MsgBox oWSheet.Name
    Next oWSheet
End Sub
```

3. Perintah Do... Loop

Perintah Do.. Loop digunakan untuk menguji suatu kondisi awal dari pengulangan. Perintah Do.. Loop akan menjalankan pengulangan sepanjang kondisi **True** dan berhenti ketika kondisi menjadi **False**.

Syntax dari perintah Do While.. Loop:

Do[While kondisi]

Statement

Loop

Contoh:

```
Sub Do_While_Loop( )  
    I =1  
    Do While I<= 10  
        Cells(I,1)=I  
    Loop  
End Sub
```

4. Perintah Do...Until

Perintah Do Until digunakan sampai kondisi yang diinginkan terpenuhi. Syntax dari perintah Do Until.. Loop yaitu:

Do[Until kondisi]

Statement

Loop

Contoh :

```
Sub test_do()  
x = 0  
Do Until x = 100  
    x = x + 1  
Loop  
MsgBox x  
End Sub  
Sub Do_Until_Loop()  
    I=1  
    Do until I=11  
        Cells(I,1)=1  
    Loop  
End Sub
```

5. Perintah Do...Loop While

Syntax dari perintah Do..Loop While :

Do***Statement*****Loop[While kondisi]**

Contoh:

```
Sub Do_Loop_While( )  
    I=1  
    Do  
        Cells( I,1)=I  
        I=I+1  
    Loop While  
End Sub
```

6. Perintah Do...Loop Until

Syntax dari perintah Do..Loop Until:

Do***Statement*****Loop[Until**

kondisi] Contoh :

```
Sub Do_Loop_Until()  
    I=1  
    Do  
        Cells(I,1)=1  
    Loop Until  
I=1 End Sub
```

7. Perintah While... Wend

Perintah While..Wend akan terus melakukan pengulangan sampai kondisi bernilai benar.

Dan akan berhenti sampai kondisi bernilai salah Contoh:

```
Sub test_do()  
    x = 0  
    While x < 50  
        x = x + 1  
    Wend  
    MsgBox x  
End Sub
```

Catatan: Kadang – kadang terjadi kesalahan yang mengakibatkan pengulangan tanpa batas.

Dan itu terjadi ketika kondisi tidak pernah bernilai False. Dalam Kasus demikian, kita dapat stop pengulangan dengan menekan **ESC** atau **CTRL + BREAK**.

2.3 Function

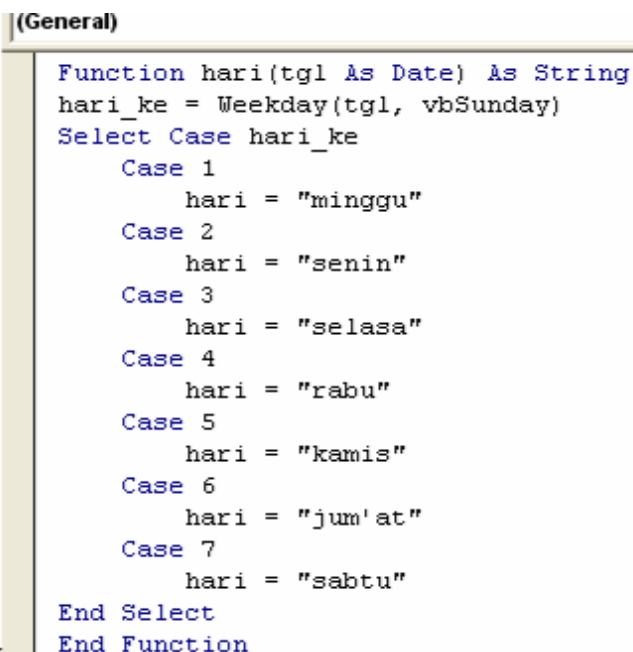
Perintah Function lebih sulit dikerjakan didalam spreadsheet daripada menggunakan kode. Formula dikunci sehingga user tidak dapat merubah atau melihat. Contoh pemakai function dengan rincian sebagai berikut:

First \$2500 is tax free.
Next \$2500 is taxable at 5%.
Anything above \$5000 is taxable at 10%.
In cell A1 type Income and in cell B1 type in your income in numbers say \$20000.
In cell A2 type Tax payable and in cell B2 type =tax(B1).
Put the following code in a module. The tax payable here would be \$1625.

```
Public Function tax(income As Single)
Select Case income
Case Is <= 2500
tax = 0
Case Is <= 5000
tax = (5000 - 2500) * 0.05
'in this case is 125
Case Else
tax = (income - 5000) * 0.1 + 125
End Select
End Function
```

Penggunaan fungsi untuk menampilkan nama hari sesuai dengan tanggal kita ketikkan pada lembar kerja.

1. Bukalah Modul VBA baru
2. Ketikkan program dibawah ini



The screenshot shows the VBA code editor with the 'General' tab selected. The code defines a function named 'hari' that takes a date as input and returns the day of the week as a string. The function uses a 'Select Case' statement to map the numeric values returned by the 'Weekday' function to the corresponding day names: 1 for 'minggu', 2 for 'senin', 3 for 'selasa', 4 for 'rabu', 5 for 'kamis', 6 for 'jum'at', and 7 for 'sabtu'.

```
(General)
Function hari(tgl As Date) As String
hari_ke = Weekday(tgl, vbSunday)
Select Case hari_ke
Case 1
hari = "minggu"
Case 2
hari = "senin"
Case 3
hari = "selasa"
Case 4
hari = "rabu"
Case 5
hari = "kamis"
Case 6
hari = "jum'at"
Case 7
hari = "sabtu"
End Select
End Function
```

Gambar 2.20 Kode program menampilkan hari

3. Panggil fungsi menampilkan hari dengan mengetikkan rumus pada sel C2 : = hari[B2], kemudian tekan tombol **Enter**.
4. Pastikan sel B2 terpilih, kemudian buatlah nilai date atau tanggal pada sel tersebut, misalkan ketikkan tanggal 12/02/07 kemudian tekan **Enter**.

Apa hasil dari program diatas ? Tampilkan!

