

VISUAL BASIC 6.0

SETYO BUDI, M.KOM

Pendahuluan

- ❑ Program adalah suatu proses yang saling menyambung. Untuk mencapai tujuan program harus melalui tahap-tahap tertentu, sehingga dibutuhkan alur program yang jelas dan efisien.
- ❑ VB 6 mengenal beberapa tipe alur seperti percabangan dan perulangan.
- ❑ Kita dapat menggunakan **perintah percabangan (branching)** dan **perulangan (looping)** untuk beberapa hal antara lain :
 - Mengeksekusi perintah kondisional
 - Menyatakan perulangan

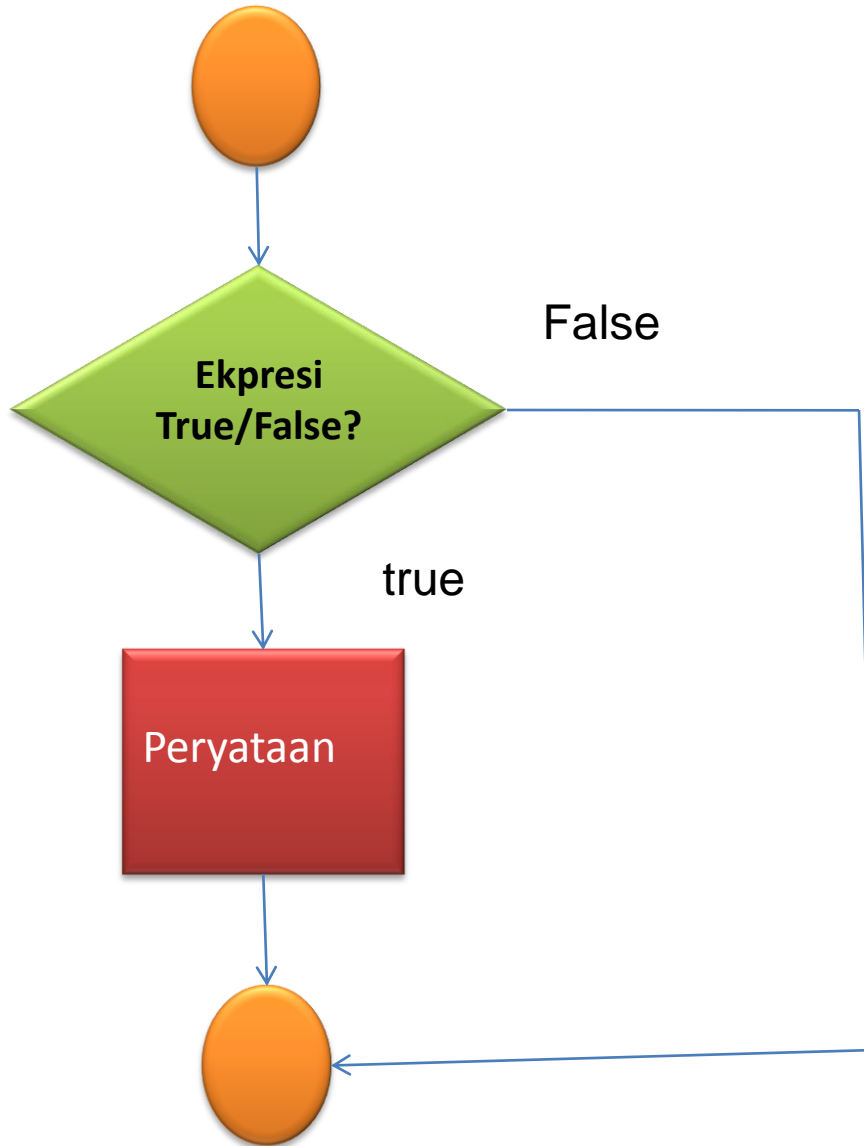
Pendahuluan

- ❑ Struktur kendali / percabangan memungkinkan kita untuk mengatur jalannya program.
- ❑ Untuk mewujudkan alur, VB 6 memiliki beberapa sarana, antara lain :
 - **If..then**
 - **Select..case**

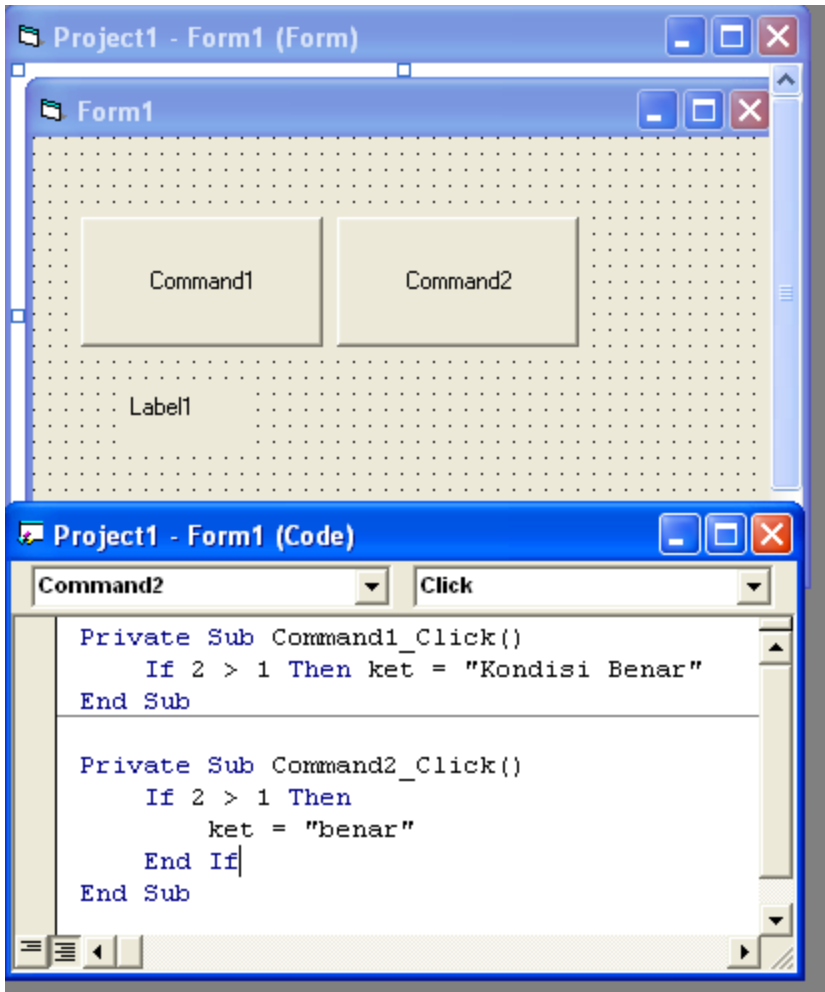
Statement If ...Then

- ❑ Statement **if** berfungsi untuk melakukan seleksi berdasarkan suatu kondisi tertentu
- ❑ **If..then** digunakan untuk menjalankan suatu pernyataan secara kondisional.
- ❑ Jika ekspresi tersebut benar (true) maka rangkaian statement akan dijalankan.
- ❑ Jika ekspresi tersebut salah (false) maka akan diloncati.

Perintah IF-THEN



Bentuk Umum Statement IF-THEN



If <kondisi> Then pernyataan

Atau

**If <kondisi> Then
Pernyataan**

End If

Perintah IF-THEN

Contoh :

If 2 > 1 Then Ket = “Kondisi Benar”

Atau

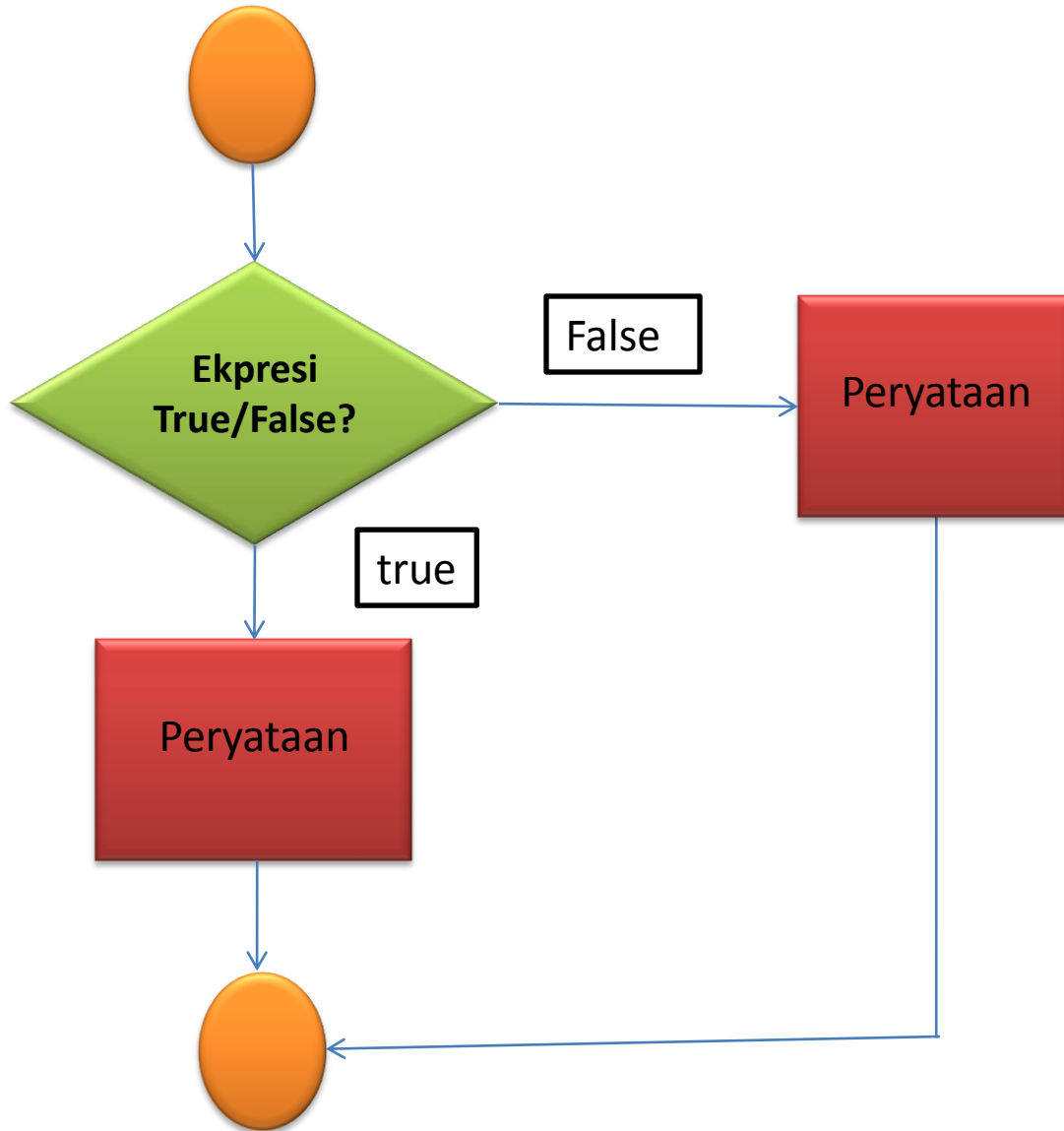
**If 2 > 1 Then
 ket = “Kondisi Benar”
endif**

Hasilnya : Kondisi Benar

Statement If..Then..Else

- Sama fungsinya dengan **If..then**, statement **If..Then..Else** digunakan untuk menjalankan suatu pernyataan secara kondisional
- Statement **If..Then..Else** akan mengerjakan perintah dengan mengevaluasi ekspresi boolean dengan memberikan alternatif lain jika ekspresi boolean tersebut bernilai false.
- Statement **If..Then..Else** cocok untuk memecahkan masalah dengan lebih dari satu kemungkinan kondisi / pilihan

If..Then..Else



Bentuk Umum Statement If..Then..Else

If <ekspresi> **Then**

[*blok pernyataan-1*]

Else

[*blok pernyataan-n*]

End If

Ket : Digunakan jika perintah setelah **Then** berisi banyak perintah (blok perintah), sedangkan **Else** akan dieksekusi jika kondisi **If** tidak terpenuhi

Contoh :

If 2 > 1 **Then**

Ket = "Kondisi Benar"

Else

Ket = "Kondisi Salah"

End If

Hasil : Kondisi Benar

Contoh IF-Then Else

Pengkondisian dengan operator logika “And”

If (2 = 2) AND (4 > 2) Then

LblStatus.Caption = “KONDISI BENAR”

Else

LblStatus.Caption = “KONDISI SALAH”

End If

Ket : Untuk And kedua kondisi harus

bernilai benar agar program akan mengeksekusi perintah setelah then. Kondisi diatas akan mengeksekusi (menghasilkan) LblStatus dengan nilai caption “**Kondisi Benar**”.

Contoh IF-Then Else

Pengkondisian dengan operator logika “Or”

```
If (2 = 2) OR (2 = 0) Then
```

```
    LblStatus.Caption = “KONDISI BENAR”
```

```
Else
```

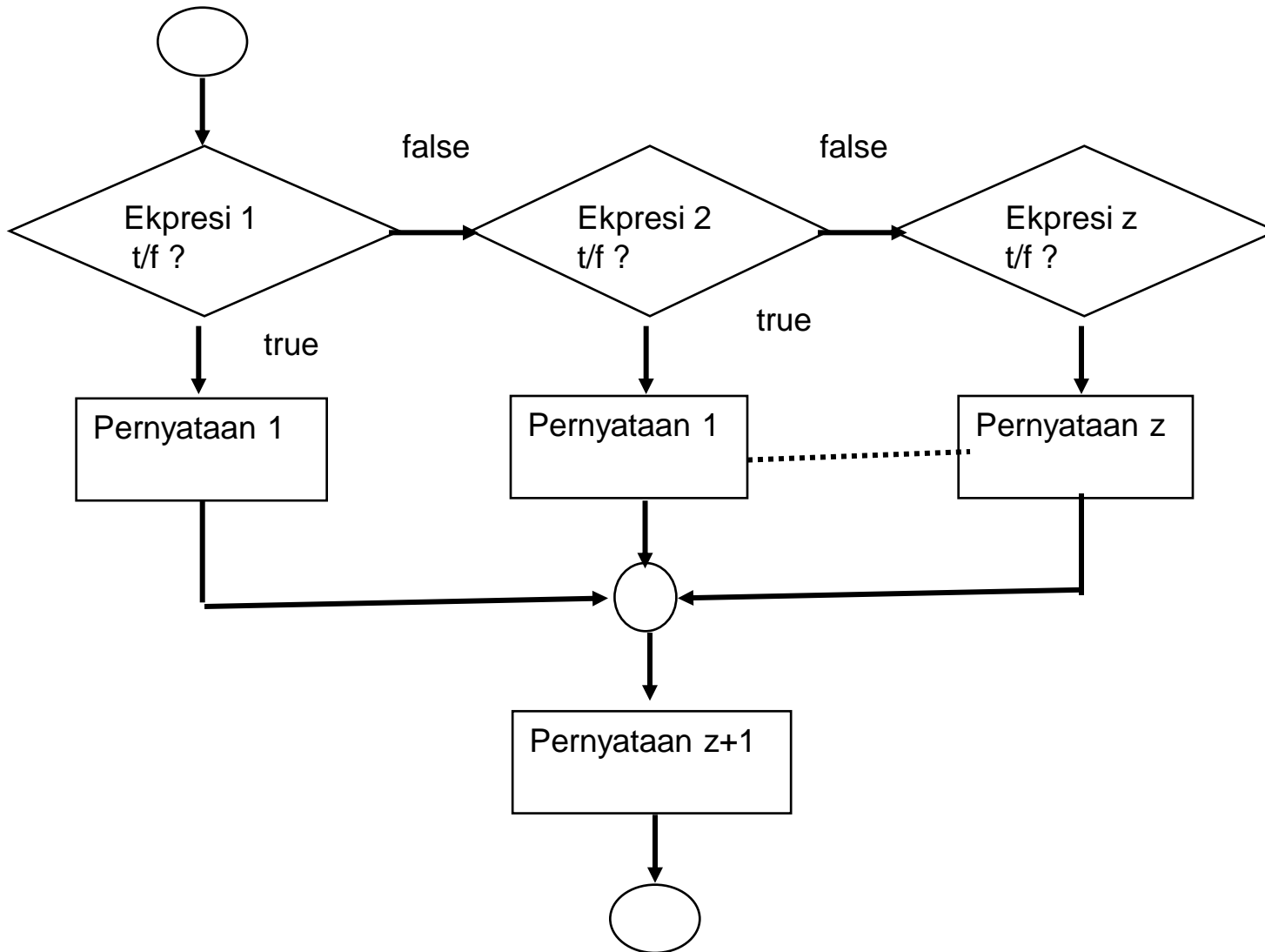
```
    LblStatus.Caption = “KONDISI SALAH”
```

```
End If
```

Ket : Untuk Or jika salah satu saja kondisi bernilai benar, maka program akan mengeksekusi perintah setelah then. Kondisi diatas akan mengeksekusi (menghasilkan) LblStatus dengan nilai caption “Kondisi Benar”.

Perintah IF-THEN BERSARANG

Membuat **if ... then** lain didalam **if . then** yang telah ada



Perintah IF-THEN BERSARANG

If <ekspresi1> Then

[blok pernyataan-1]

Elseif <ekspresi2> Then

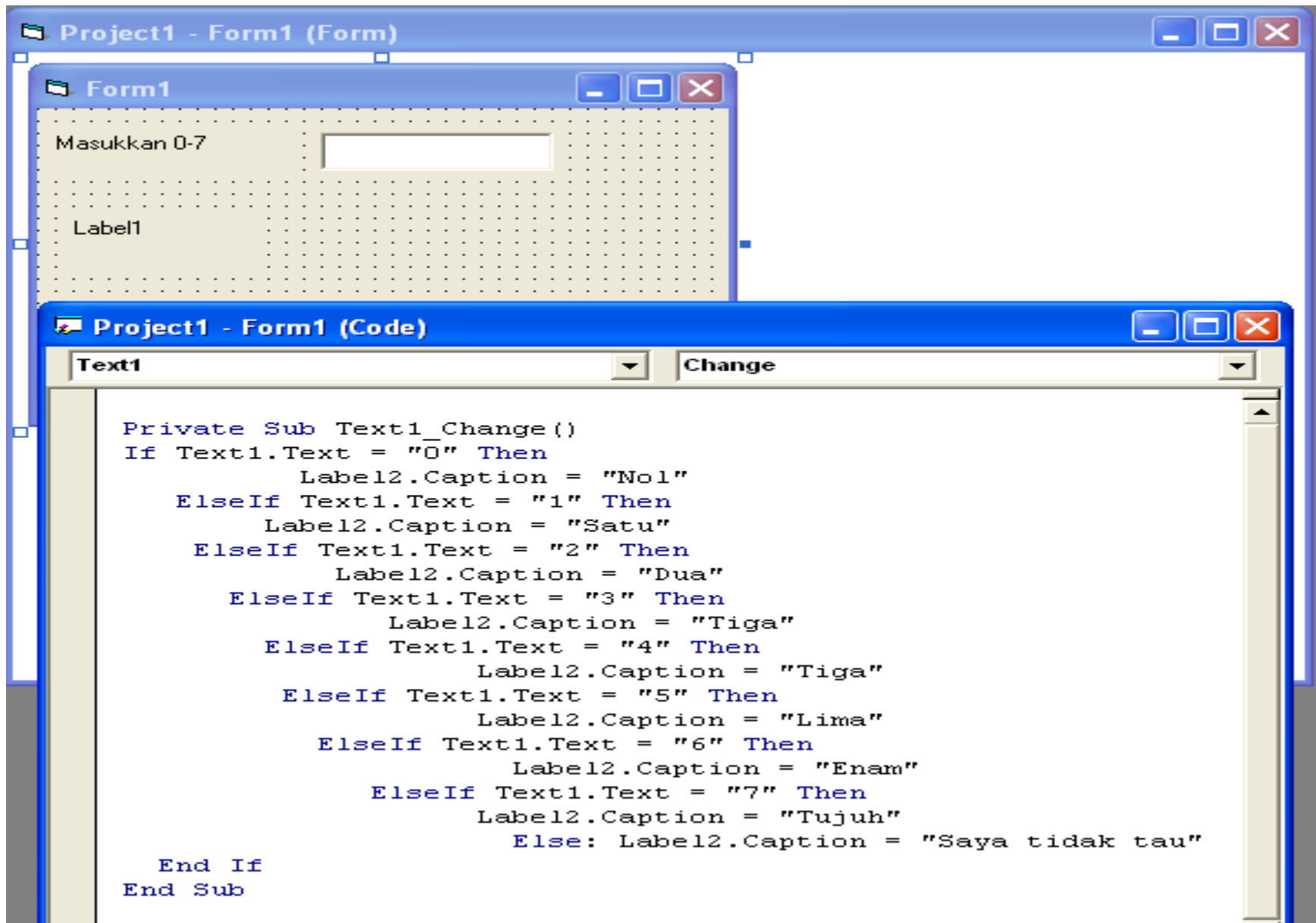
[blok pernyataan-2]] ...

Else

[blok pernyataan-n]]

End If

Perintah IF-THEN BERSARANG



Perintah IF-THEN BERSARANG

Private Sub Text1_Change()

```
If Text1.Text = "0" Then
    Label2.Caption = "Nol"
Elseif Text1.Text = "1" Then
    Label2.Caption = "Satu"
Elseif Text1.Text = "2" Then
    Label2.Caption = "Dua"
Elseif Text1.Text = "3" Then
    Label2.Caption = "Tiga"
Elseif Text1.Text = "4" Then
    Label2.Caption = "Empat"
Elseif Text1.Text = "5" Then
    Label2.Caption = "Lima"
Elseif Text1.Text = "6" Then
    Label2.Caption = "Enam"
Elseif Text1.Text = "7" Then
    Label2.Caption = "Tujuh"
Else: Label2.Caption = "Saya tidak tau"

End If
End Sub
```


Percabangan SELECT-CASE

Program akan menilai setiap kondisi, dan akan mengeksekusi pernyataan yang berada pada lokasi terbentuknya nilai true SELECT.- CASE lebih spesifik digunakan, untuk melakukan penyeleksian lebih dari satu kondisi.

```
SELECT CASE <ekspresi>
```

```
  [CASE nilai 1]
```

```
    [perintah 1]
```

```
    [perintah 2]
```

```
  [CASE nilai 2]
```

```
    [perintah 3]
```

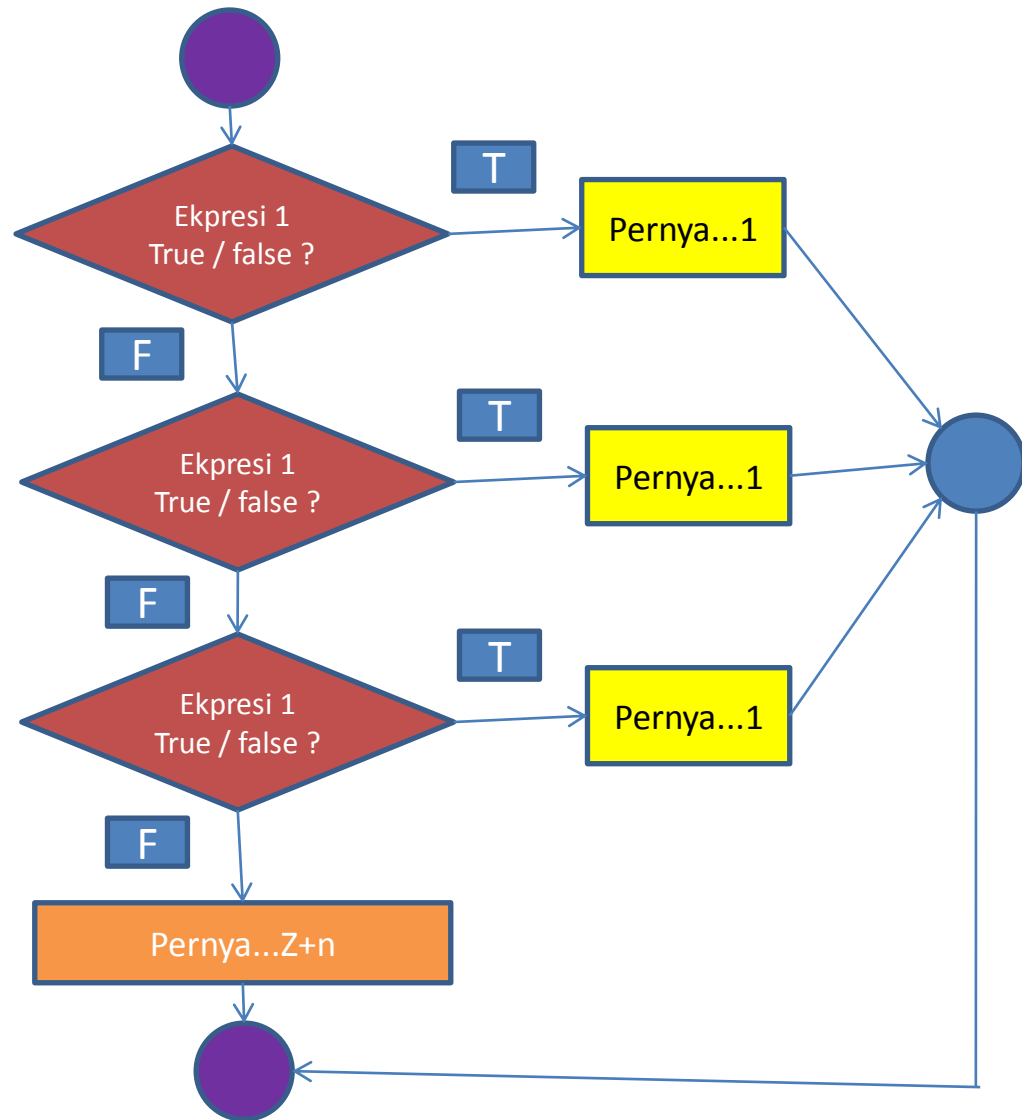
```
    [perintah 4]
```

```
  CASE ELSE
```

```
    [ Perintah n ]
```

```
END SELECT
```

Percabangan SELECT-CASE



Percabangan SELECT-CASE

```
Private Sub Form_Load()
```

```
    Dim pilihan As String
```

```
End Sub
```

```
Private Sub Text1_Change()
```

```
    pilihan = Text1.Text
```

```
    Select Case pilihan
```

```
    Case 0
```

```
        Label2.Caption = "nol"
```

```
    Case 1
```

```
        Label2.Caption = "satu"
```

```
    Case 2
```

```
        Label2.Caption = "dua"
```

```
    Case Else
```

```
        Label2.Caption = "Saya tidak tau"
```

```
    End Select
```

```
End Sub
```

PERULANGAN

Instruksi perulangan adalah instruksi yang dapat mengulang pelaksanaan sederetan instruksi-instruksi lainnya berulang-ulang sesuai persyaratan yang telah di tetapkan.

- Dengan statement perulangan kita cukup membuat pernyataan sebanyak satu kali maka dapat dikerjakan beberapa kali
- Dalam visual basic perulangan dapat dibagi menjadi dua:
 1. Perulangan terhitung, dan
 2. Perulangan tak terhitung

Perulangan Terhitungr

- Disebut perulangan terhitung karena **nilai awal dan nilai akhirnya diketahui secara pasti.**
- Pada perulangan jenis ini telah **diketahui berapa kali perulangan akan dilakukan**
- Yang termasuk perulangan terhitung adalah **FOR-NEXT**

Statement FOR-NEXT

- For-Next menggunakan suatu variabel yang disebut *counter* yang akan bertambah atau berkurang pada setiap perulangan.
- Mengulang sebanyak yang kita tentukan.

Perulangan Terhitung

- Bentuk Umum

For <var> = <start> To <end> Step [n]

statements

Next

- **Var** adalah nama variabel numerik
- **Start** adalah nilai awal variabel, atau batas bawah dari variabel
- **End** adalah nilai akhir atau batas atas nilai variabel
- **n** adalah banyak lompatan variabel jika tidak ditulis maka n akan bernilai 1

Contoh Statement For-Next

Private Sub

Command1_Click()

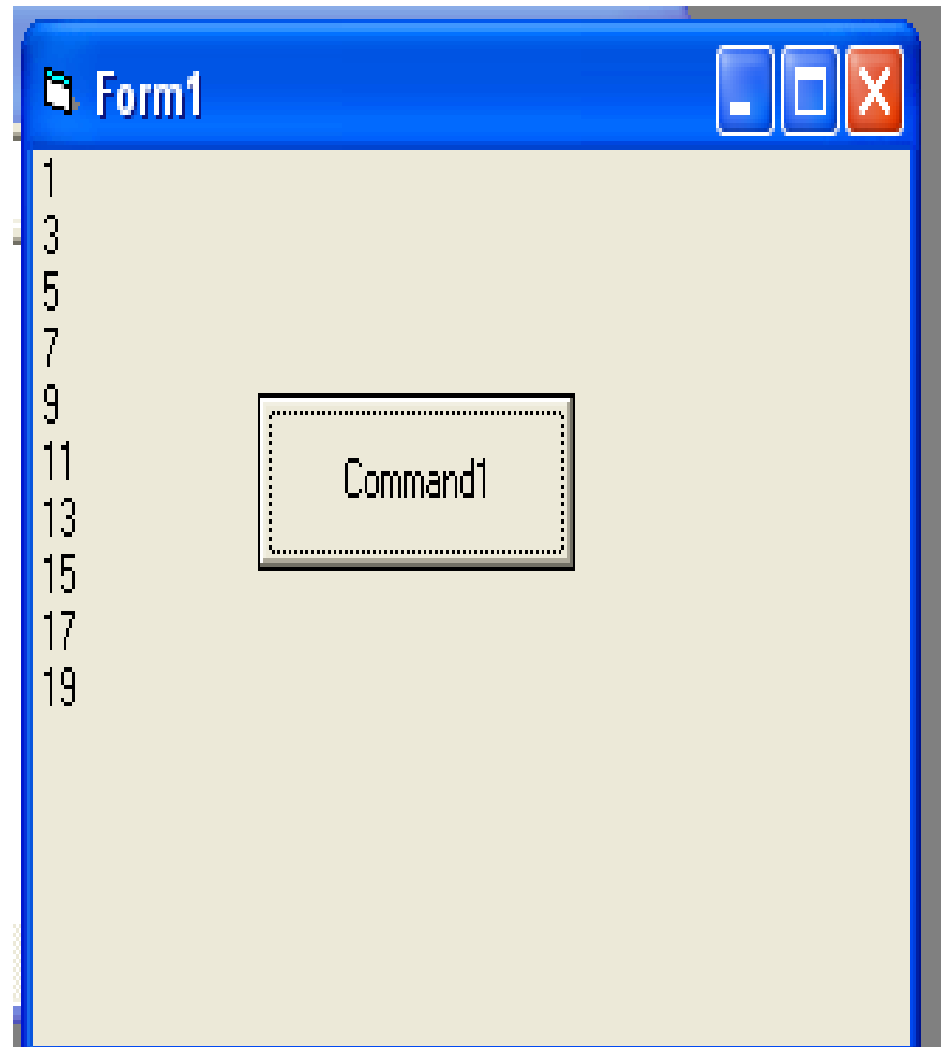
Dim i As Integer

For i = 1 To 20 Step 2

Print i

Next

End Sub



Perulangan Tak Terhitung

- Pada perulangan tak terhitung, tidak dapat dipastikan jumlah pengulangan yang dilakukan.
- Perulangan dilakukan tergantung pada kondisi tertentu.
- Untuk perulangan tak terhitung digunakan perintah
 1. WHILE-WEND
 2. DO-LOOP

While-Wend

- Fungsi dari perintah ini adalah mengulang perintah yang terletak antara WHILE dan WEND, selama kondisi pada perintah WHILE masih bernilai benar .
- Sintaks perulangan While ...Wend adalah :

WHILE <ekspresi>

perintah1

perintah2

perintah3

WEND

Setiap perintah WHILE harus berpasangan dengan WEND

Contoh WHILE-WEND

Private Sub Command1_Click()

```
Dim i As Integer
```

```
i = 0
```

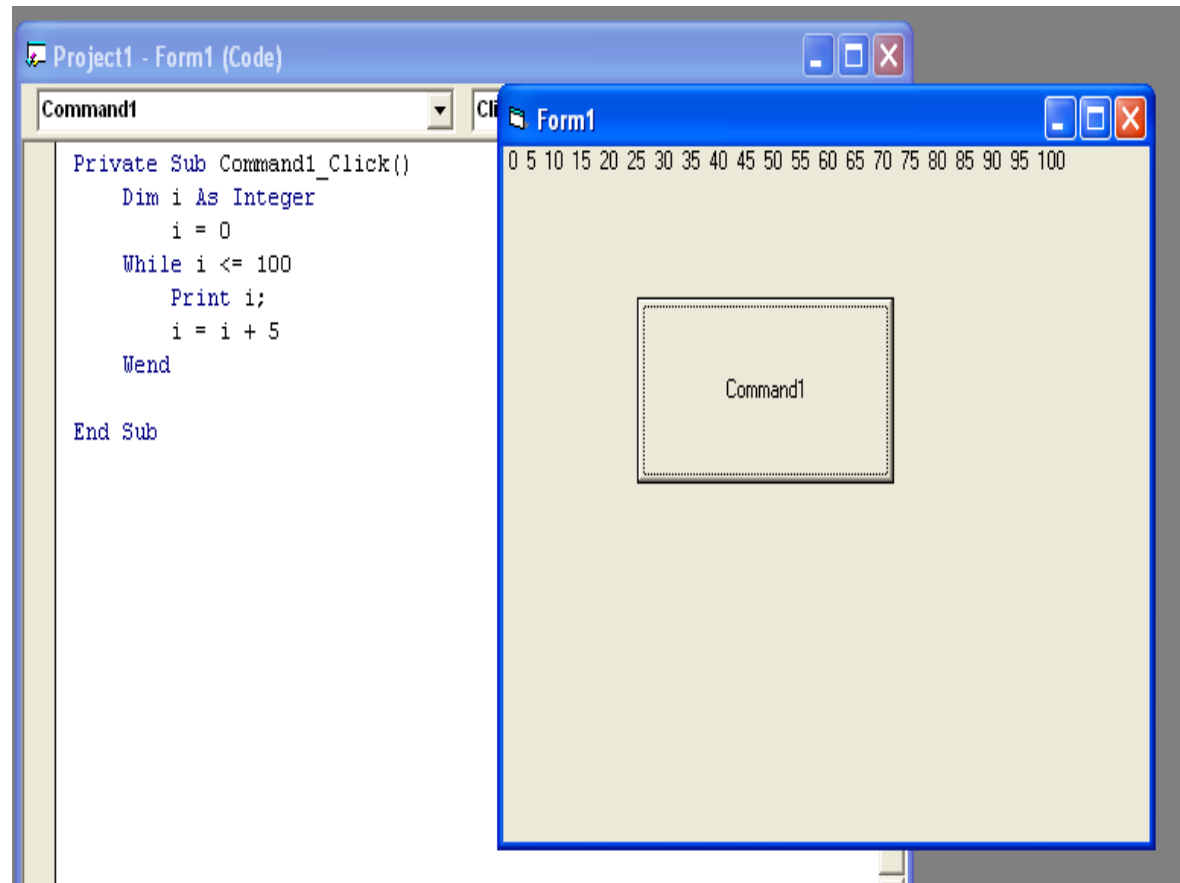
```
While i <= 100
```

```
Print i;
```

```
i = i + 5
```

```
Wend
```

End Sub



Do – Loop

Perintah DO - LOOP hampir sama dengan perintah WHILE - WEND. Bedanya, jika pada WHILE - WEND perulangan dilakukan selama kondisi pada WHILE benar, sedangkan pada DO - LOOP perulangan dilakukan selama kondisi pada LOOP salah.

Bentuk umumnya:

DO

- perintah1
- perintah2

.....

LOOP UNTIL <kondisi>

Selama kondisi masih salah, maka perintah1, perintah2 akan dijalankan, hingga kondisi bernilai benar.

Contoh program DO-LOOP :

```
Private Sub Form_Load()
```

```
    I=20
```

```
    Do
```

```
        Debug.Print I;
```

```
        I=i+5
```

```
    Loop until i>=35
```

```
End sub
```