

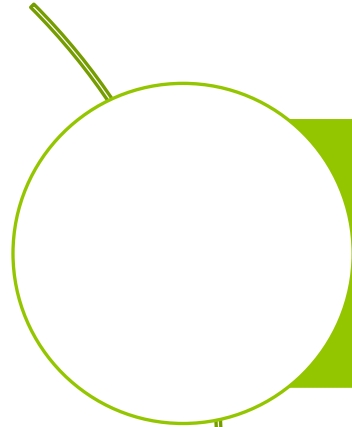
PEMROGRAMAN TERSTRUKTUR 2015

Pertemuan V



Oleh:

**AISYAH FITRI
YUNIASIH, S.ST,
SE, M.Si.**



Review Tutorial Access



Struktur Berulang

REFERENSI

- 1. Sprankle, M. dan Hubbard, J. (2012). Problem solving & programming concepts. 9th edition. Prentice Hall. Chapter 7.
- 3. Vine, M. (2005). Microsoft Access VBA programming for the absolute beginner. 2nd edition. Thomson Course Technology PTR. Chapter 4

DATA DAN DATABASE

- Data

Representasi fakta dunia nyata yang mewakili suatu objek

- Database → Basis Data

Kumpulan data yang saling berhubungan yang diorganisasikan sedemikian rupa sehingga tidak terjadi redundansi dalam penyimpanannya serta dapat dimanfaatkan kembali dengan cepat dan mudah untuk menghasilkan informasi yang akurat dalam memenuhi berbagai kebutuhan

DATABASE MANAGEMENT SYSTEM (DBMS)

- Piranti lunak yang digunakan untuk memanipulasi & menampilkan isi database dengan menjamin integritasnya → interface antara database dan user

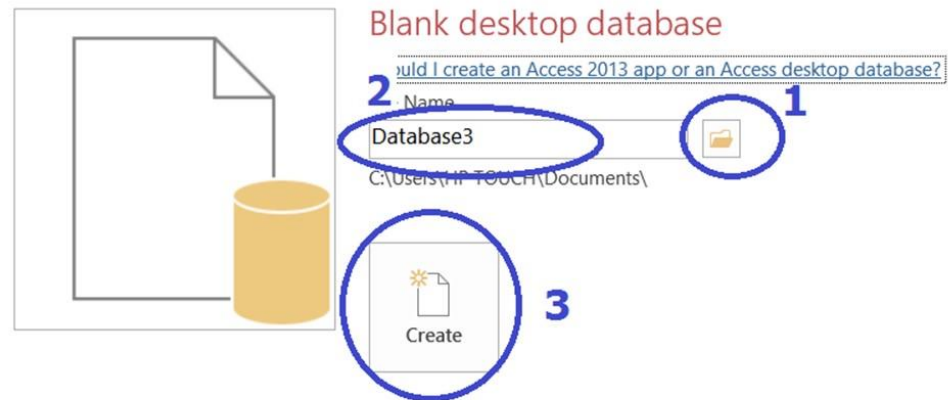
Contoh: dBase III+, dBase IV, FoxBase, Rbase, **MS Access**, Borland-Paradox atau Borland-Interbase, MS SQLServer, CA Open ingres, Oracle, Informix, Sybase, MySQL, PostgreSQL, FoxPro

DATABASE ACCESS

- Database → sekumpulan 'data' yang terorganisir.
- Database Access → sekumpulan obyek yang terorganisir
- Beberapa obyek database Access yang penting:
 - **Table**: tempat penyimpanan data
 - **Form**: untuk entri dan menampilkan data
 - **Query**: untuk penampilan dan modifikasi data dari satu atau lebih tabel
 - **Modul**: program atau aplikasi

MEMBUAT DATABASE BARU

- Buatlah database dengan nama "Petruk-05-NIM"
NIM = nomor induk mahasiswa
- Langkahnya:
 - Aktifkan Access
 - Klik "New Database" di area *template*
 - Isikan nama database dan tekan ENTER



- 1 - pilih folder**
- 2 - isi nama database**
- 3 - klik ikon 'Create'**

TABLE

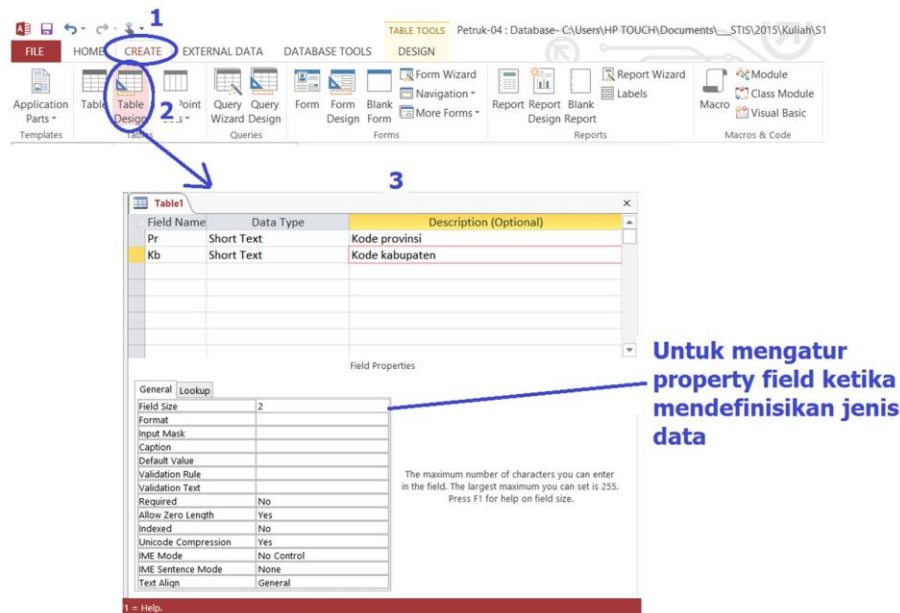
- Table: tempat penyimpanan data dalam sistem database Access
- **Baris**: merepresentasikan **record** atau **entitas**
- **Kolom**: merepresentasikan **field** atau **atribut**
- **Nilai field**: nilai atribut tertentu dari suatu entitas

FIELD

- Sebelum suatu tabel dapat digunakan, field-field yang diperlukan perlu dirancang terlebih dahulu:
- **Field Name:** nama field
- **Data Type:** jenis data field
- **Description:** penjelasan tentang field

MEMBUAT TABEL BARU: MERANCANG STRUKTUR DATA

- CREATE → Table Design
- Definisikan setiap field (nama, jenis data, deskripsi) yang diperlukan

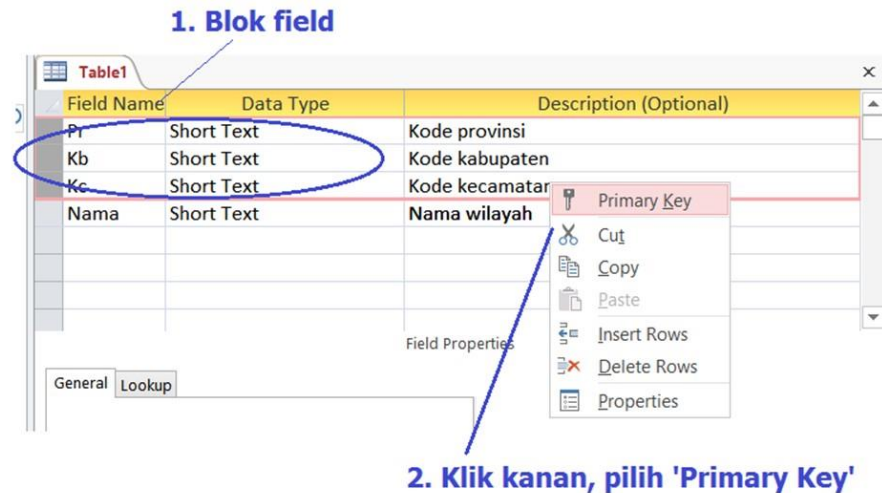


PRIMARY KEY

- Setiap record harus memiliki 'identitas' unik
- Primary key:** satu atau lebih field yang digunakan sebagai identitas atau pengenal record

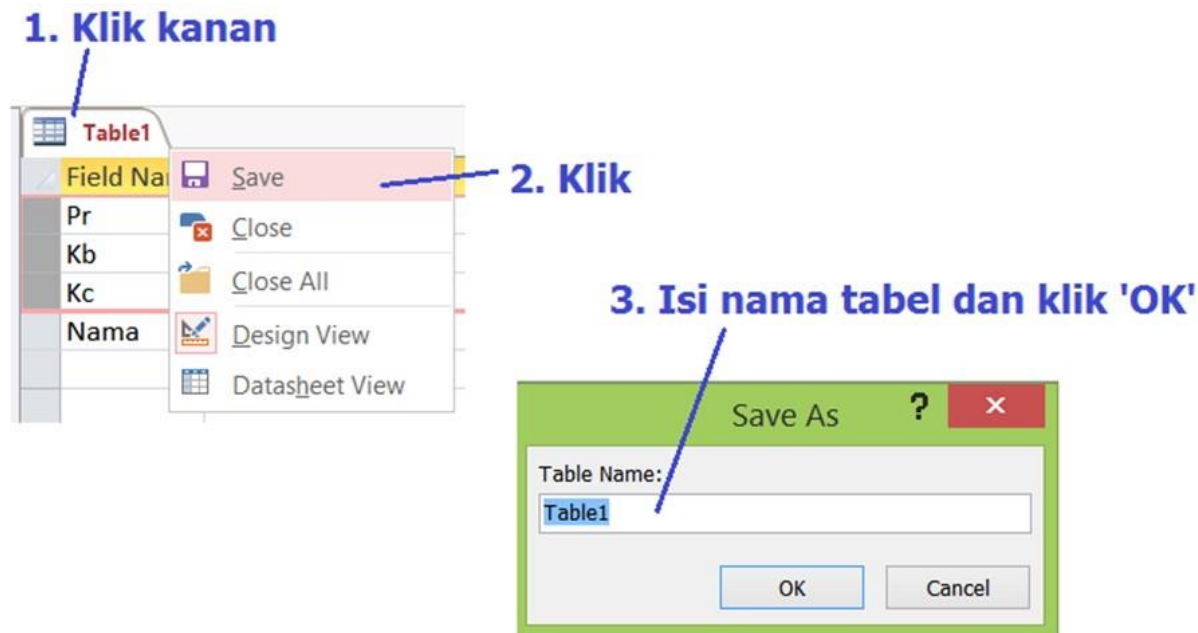
Cara pembuatan

- Klik kolom paling kiri dari field yang akan menjadi primary key (atau blok seluruh field yang diperlukan), klik kanan, pilih 'Primary Key'



MENYIMPAN TABEL

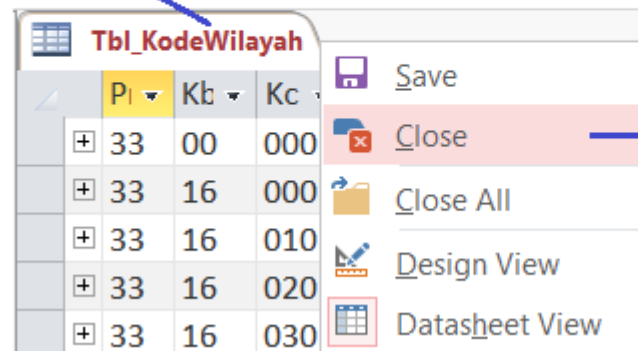
- Klik kanan nama tabel di panel
- Pilih 'Save'. Untuk penyimpanan pertama akan muncul jendela untuk memberikan nama tabel



MENUTUP TABEL

- Untuk menghindari kerusakan (karena salah entri dan sebab lain), setelah selesai digunakan sebaiknya tabel ditutup
- Jika ada modifikasi dari kondisi terakhir file ketika dibuka, akan muncul jendela konfirmasi

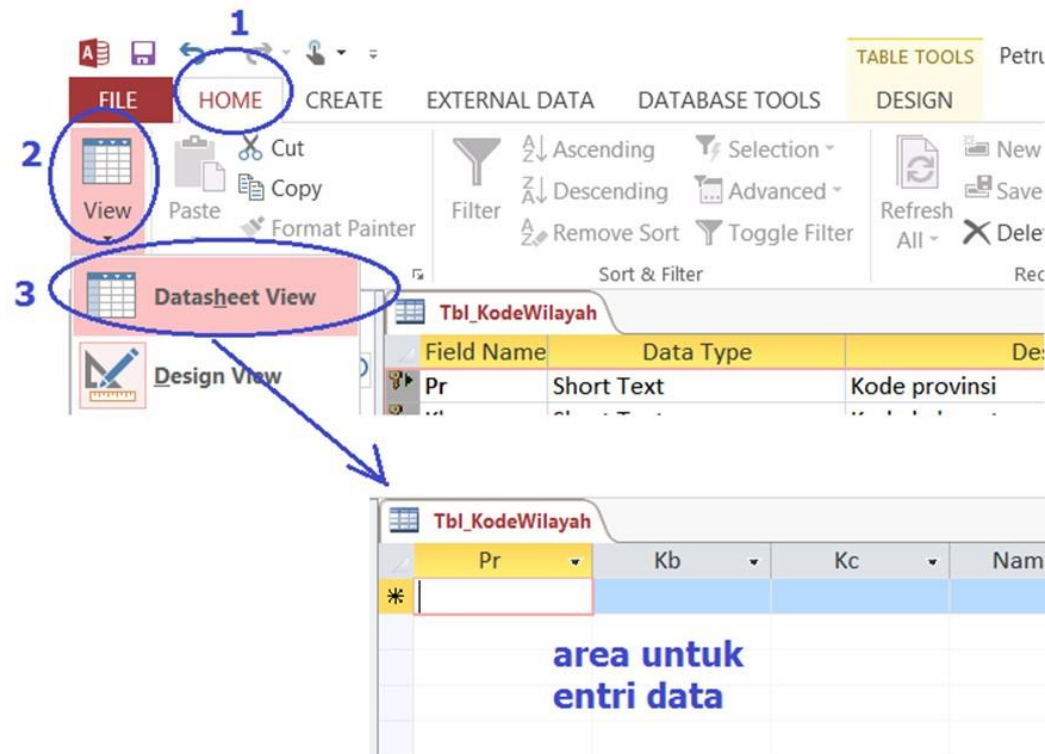
1. Klik kanan



2. Klik

ENTRI DATA

- HOME → View → Datasheet View
- Layar yang ditampilkan dapat digunakan untuk melakukan entri data



LATIHAN

- Buat dan entrilah data dua tabel berikut:

Pr	Kb	Kc	Pddk_K	Pddk_D
33	16	010	0	45,177
33	16	020	12,780	61,189
33	16	030	9,915	28,806
33	16	040	4,642	49,578
33	16	050	56,268	15,878

Pi	Kb	Kc	Nama
33	00	000	Jawa Tengah
33	16	000	Blora
33	16	010	Jati
33	16	020	Randublatung
33	16	030	Kradenan
33	16	040	Kedungtuban
33	16	050	Cepu

Primary key: Pr, Kb, Kc

QUERY

- Digunakan untuk menampilkan data
- Jika melibatkan lebih dari satu tabel, perlu didefinisikan hubungan antar tabel
- Untuk memahami konsep query, gunakan dua tabel yang sudah dibuat dalam latihan sebelumnya:
 - **Tbl_DataWilayah**
 - **Tbl_KodeWilayah**

ANDAI INGIN DITAMPILKAN DATA:

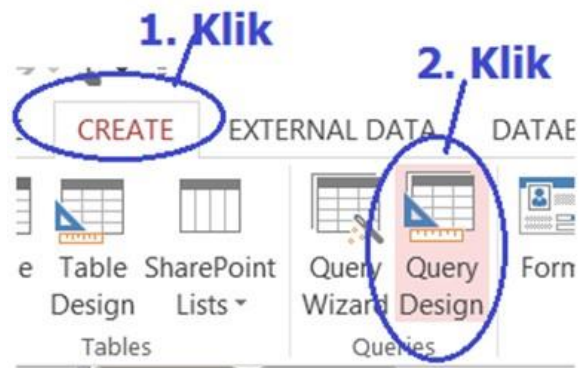
Tbl_DataWilayah

Hasil penjumlahan

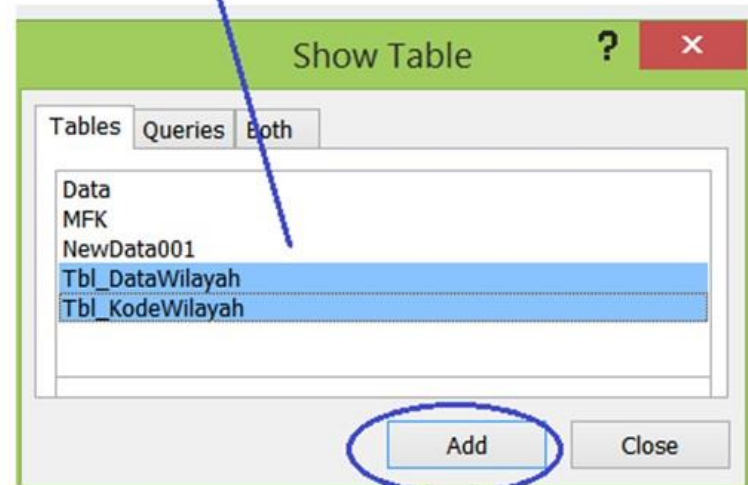
Pr	Kb	Kc	Nama	Pddk_K	Pddk_D	Jumlah
33	16	010	Jati	0	45,177	45177
33	16	020	Randublatung	12,780	61,189	73969
33	16	030	Kradenan	9,915	28,806	38721
33	16	040	Kedungtuban	4,642	49,578	54220
33	16	050	Cepu	56,268	15,878	72146

Tbl_KodeWilayah

PEMBUATAN QUERY



3. Pilih tabel yang akan ditampilkan, kemudian klik 'Add'



Add

LAYAR RANCANGAN QUERY



MERANCANG RELASI

- Relasi:

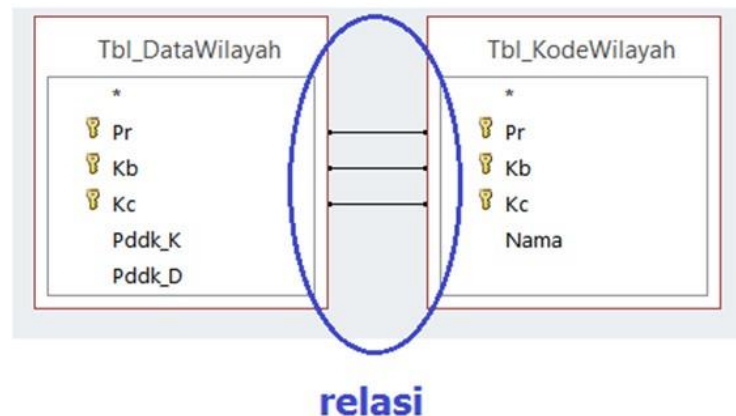
Tbl_DataWilayah.Pr \leftrightarrow Tbl_KodeWilayah.Pr

Tbl_DataWilayah.Kb \leftrightarrow Tbl_KodeWilayah.Kb

Tbl_DataWilayah.Kc \leftrightarrow Tbl_KodeWilayah.Kc

- Caranya:

- Drag 'Pr' dari tabel kiri ke kanan
- Drag 'Kb' dari tabel kiri ke kanan
- Drag 'Kc' dari tabel kiri ke kanan



MERANCANG TAMPILAN DATA

- Area untuk merancang tampilan data ada di bagian bawah area rancangan relasi
- Klik panah bawah di baris 'Field' akan ditampilkan semua field yang dapat ditampilkan.
- Penulisan field: "**NamaTabel>NamaField**"
- Untuk kolom yang berupa ekspresi: tuliskan formulanya

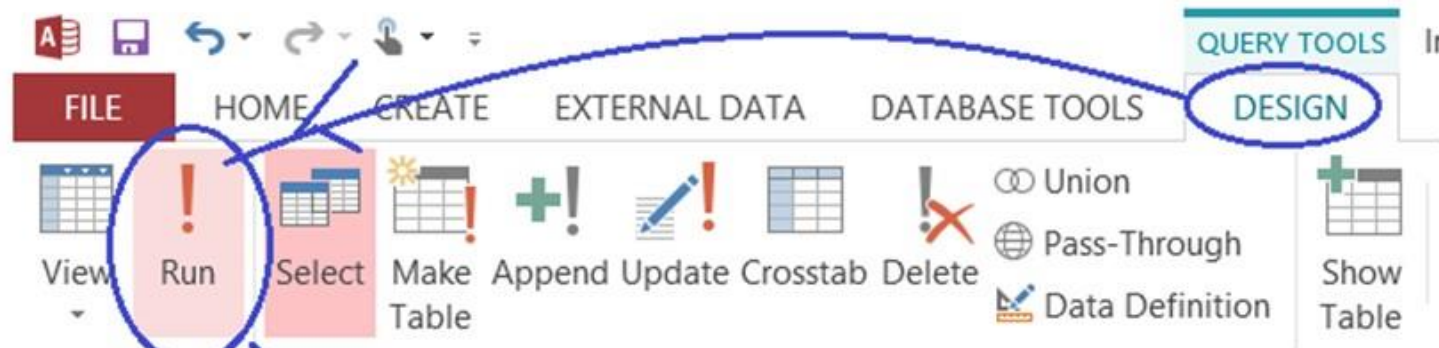
RANCANGAN TAMPILAN

dari tabel yang berbeda

Field:	Pr	Kb	Kc	Nama	Pddk_K	Pddk_D	Jumlah: [Pddk_K]+[Pddk
Table:	Tbl_DataWilayah	Tbl_DataWilayah	Tbl_DataWilayah	Tbl_KodeWilayah	Tbl_DataWilayah	Tbl_DataWilayah	
Sort:							
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:							
or:							

TAMPILAN HASIL QUERY

- Klik Run



Pr	Kb	Kc	Nama	Pddk_K	Pddk_D	Jumlah
33	16	010	Jati	0	45,177	45177
33	16	020	Randublatung	12,780	61,189	73969
33	16	030	Kradenan	9,915	28,806	38721
33	16	040	Kedungtuban	4,642	49,578	54220
33	16	050	Cepu	56,268	15,878	72146
*						

ENTRI DATA DENGAN LAYAR QUERY

- Layar query sebenarnya memiliki fungsi yang mirip dengan layar form
 - Dapat untuk menampilkan data
 - Dapat untuk melakukan entri data
- Data yang dientri melalui layar query akan menghasilkan isian tabel yang konsisten
- Misalnya data “Pr”, “Kb” dan “Kc” pada **Tbl_DataWilayah** diubah, maka field yang sama di **Tbl_KodeWilayah** akan mengikuti berubah

FORM: UNTUK ENTRI & TAMPILKAN DATA

- Jika jumlah field relatif banyak, entri data melalui 'Datasheet View' akan merepotkan
 - nama field belum tentu mencerminkan fieldnya
 - Tidak seluruh field dapat ditampilkan
 - Record (baris) yang ditampilkan relatif banyak
- Untuk menampilkan data per record, dapat digunakan 'Form'

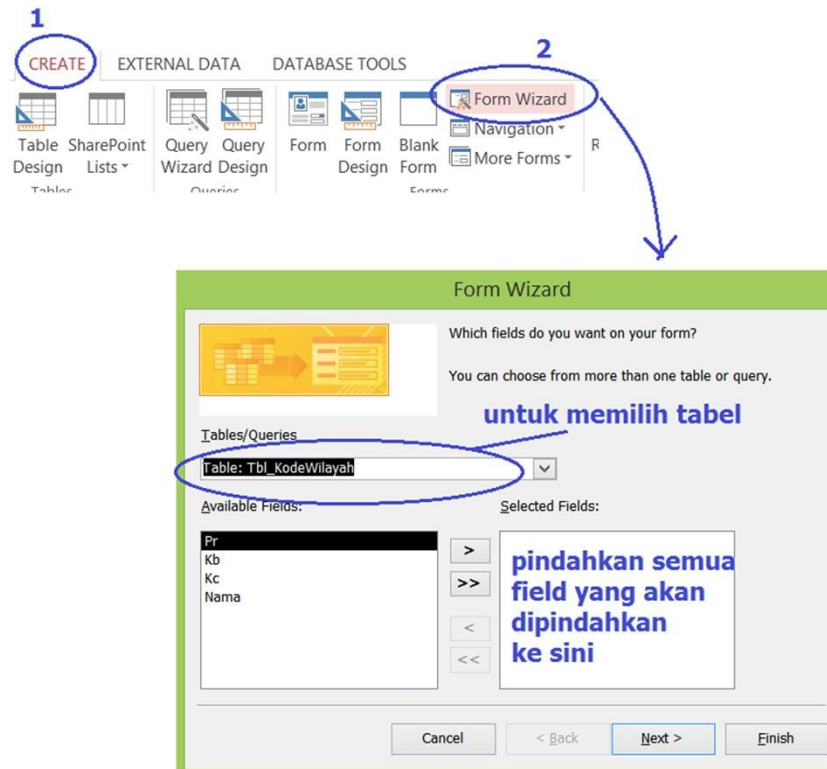
MEMBUAT FORM DENGAN WIZARD

(sebaiknya semua obyek ditutup terlebih dahulu dan **hanya tabel yang akan dibuatkan formnya yang dibuka**)

- Rangkaian langkah berikut mengandaikan pembuatan form untuk **Tbl_KodeWilayah**

MENGAKTIFKAN FORM WIZARD

- CREATE → Form Wizard
- Setelah semua field yang diinginkan dipindahkan ke kolom di sebelah kanan, klik 'Next'

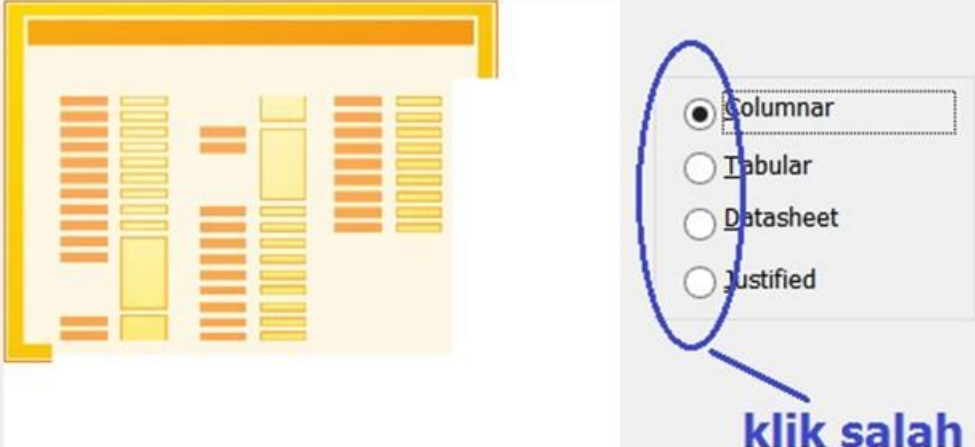


MEMILIH LAYOUT TAMPILAN

- Setelah dipilih layout tampilan, klik 'Next'

Form Wizard

What layout would you like for your form?



klik salah satu
untuk memilih
layout tampilan

Cancel < Back Next > Finish

MEMBERI JUDUL DAN MENAMPILKAN DATA

- Setelah diisikan judul Form dan dipilih salah satu opsi, klik 'Finish'

Form Wizard

What title do you want for your form?

Kode Wilayah

untuk menuliskan judul

That's all the information the wizard needs to create your form.

Do you want to open the form or modify the form's design?

☒ Open the form to view or enter information.

☐ Modify the form's design.

menampilkan data

Cancel < Back Next > Finish

TAMPILAN & ENTRI DATA

- Isian record tampak lebih rapi.
- Untuk melihat isian record lain, gunakan navigasi record.
- Untuk entri data: pergi ke record terakhir dan entri terus sampai muncul record kosong

The screenshot shows a web application window titled 'Kode Wilayah'. The form contains several input fields: 'Pr' (containing '33'), 'Kb' (containing '00'), 'Kc' (containing '000'), and 'Nama' (containing 'Jawa Tengah'). A blue circle highlights these input fields, with a line pointing to the text 'isi record'. Another blue circle highlights the record navigation controls at the bottom, which include 'Record: 1 of 7', navigation arrows, and a search bar. A line points from this circle to the text 'navigasi record'.

MENGENAL KOMPONEN FORM

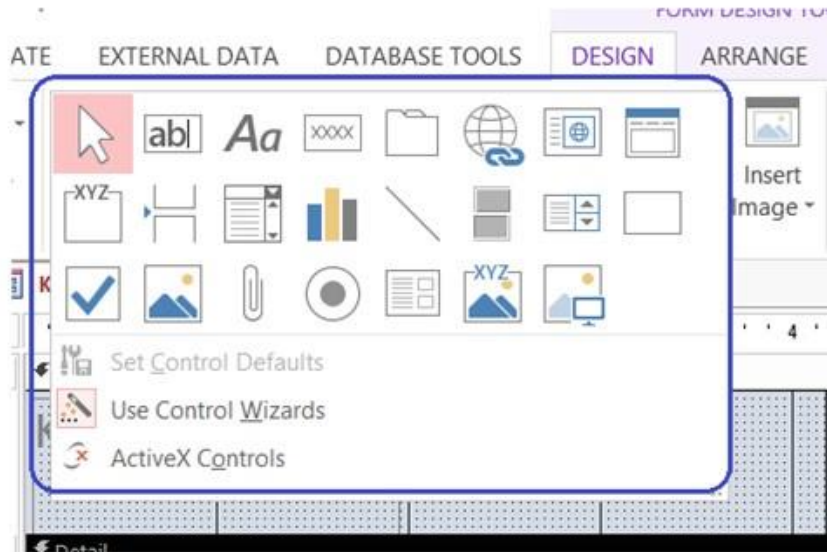
- HOME → Design View (layar rancangan berikut akan ditampilkan:

The screenshot shows an Access form in Design View. The form is divided into three main sections: Form Header, Detail, and Form Footer. The Form Header section contains a label 'Kode Wilayah' which is circled in blue. The Detail section contains a table with three columns: 'Kb', 'Kc', and 'Nama'. Each column has a corresponding text box. The 'Kb' and 'Kc' text boxes are circled in blue. The 'Nama' text box is also circled in blue. Annotations with blue lines point to these elements:

- A blue line points from the text 'label, tampilannya ('Caption') bisa disesuaikan' to the 'Kode Wilayah' label.
- A blue line points from the text 'Text Box, isiannya diambil dari isian record dari tabel terkait' to the 'Nama' text box.

- Untuk mengganti caption: klik-ganda di label yang dituju, klik captionnya, caption bisa langsung diganti

MENAMBAHKAN KOMPONEN LAIN



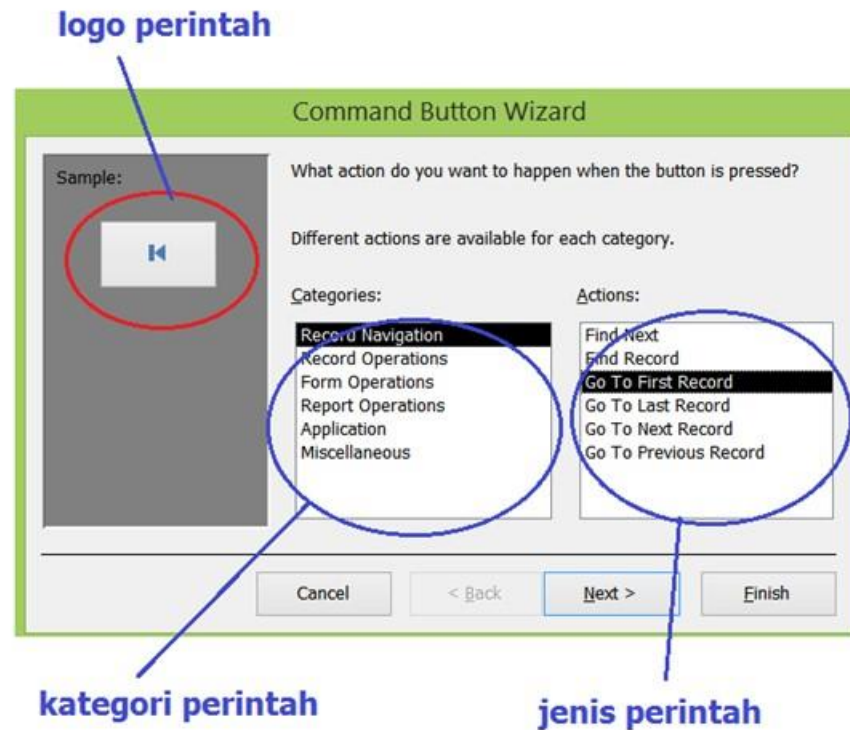
**Panel komponen
yang dapat ditambahkan
ke form**

**(drag ikon yang diinginkan
ke lokasi form yang sesuai)**

**Fungsi masing-masing ikon dapat
dilihat dari tip yang ditampilkan
ketika kursor digerakkan ke ikon
tertentu**

MENAMBAHKAN TOMBOL PERINTAH

- Klik ikon 'XXXX' (Button), gerakkan kursor ke lokasi di layar rancangan yang sesuai dan klik, akan muncul jendela Button Wizard:



- Setelah dipilih kategori dan jenis perintah, klik 'Next'

MEMILIH JENIS TAMPILAN DI TOMBOL

- Setelah dipilih jenis tampilan, klik 'Next'

pilihan untuk menampilkan teks atau gambar di tombol perintah

Command Button Wizard

Sample:

Do you want text or a picture on the button?

If you choose Text, you can type the text to display. If you choose Picture, you can click Browse to find a picture to display.

☐ Text: First Record
 ☒ Picture:

Arrow Up
 Go To First

 Browse...

☐ Show All Pictures

Cancel
 < Back
 Next >
 Finish

MEMBERIKAN NAMA TOMBOL

- Setelah diisi nama tombol, klik 'Finish'

Beriikan nama tombol yang
'bermakna', terutama untuk
kemudahan dalam pemrograman

Command Button Wizard

Sample:

What do you want to name the button?

A meaningful name will help you to refer to the button later.

FirstRec

That's all the information the wizard needs to create your command button. Note: This wizard creates embedded macros that cannot run or be edited in Access 2003 and earlier versions.

Cancel < Back Next > Finish

LATIHAN (± 15 MENIT)

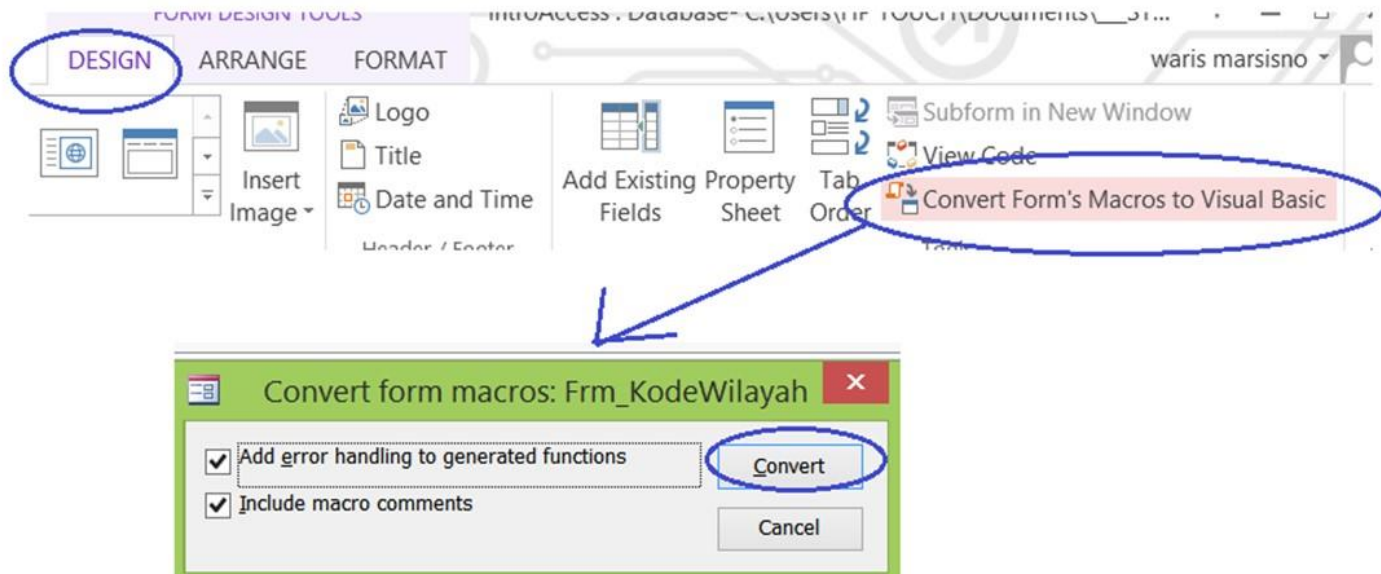
- Dengan cara yang sama, tambahkan tombol untuk:
 - Previous Record
 - Next Record
 - Last Record
 - Save Record
 - Delete Record

SETIAP PERINTAH DAPAT DIKONVERSI KE VBA

- Tombol-tombol navigasi (baik yang otomatis maupun yang dibuat oleh pengguna) pada dasarnya merupakan perintah macro yang dapat dikonversi ke VBA.
- Setiap tombol yang telah dibuat dalam latihan sebelumnya dapat dikonversikan ke VBA dan dapat dimodifikasi.

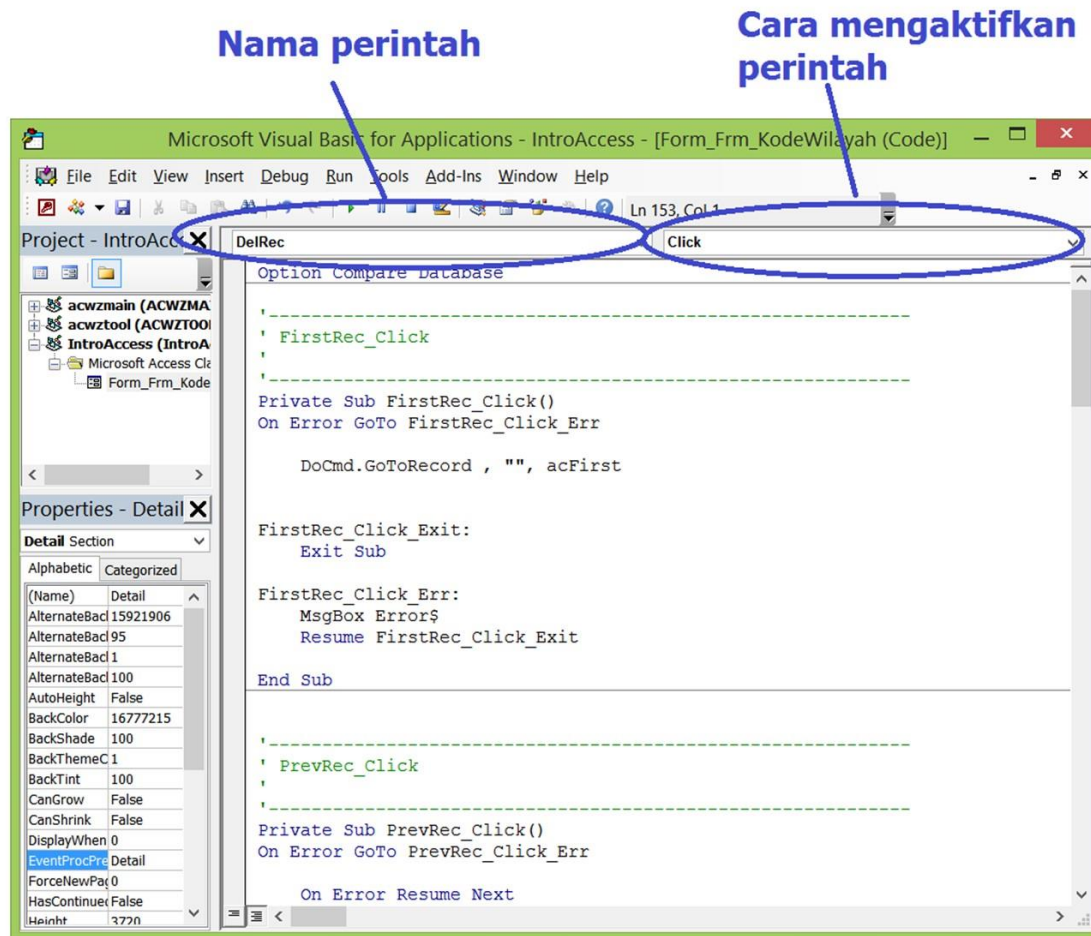
MELAKUKAN KONVERSI KE VBA

- DESIGN → Convert Form's Macros to Visual Basic



DAFTAR PERINTAH ATAU 'CODE'

- Untuk melihat perintah hasil konversi: klik ikon 'View Code'



MODIFIKASI PERINTAH

- Pilih tombol perintah 'FirstRec'

```
'-----  
' FirstRec_Click  
'  
'-----
```

```
Private Sub FirstRec_Click()  
On Error GoTo FirstRec_Click_Err
```

```
DoCmd.GoToRecord , "", acFirst  
MsgBox "Ke record pertama?"
```

```
FirstRec_Click_Exit:  
Exit Sub
```

```
FirstRec_Click_Err:  
MsgBox Error$  
Resume FirstRec_Click_Exit
```

```
End Sub
```

Ketik perintah ini.

Setelah itu klik ikon bergambar kunci untuk kembali ke layar Access.

Kemudian: View → Form View

Kemudian coba klik 'First Record'

Apa yang terjadi?

BEBERAPA CATATAN TENTANG PEMROGRAMAN ACCESS VBA

- Program bersifat modular yang terdiri dari prosedur-prosedur:
 - Sub rutin
 - Function
- Aktivasi prosedur bersifat 'event driven'
 - Dengan klik, enter, dan sebagainya

TENTANG ACCESS VBA

- Visual Basic for Application (VBA) merupakan bahasa pemrograman yang tertanam dalam Access.
- Untuk dapat menggunakan secara optimal diperlukan **pengetahuan tentang Access:**
 - Merupakan sistem manajemen database.
 - Obyek data base yang dikelola: **tabel, query, form, report, macro, modul.**

VBA MERUPAKAN BAHASA PEMROGRAMAN

- Berupa kode instruksi untuk dieksekusi oleh komputer.
- Programming = pemrograman = penulisan program = penulisan kode.
- Penulisan kode atau instruksi mengikuti sintaksis tertentu.

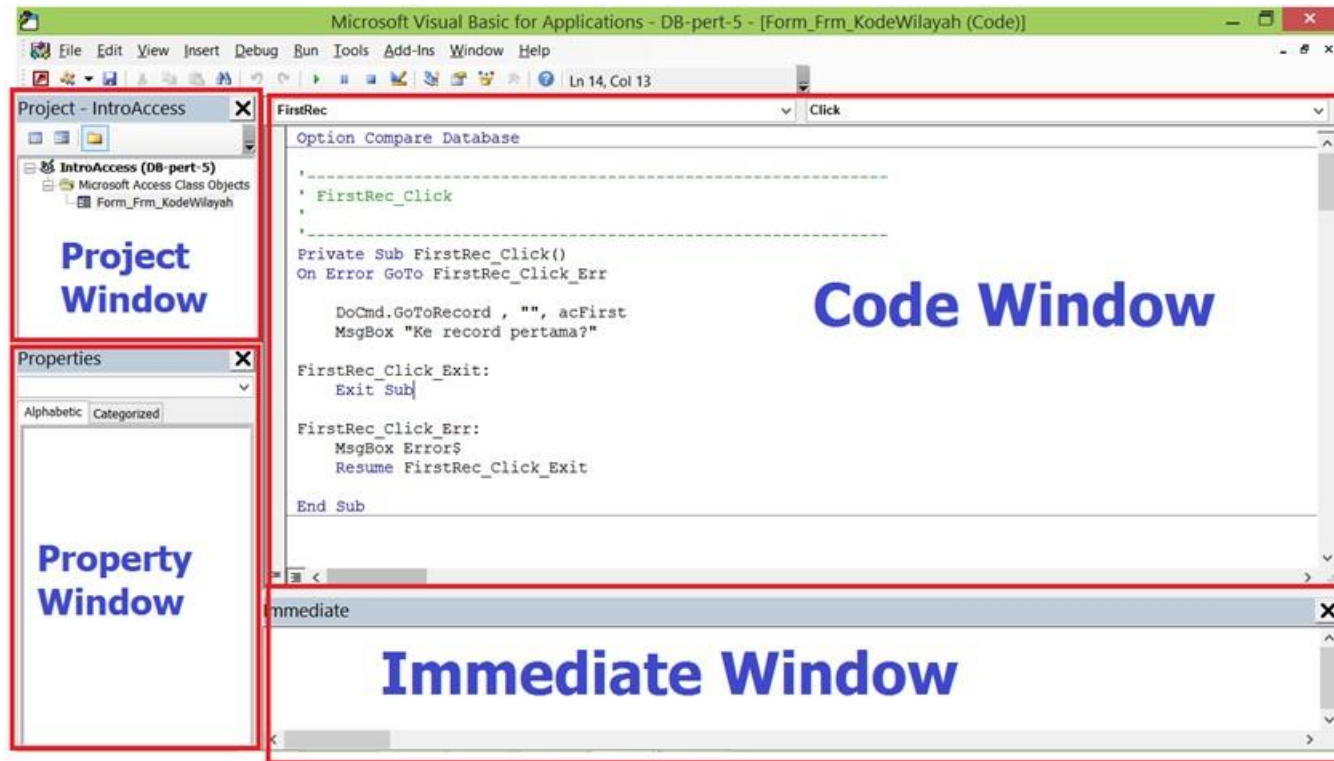
TEMPAT PENYIMPANAN PROGRAM: MODUL

- Program untuk Access VBA disimpan dalam modul.
- **Standard module:** program VBA yang dapat diakses oleh semua obyek yang ada dalam suatu database.
- **Class module:** program VBA yang hanya dapat diakses dari kelas obyek tertentu, misalnya program yang hanya berlaku untuk form, report dan panel navigasi.

STANDARD VS CLASS MODULES

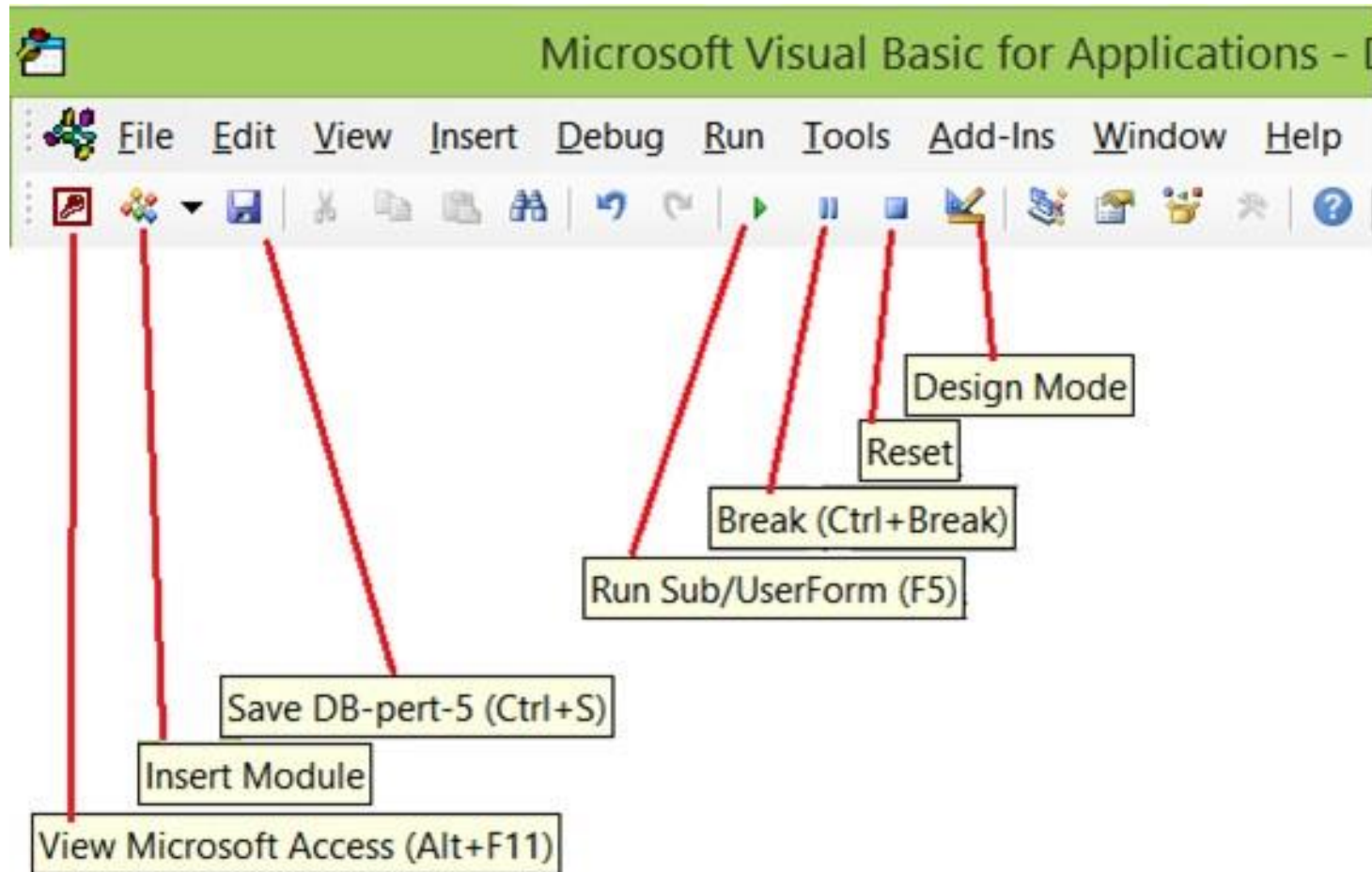
- ◉ **Standard module:** semua prosedur di dalam modul ini dapat dijalankan (*di-run*) dari semua obyek yang ada di dalam database.
- ◉ **Class module:** prosedur di dalam modul ini hanya dapat dijalankan dari obyek di mana modul dibuat.
- ◉ **Semua modul Access VBA** berisi perintah atau kode VBA

LAYAR VB EDITOR



- ◉ **Code Window:** untuk menuliskan program
- ◉ **Immediate Window:** Menampilkan hasil perintah pendek (untuk uji coba saja)

Beberapa menu dalam VBA window

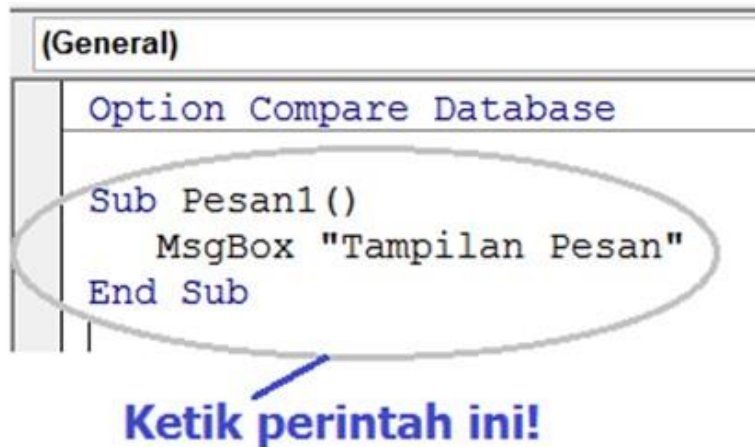


MENULIS PROGRAM DALAM MODUL

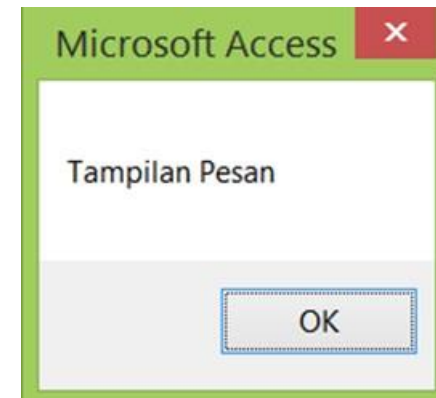
- Tutup semua obyek.
- Jalankan menu: CREATE → Module
(Gantilah nama modul di 'Property Window' dengan 'Contoh')
- Tuliskan program di 'Code Window' dan lihat hasilnya di 'Immediate Window'

CONTOH 1

1. Di 'Code Window'



3. Output



2. Di 'Immediate Window'



CONTOH 2

- Di 'Code Window' tuliskan:

```
Function HslKali(x As Single, y As Single) As _  
Single
```

```
    HslKali = x * y
```

```
End Function
```

- Di 'Immediate Window' ketik:

```
?HslKali(7,8)
```

kemudian tekan ENTER

Dibawahnya akan ditampilkan output: 56

CONTOH 3

Code Window

```
Sub TampilanHsl()  
    Dim z As Integer  
    Dim strHsl As String  
    z = HslKali(7, 8)  
    strHsl = Str(7) + " * " + Str(8) + " = " + Str(z)  
    lyr = MsgBox(strHsl, , "Hasil Kali")  
End Sub
```

Immediate Window

Immediate
TampilanHsl

Output (Message Box)



PERPINDAHAN ANTARA CODE WINDOW DAN ACCESS WINDOW

- Pindah ke atau membuka 'Code Window' ketika 'Access Window' terbuka:
ALT-F11
- Pindah ke atau membuka 'Access Window' ketika 'Code Window' terbuka:
ALT-F11

MENJALANKAN PROGRAM DARI 'CODE WINDOW'

- Selain dijalankan dari 'Immediate Window', untuk menjalankan subrutin dapat pula dilakukan dengan cara:
 - Gerakkan kursor ke subrutin yang akan dijalankan (di antara Sub..End Sub)
 - Tekan F5 atau ikon Run (▶).

```
Sub TampilanHsl()  
Dim z As Integer  
Dim strHsl As String  
z = HslKali(7, 8)  
strHsl = Str(7) + " * " + Str(8) + " = " + Str(z)  
lyr = MsgBox(strHsl, , "Hasil Kali")  
End Sub
```

Letakkan kursor di salah
satu baris perintah antara
Sub ... End Sub.

Kemudian tekan F5
atau tekan ikon ▶ .

PROSEDUR DALAM ACCESS VBA

- Setiap program Access VBA ditulis dalam prosedur (*procedure*).
- Struktur prosedur:

```
Sub / Function / Property
```

```
    'Perintah' atau 'Kelompok perintah'
```

```
End Sub / Function / Property
```

- Perintah dieksekusi/dijalankan secara berurutan: dari baris pertama sampai baris terakhir dalam prosedur

JENIS-JENIS PROSEDUR ACCESS VBA

Terdapat 4 hal terkait dengan prosedur dalam Access VBA:

1. Subrutin
2. Fungsi
3. Event
4. Property

PROSEDUR SUBRUTIN

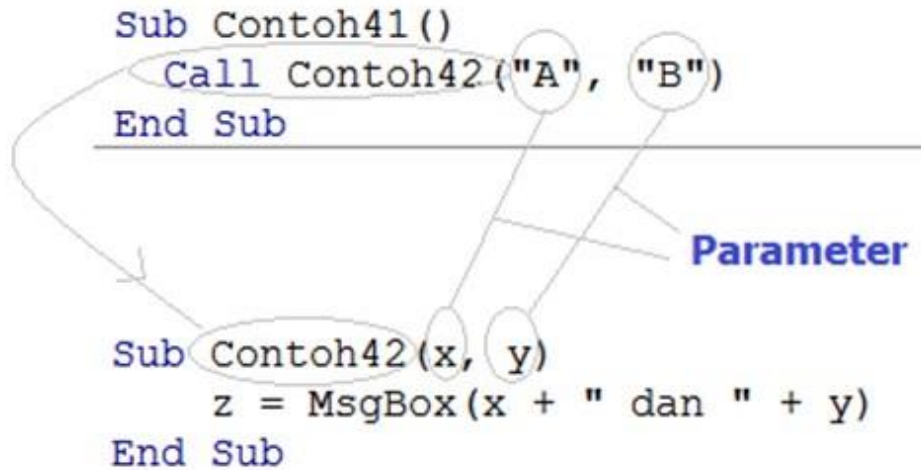
```
Sub NamaSubrutin ([parameters])
```

```
    Perintah atau kelompok perintah
```

```
End Sub
```

- *NamaSubrutin*: pengenal yang digunakan untuk memanggil atau mengaktifkan Subrutin
- *Parameter*: variabel yang digunakan untuk mentransfer data antar subrutin
- Pemanggilan subrutin: Call *Nama*(*par*)

CONTOH 4



Ketika Contoh42 dieksekusi:

Nilai x = "A"

Nilai y = "B"



PROSEDUR FUNGSI

Function *NamaFungsi* (*par-1 As Jenis-data,...*)_
As Jenis-data

Perintah atau *Kelompok Perintah*

End **Function**

Pemanggilan fungsi:

- Melalui ekspresi variabel
- Hasilnya disimpan di dalam variabel

CONTOH 5

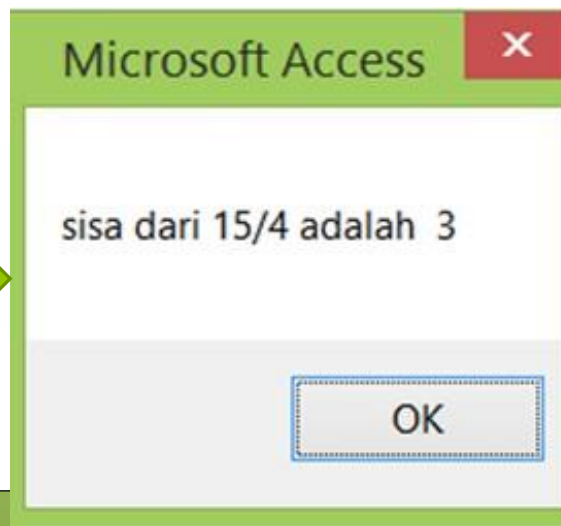
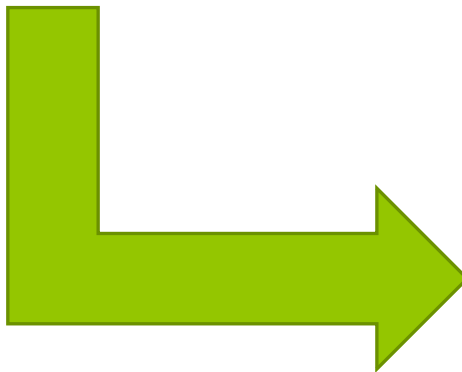
```
Sub Contoh51()  
    Dim modHsl As Integer  
    modHsl = Contoh52(15, 4)  
    z = MsgBox("sis dari 15/4 adalah " + Str(modHsl))  
End Sub
```

pemanggilan

parameter

```
Function Contoh52(p, q) As Integer  
    Contoh52 = p Mod q  
End Function
```

return value



- Fungsi pada dasarnya merupakan variabel
- Nilainya adalah sama dengan *return value*

PROSEDUR EVEN

- Prosedur Even: prosedur yang dieksekusi jika even tertentu terjadi
- Contohnya ketika Button atau tombol tertentu di-klik pada suatu form
- Isian dari prosedur even adalah kombinasi dari prosedur fungsi dan prosedur subrutin

PROSEDUR PROPERTY

- Prosedur property digunakan untuk mengatur nilai property dari form, report dan kelas modul
- Terdapat tiga jenis prosedur property
 - Property Get: memperbolehkan aplikasi lain untuk membaca nilai property
 - Property Let: memperbolehkan aplikasi untuk mengatur property
 - Property Set: fungsinya mirip dengan Property Let

VARIABEL

- ◉ Dalam pemrograman, **variabel** adalah suatu nama yang diberikan untuk referensi atau pengenalan terhadap suatu item dari data.
- ◉ Untuk membaca data input atau menampilkan output, diperlukan variabel untuk penyimpanan data tersebut.

NAMA VARIABEL ACCESS VBA

- Boleh mengandung semua jenis karakter, kecuali karakter berikut:
 . # \$ % & @ !
- Tidak boleh memuat spasi.
- Tidak boleh diawali dengan spasi atau bilangan (angka).
- Boleh sampai 254 karakter, tapi sebaiknya jangan terlalu panjang.
- Tidak boleh berupa *reserved word*.
- Karena Access memiliki banyak obyek, sebaiknya nama variabel mengindikasikan jenis obyek yang diwakili. Misalnya str untuk menunjukkan variabel string.
- VBA tidak case sensitif.

JENIS DATA

- Jenis data menentukan cara komputer menyimpan data di dalam memori.

Data Type	Storage Size	Range
Byte	1 byte	0 to 255
Boolean	2 bytes	True or False
Integer	2 bytes	-32,768 to 32,767
Long (long integer)	4 bytes	-2,147,483,648 to 2,147,483,647
Single (single-precision floating-point)	4 bytes	-3.402823E38 to -1.401298E-45 for negative values; 1.401298E-45 to 3.402823E38 for positive values
Double (double-precision floating-point)	8 bytes	-1.79769313486231E308 to -4.94065645841247E-324 for negative values; 4.94065645841247E-324 to 1.79769313486232E308 for positive values
Currency (scaled integer)	8 bytes	-922,337,203,685,477.5808 to 922,337,203,685,477.5807

JENIS DATA

Data Type	Storage Size	Range
Decimal	14 bytes	+/-79,228,162,514,264,337,593,543,950,335 with no decimal point; +/-7.9228162514264337593543950335 with 28 places to the right of the decimal; smallest nonzero number is +/-0.00000000000000000000000000000001
Date	8 bytes	January 1, 100, to December 31, 9999
Object	4 bytes	Any Object reference
String (variable-length)	10 bytes + string length	0 to approximately 2 billion
String (fixed-length)	Length of string	1 to approximately 65,400
Variant (with numbers)	16 bytes	Any numeric value up to the range of a Double
Variant (with characters)	22 bytes + string length	Same range as for variable-length String
User-defined (using Type)	Number required by elements	The range of each element is the same as the range of its data type.

- Untuk jenis data variant, VBA akan menyesuaikan jenis data yang disimpan dan akan memilih jenis yang sesuai

VARIABEL EKSPLISIT DAN IMPLISIT

- ◉ **Variabel eksplisit:** yang jenis datanya dideklarasikan secara eksplisit.
- ◉ **Variabel implisit:** variabel yang tidak dideklarasikan jenis datanya. Diperlakukan sebagai variabel dengan jenis data variant.
- ◉ Walaupun pendeklarasian jenis data variant memudahkan pemrogram, karena jenis data menyesuaikan dengan data yang ada, tetapi sebaiknya dihindari agar tidak terjadi salah operasi aritmetika.

PENDEKLARASIAN VARIABEL: DIM

- Digunakan perintah Dim:

Dim nama-var As jenis-data

- Jika jenis data tidak dituliskan, VBA akan memperlakukannya sebagai jenis data *variant*.

Contoh:

Dim Var5 As Integer

Dim Var10 As String

→ Var5 dideklarasikan sebagai variabel *integer*

Var10 dideklarasikan sebagai variabel *string*.

ALTERNATIF DEKLARASI VARIABEL

- Dengan menggunakan karakter khusus di akhir nama-variabel:

% untuk jenis data **integer**

& untuk jenis data long **integer**

! Untuk jenis data **single**

untuk jenis data **double**

@ untuk jenis data **currency**

\$ untuk jenis data **string**

Contoh:

Dim var2\$ → mendeklarasikan var2 sebagai variabel string.

OPERATOR

- Operator: merupakan elemen perintah untuk melakukan suatu operasi terhadap satu atau lebih variabel yang memiliki nilai.
- Terdapat 4 jenis operator dalam VBA
 1. Operator aritmetika
 2. Operator perbandingan
 3. Operator penggabungan
 4. Operator logika dan bitwise.

OPERATOR ARITMETIKA

$+$: penjumlahan

$-$: pengurangan atau negasi bilangan

$*$: perkalian

$/$: pembagian

\wedge : perpangkatan

\backslash : pembagian bilangan bulat

Mod : modulo, sisa hasil bagi

Urutan eksekusi: (\wedge) , $(*, /)$, (\backslash) , (Mod) , $(+, -)$

OPERATOR PEMBANDINGAN

- Operator perbandingan (comparison) digunakan untuk membandingkan dua ekspresi dan menghasilkan nilai Boolean (True atau False)
- Jenis operator perbandingan:
 - = kesamaan
 - <> ketaksamaan
 - < lebih kecil
 - > lebih besar
 - <= lebih kecil atau sama dengan
 - >= lebih besar atau sama dengan

MEMBANDINGKAN STRING

“73” < “9” → **True**, karena karakter pertama di string pertama lebih kecil dibandingkan karakter pertama di string kedua.

“aa” < “a” → **False**, untuk karakter yang sama, yang lebih panjang diperlakukan sebagai lebih besar.

“aa” < “ab” → **True**.

“ab” < “aab” → **False**.

OPERATOR PENGGABUNGAN (*CONCATENATION*)

- Menggabungkan dua atau lebih string menjadi satu string.
- Terdapat dua operator yang dapat digunakan, yaitu “&” dan “+” dan keduanya akan memberikan hasil yang sama.
- Contoh:
$$\text{strA} = \text{“Ab”} + \text{“cde”} \rightarrow \text{strA} = \text{“Abcde”}$$
$$\text{strA} = \text{“Ab”} \& \text{“cde”} \rightarrow \text{strA} = \text{“Abcde”}$$
- Dianjurkan untuk menggunakan operator “&” yang secara khusus memang dirancang untuk operasi penggabungan.

OPERATOR LOGIKA/BITWISE

- Digunakan untuk **membandingkan dua ekspresi Boolean** dan **menghasilkan nilai Boolean**.

And : logika konjungsi (penggabungan) dari dua ekspresi Boolean.

$T \text{ And } T = T$

$T \text{ And } F = F$

$F \text{ And } T = F$

$F \text{ And } F = F$

OPERATOR LOGIKA/BITWISE

Not : negasi dari ekspresi Boolean.

$$y = \text{Not } 20 < 30 \rightarrow y = F$$

$$x = \text{Not } 20 < 10 \rightarrow x = T$$

Or : disjungsi dari dua ekspresi Boolean.

$$T \text{ Or } T = T$$

$$T \text{ Or } F = T$$

$$F \text{ Or } T = T$$

$$F \text{ Or } F = F$$

OPERATOR LOGIKA/BITWISE

Xor : eksklusif dari dua ekspresi Boolean.

$$T \text{ Xor } T = F$$

$$T \text{ Xor } F = T$$

$$F \text{ Xor } T = T$$

$$F \text{ Xor } F = F$$

AndAlso : konjungsi logika short circuit terhadap dua ekspresi Boolean. Evaluasi ekspresi kedua hanya dilakukan jika ekspresi pertama T.

$$T \text{ AndAlso } T = T$$

$$T \text{ AndAlso } F = F$$

$$F \text{ (yang kedua tidak dievaluasi)} = F$$

OPERATOR LOGIKA/BITWISE

OrElse : kebalikan dari operator AndAlso.

T (yg kedua tdk dievaluasi) = T

F OrAlso T = T

F OrAlso F = F

IsFalse : menguji apakah suatu ekspresi salah.

IsTrue : menguji apakah suatu ekspresi benar.

EKSPRESI

- Ekspresi dalam pemrograman adalah sederetan nilai yang dikombinasikan dengan operator untuk memperoleh nilai yang baru.
- Ekspresi bisa melibatkan bilangan atau nilai data tertentu:

$5 + 5^2 \rightarrow$ hasilnya 30

“Selamat” & “ Siang” \rightarrow hasilnya “Selamat Siang”

- Bisa juga melibatkan variabel:

strHari + “Selasa” \rightarrow jika strHari = variabel string

MEMBERIKAN NILAI KE VARIABEL

- Terdapat dua cara untuk memberikan nilai (*value assignment*) ke suatu variabel

- Memberikan nilai

Dim x As Single

x = 2.45678

- Menggunakan ekspresi (bisa melibatkan fungsi)

Dim p, q, r As Single

p = 2.45678

q = 3.654

r = p * q

BARIS KOMENTAR

- Jika program semakin kompleks, seringkali diperlukan catatan tentang setiap fungsi dan subrutin yang dibuat. Begitu juga dengan variabel-variabelnya.
- Dalam VBA, catatan dapat diberikan pada baris komentar (*comment line*).
- Baris komentar dalam VBA ditandai dengan *single apostrophe* di kolom pertama suatu baris.
- Semua yang tertulis di baris komentar tidak dieksekusi oleh komputer.

LINGKUP VARIABEL DALAM VBA

- Secara umum lingkup variabel VBA dapat dibedakan menjadi tiga:
 - Lingkup prosedur atau fungsi (**variabel lokal**)
 - Lingkup modul
 - Lingkup database (**variabel global**)

VARIABEL LOKAL (LINGKUP PROSEDUR ATAU FUNGSI)

- Dideklarasikan dengan perintah 'Dim' di antara **Sub ... End Sub** atau di antara **Function ... End Function**.
- Hanya berlaku di lingkungan subrutin atau fungsi di mana variabel didefinisikan.
- Tidak dikenal di lingkungan subrutin atau fungsi lain.

VARIABEL LINGKUP MODUL

- Dideklarasikan dengan perintah 'Private' atau 'Dim' sebelum prosedur atau fungsi didefinisikan.
- Hanya berlaku di lingkungan modul yang bersangkutan.

VARIABEL GLOBAL (LINGKUP SEMUA OBYEK DI DALAM DATABASE)

- Dideklarasikan dengan 'Public' di area sebelum pendefinisian subrutin dan fungsi di suatu modul.
- Variabel ini berlaku untuk seluruh obyek di dalam database.
- 'Public' dapat digunakan untuk mendeklarasikan fungsi yang berlaku umum di suatu database.

CONTOH 6 & CONTOH 7

VARIABEL LOKAL DAN GLOBAL

Modul: Contoh7

```
Sub Coba()
  Call hitungSqrRt(1, 2, 3)
End Sub
```

Modul: Contoh6

variabel lokal

subrutin publik

```
Public Sub hitungSqrRt(a As Single, b As Single, c As Single)
  Dim varDet As Single
  varDet = functDet(a, b, c)
  If varDet < 0 Then
    MsgBox ("Determinan negatif!" & vbCrLf & "Tidak bisa dihitung akarnya!!")
  Else
    Call subSqrRt(a, b, varDet)
  End If
End Sub
```

```
Function functDet(x, y, z) As Single
  functDet = y * y - 4 * x * z
End Function
```

variabel lokal

```
Sub subSqrRt(p, q, inpDet)
  Dim x1, x2 As Single
  x1 = (-q + inpDet) / 2 * p
  x2 = (-q - inpDet) / 2 * p
  MsgBox ("x1 = " & Str(x1) & vbCrLf & "x2 = " & Str(x2))
End Sub
```

TRANSFER DATA ANTAR SUBROUTIN DAN FUNGSI

- Subrutin dan fungsi seringkali dipanggil dari subrutin dan fungsi lainnya.
- Transfer data dilakukan melalui variabel yang menjadi argumen dari subrutin dan fungsi.
- Nama variabel untuk memanggil tidak harus sama dengan yang dipanggil.
- Yang diperhatikan adalah urutan letak variabel.

```
varDet = functDet(a, b, c)
```

nilai x = a
nilai y = b
nilai z = c

```
Function functDet(x, y, z) As Single  
    functDet = y * y - 4 * x * z  
End Function
```

KONSTANTA

- Nilai dari suatu variabel dapat berubah ketika program dieksekusi.
- Jika diinginkan sesuatu yang tidak berubah nilainya (misal jumlah sampel), maka dapat digunakan konstanta.
- Untuk mendeklarasikan konstanta digunakan perinth 'Const'

```
Const Jawaban = 'setuju' As String
```

```
Const nSample = 100
```

```
Const pajak = 0.05
```

KONSTANTA DENGAN ENUM

- Konstanta sejenis yang mengacu ke suatu obyek tertentu dapat dideklarasikan dengan menggunakan Enum.
- Contoh

Publik **Enum Bunga**

Koperasi = 0.01

Bank = 0.015

End Enum

- Contoh penggunaan

$Y = \text{modal} * \text{Bunga.Koperasi}$

MSGBOX

- MsgBox merupakan salah satu fungsi yang tersedia dalam Access VBA.
- Kegunaannya adalah untuk menampilkan informasi kepada pengguna melalui suatu kotak dialog.
- Pengguna tidak bisa mengetikkan apa pun ke dalam kotak dialog, kecuali merespon pertanyaan yang pada umumnya dimunculkan.
- Terdapat dua jenis kotak dialog yang umum digunakan:
 - Hanya menampilkan suatu pesan.
 - Meminta pengguna untuk memutuskan sesuatu.

SINTAKSIS MSGBOX

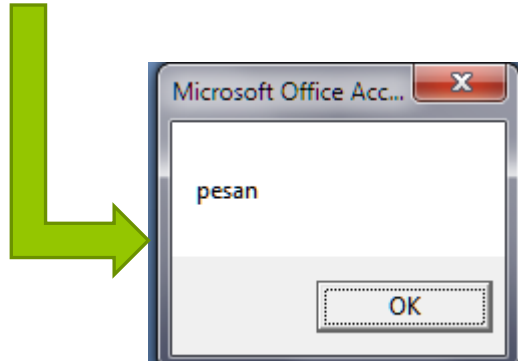
- Sintaksis fungsi MsgBox:

```
MsgBox ([Message], [Buttons], [Title],  
[HelpFile], [Context])
```

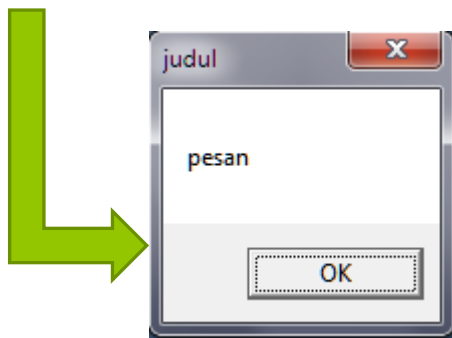
- *Message*: string yang ditampilkan
- *Buttons*: jenis kelompok tombol (*button*) yang ditampilkan.
- *Title*: string, ditampilkan sebagai judul kotak.
- *HelpFile*: string, nama file yang berfungsi sebagai bantuan penjelasan.
- *Context*: nomor yang terkait dengan help. Jika diberikan helpfile maka context juga harus diberikan.
- Argumen yang wajib ada adalah 'Message'

BEBERAPA CONTOH MSGBOX

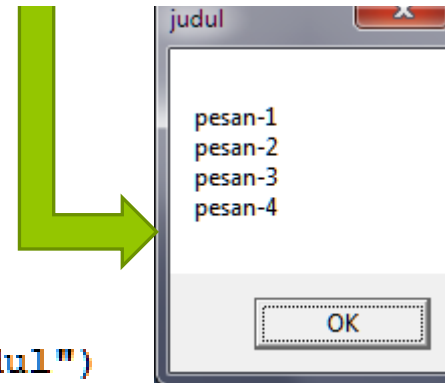
```
Sub subMsg0 ()
    MsgBox "pesan"
End Sub
```



```
Sub subMsg ()
    x = MsgBox("pesan", vbOKOnly, "judul")
End Sub
```



```
Sub subMsg1 ()
    x = MsgBox("pesan-1" & vbCrLf & _
        "pesan-2" & vbCrLf & "pesan-3" & _
        vbCrLf & "pesan-4", vbOKOnly, "judul")
End Sub
```



vbCrLf: pindah ke baris berikutnya.

Garis bawah (*underscore, _*) di kolom terakhir digunakan untuk menyambung perintah di baris berikutnya)

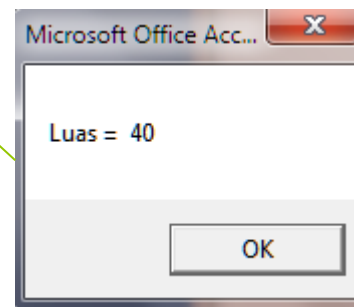
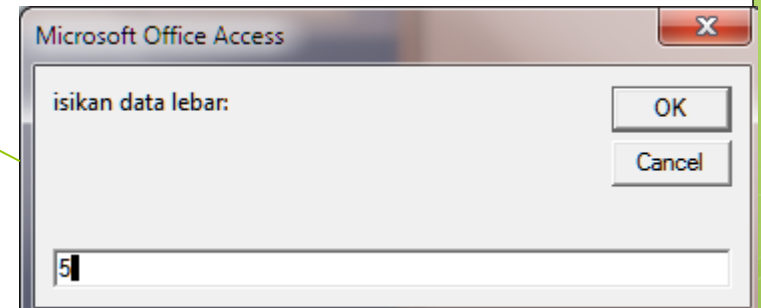
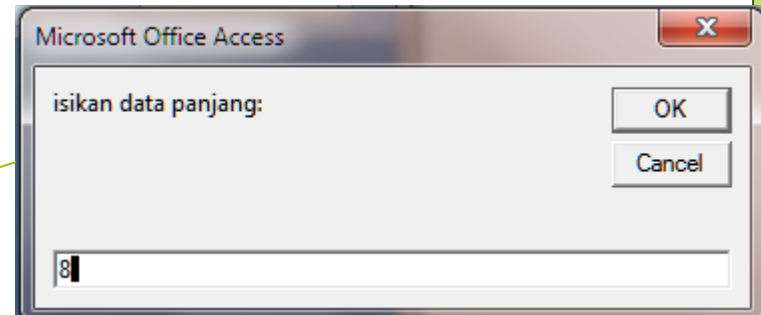
INPUT BOX

- Merupakan fungsi yang digunakan untuk membaca input yang diberikan oleh pengguna.
- Input dibaca setelah tombol 'OK' diklik.
- Jenis data dari fungsi ini adalah string.
- Sintaksisnya:

InputDialog(message[,title][,default][,xpos][,ypos] [,helpfile,context])

CONTOH 8

```
Sub HitungLuas()  
    Dim x, y As Single  
    x = Str(InputBox("isikan data panjang: "))  
    y = Str(InputBox("isikan data lebar: "))  
    z = MsgBox("Luas = " & Str(x * y))  
End Sub
```



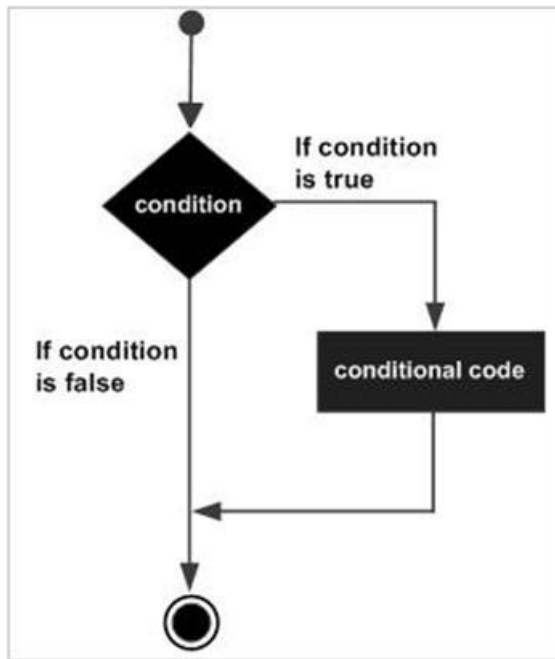
EKSEKUSI PERINTAH VBA

- Eksekusi perintah dalam subrutin dan fungsi dilakukan secara berurutan (sekuensial) dari awal sampai akhir.
- Jika diinginkan sekelompok perintah hanya dieksekusi untuk kondisi tertentu yang terpenuhi maka dapat digunakan percabangan.

JENIS PERCABANGAN

- If...Then...
- If ... Then End If
- If ... Then ... Else ... End If
- Percabangan bersarang ...
- Select Case ... End Select
- Select Case Is ... End Select

IF ... THEN ...

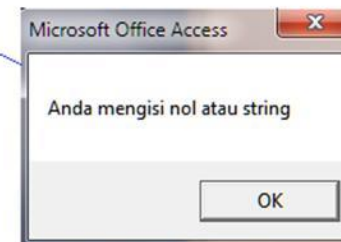
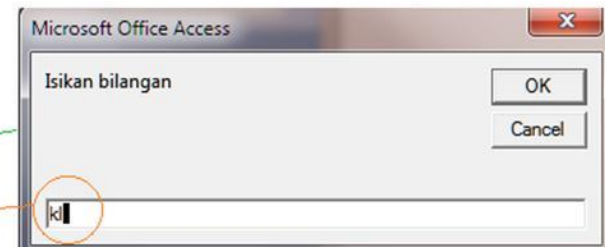


If kondisi Then perintah

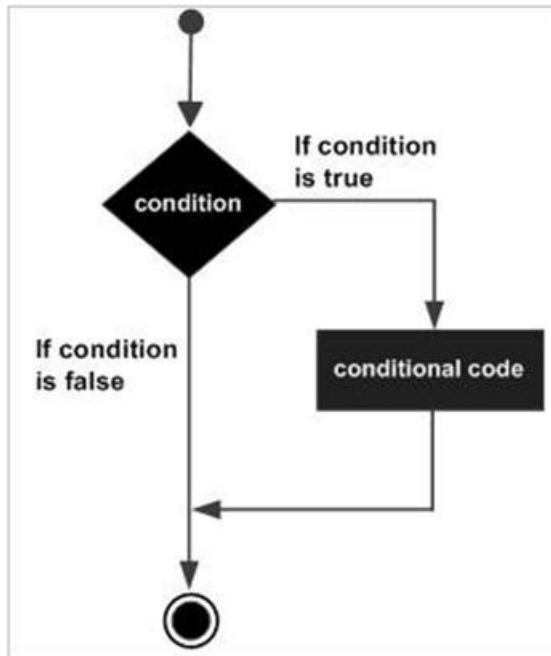
- Kondisi = operasi logika dan memiliki nilai Boolean 'True' atau 'False'.
- Kondisi boleh merupakan memuat operasi logika gabungan, misalnya
If $a < 5$ And $b > 2$ Then $c = a / b$
- Jika kondisi bernilai 'True' (benar) perintah setelah Then akan dieksekusi.
- Jika tidak benar, eksekusi langsung dilanjutkan ke perintah berikutnya.

CONTOH 9

```
Sub Contoh91()  
  Dim inpX As String  
  inpX = InputBox("Isikan bilangan")  
  If Val(inpX) <= 0 Then MsgBox "Anda mengisi nol atau string"  
End Sub
```



IF ... THEN ... END IF



If (*boolean_expression*) Then

Statement 1

....

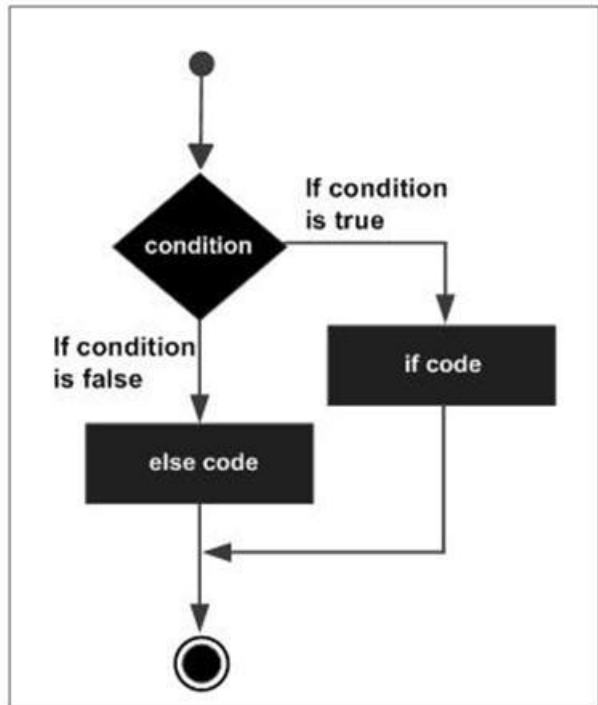
....

Statement n

End If

Jika perintah setelah Then lebih dari satu, maka harus diakhiri dengan End If.

IF ... THEN ... ELSE ... END IF



If (*boolean_expression*) Then

Statement 1

.....

Statement n

Else

Statement 1

.....

Statement n

End If

IF ... THEN ... ELSEIF ... END IF

```
If (boolean_expression) Then
    Statement 1
    ....
    Statement n
Elseif (boolean_expression) Then
    Statement 1
    ....
    Statement n
Elseif (boolean_expression) Then
    Statement 1
    ....
    Statement n
Else
    Statement 1
    ....
    Statement n
End If
```

PERCABANGAN BERSARANG

If (*boolean_expression*) Then

Statement 1

.....

Statement n

If (*boolean_expression*) Then

Statement 1

.....

Statement n

Elseif (*boolean_expression*) Then

Statement 1

.....

Statement n

Else

Statement 1

.....

Statement n

End If

Else

Statement 1

....

Statement n

End If

SELECT CASE ... END SELECT

Select Case *ekspresi*

 Case *nilai-ekspresi-1*

kelompok-perintah-1

 Case *nilai-ekspresi-2*

kelompok perintah-2

 :

 Case Else

kelompok-perintah-else

End Select

ARRAY

- ◉ Array adalah variabel yang dapat menyimpan lebih dari satu nilai.
- ◉ Setiap elemen array dapat memiliki jenis datanya sendiri-sendiri.
- ◉ Deklarasi array:
 - ◉ Dim *nama-array* (*jumlah elemen max*)
as *tipe_data*
 - ◉ *Jumlah elemen max*: harus bilangan bulat.
 - ◉ Elemen arraynya mulai dari 0 s.d.
Jumlah elemen max

ARRAY

Contoh:

- ◉ Dim arrX(3) as Integer

- arrX(0), arrX(1), arrX(2), arrX(3)
semuanya integer

- ◉ Dim arrY(1)

- arrY(0) = 0 'jenis data number

- arrY(1) = "ya" 'jenis data string

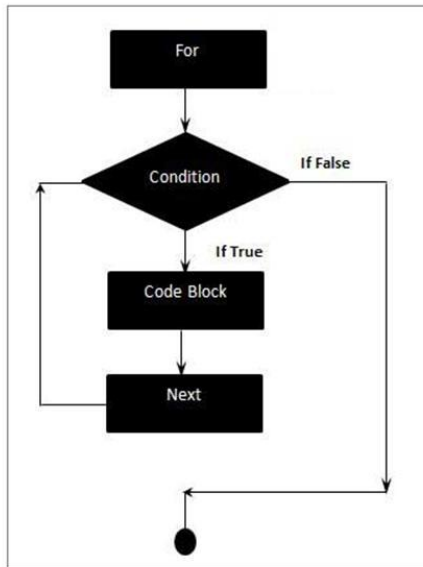
DIMENSI ARRAY

- Array bisa memiliki lebih dari satu dimensi.
- Umumnya digunakan array dua dimensi untuk merepresentasikan tabel.

CARA LAIN MENDEFINISIKAN ARRAY

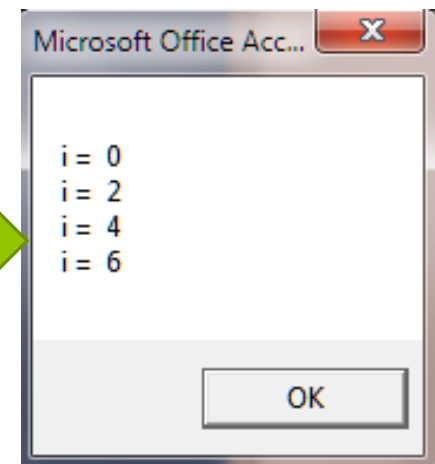
- Buah = Array("Mangga", "Pisang", "Jambu")
- Buah merupakan array yang memiliki tiga elemen:
 - Buah(0) = "Mangga"
 - Buah(1) = "Pisang"
 - Buah(2) = "Jambu"

FOR



```
For counter = Start to End  
  Loop Statements  
Next
```

```
Sub For_demo()  
  Dim a As Integer  
  
  Dim outMsg As String  
  
  outMsg = ""  
  a = 7  
  
  For i = 0 To a Step 2  
    outMsg = outMsg & "i = " & i & vbCrLf  
  Next  
  MsgBox outMsg  
End Sub
```



FOR

```
'FOR
Sub For_demo()
    Dim a As Integer

    Dim outMsg As String

    outMsg = ""
    a = 7

    For i = 0 To a Step 2
        outMsg = outMsg & "i = " & i & vbCrLf
    Next
    MsgBox outMsg
End Sub
```

FOR EACH ITEM IN ...

- Digunakan untuk melakukan pengulangan berdasarkan kondisi isian dari suatu array string

```
'cara lain deskripsi array
Sub strArray()
    Buah = Array("mangga", "pisang", "jambu")
    Dim Buahhan As String
    Buahhan = ""

    For Each Item In Buah
        Buahhan = Buahhan & Item & vbCrLf
    Next
    MsgBox Buahhan
End Sub
```

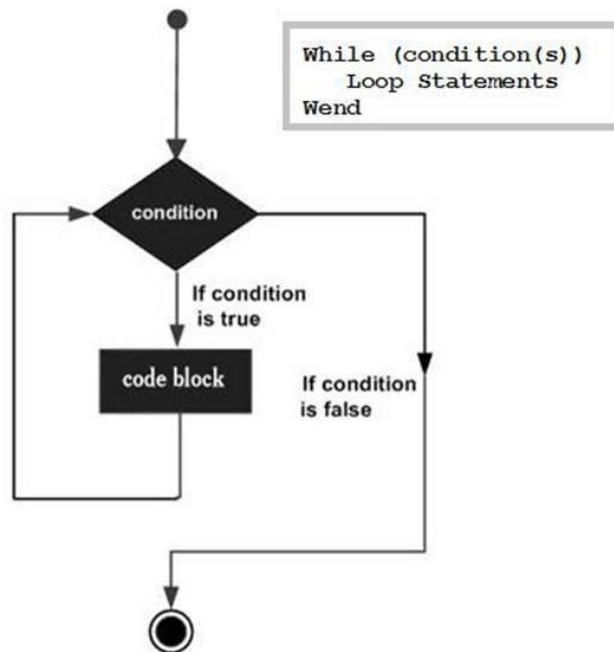


FOR EACH ITEM IN ...

```
'cara lain deskripsi array
Sub strArray()
    Buah = Array("mangga", "pisang", "jambu")
    Dim Buahan As String
    Buahan = ""

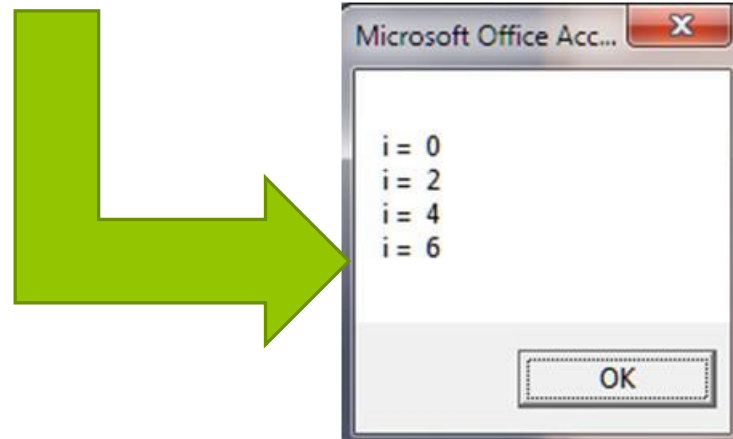
    For Each Item In Buah
        Buahan = Buahan & Item & vbCrLf
    Next
    MsgBox Buahan
End Sub
```

WHILE ... WEND



Pengulangan akan terus dilakukan selama kondisinya masih benar.

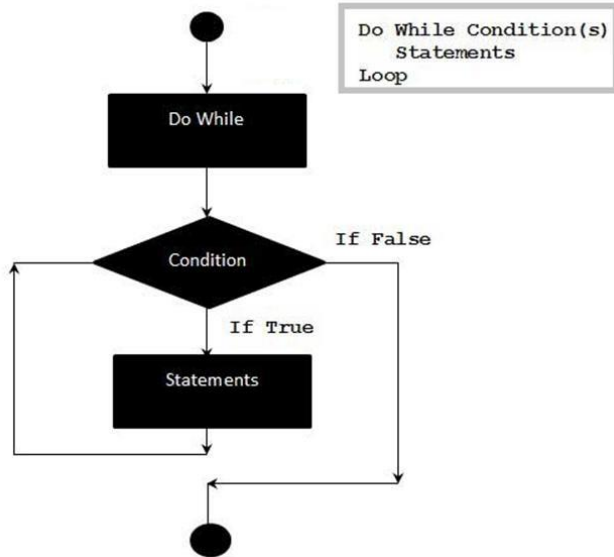
```
'while ...  
Sub cmdWhile()  
    Dim a As Integer  
    Dim outMsg As String  
  
    outMsg = ""  
    a = 7  
    i = 0  
  
    While i < a  
        outMsg = outMsg & "i = " & i & vbCrLf  
        i = i + 2  
    Wend  
    MsgBox outMsg  
  
End Sub
```



WHILE ... WEND

```
'while ...  
Sub cmdWhile()  
    Dim a As Integer  
    Dim outMsg As String  
  
    outMsg = ""  
    a = 7  
    i = 0  
  
    While i < a  
        outMsg = outMsg & "i = " & i & vbCrLf  
        i = i + 2  
    Wend  
    MsgBox outMsg|  
  
End Sub
```


DO WHILE... LOOP



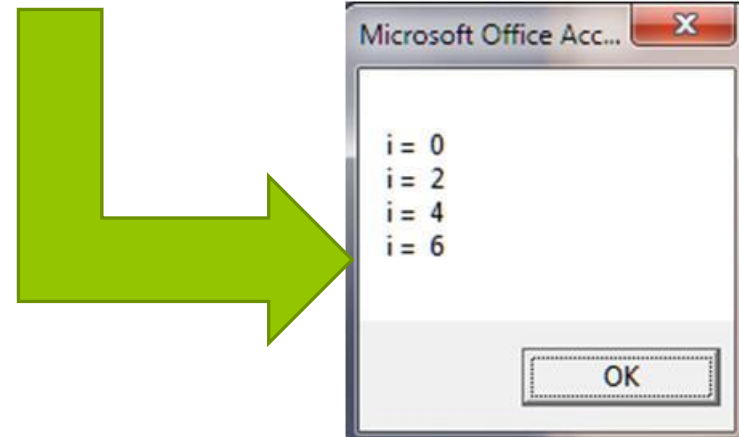
Pengulangan akan terus dilakukan selama kondisinya masih benar.

```
'do while ... loop
Sub cmdDoWhile()
    Dim a As Integer
    Dim outMsg As String

    outMsg = ""
    a = 7
    i = 0
    |
    Do While i < a
        outMsg = outMsg & "i = " & i & vbCrLf
        i = i + 2
    Loop

    MsgBox outMsg

End Sub
```



DO WHILE... LOOP

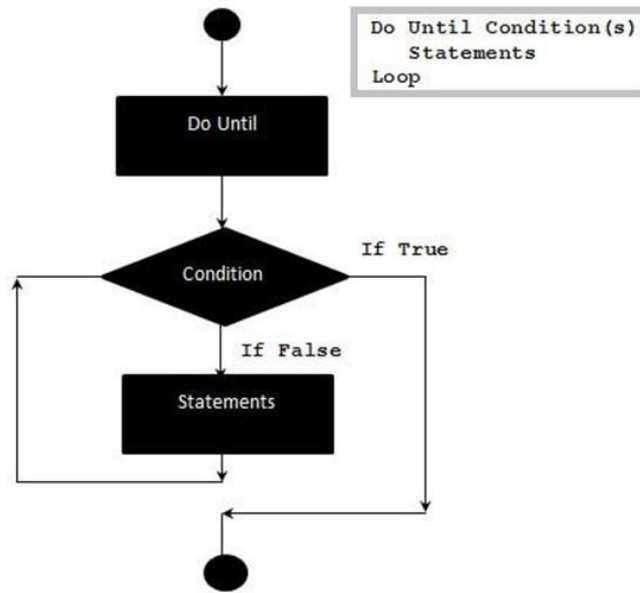
```
'do while ... loop
Sub cmdDoWhile()
    Dim a As Integer
    Dim outMsg As String

    outMsg = ""
    a = 7
    i = 0
    |
    Do While i < a
        outMsg = outMsg & "i = " & i & vbCrLf
        i = i + 2
    Loop

    MsgBox outMsg

End Sub
```

DO UNTIL... LOOP



Pengulangan akan dilakukan selama kondisinya masih salah

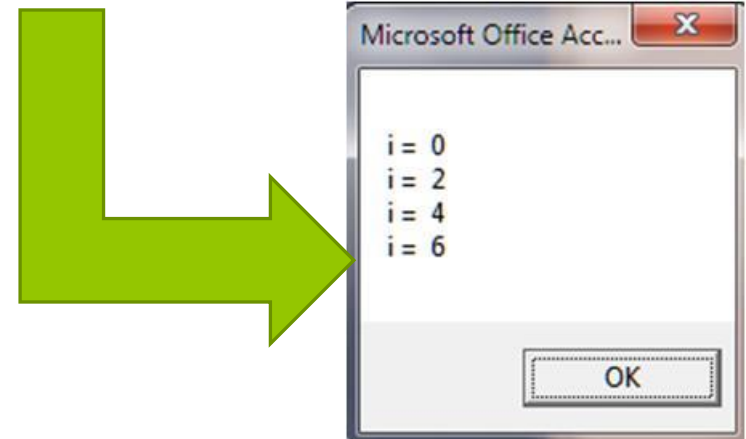
```
'do until ... loop
Sub cmdDoUntil|()
    Dim a As Integer
    Dim outMsg As String

    outMsg = ""
    a = 7
    i = 0

    Do Until i >= a
        outMsg = outMsg & "i = " & i & vbCrLf
        i = i + 2
    Loop

    MsgBox outMsg

End Sub
```



DO UNTIL... LOOP

```
'do until ... loop
Sub cmdDoUntil|()
    Dim a As Integer
    Dim outMsg As String

    outMsg = ""
    a = 7
    i = 0

    Do Until i >= a
        outMsg = outMsg & "i = " & i & vbCrLf
        i = i + 2
    Loop

    MsgBox outMsg

End Sub
```



Aisyah Fitri Yuniasih, S.ST, SE, M. Si
aisyah.fy@stis.ac.id