

BAB 7

MODUL, PROSEDUR DAN FUNGSI

Pada bab ini, akan membahas Konsep Modul pada konsep modul akan melihat penggunaannya pada multi forms, Prosedur, fungsi, Deklarasi variabel, fungsi dan prosedur pada modul, implementasi atau contoh aplikasi menggunakan modul, serta percabangan Subroutine, fungsi Error dan fungsi library pada visual Basic 6.0.

7.1 KONSEP MODUL

MODUL Digunakan utk mendeklarasikan variabel, prosedur, fungsi atau UDF & koneksi secara Public. Dimana nama variabel, prosedur, fungsi atau UDF (Function Defined User atau fungsi yang dibuat oleh user) dan koneksi yang dideklarasikan dalam Modul akan dikenal di beberapa form (Multiple Form) dalam satu project.

Penggunaan MODUL pada MULTIPLE FORM

Penggunaan modul dapat dilakukan pada multiple form dan MDI (Multi Document Interface, MDI ini Akan dibahas pada tersendiri yaitu Bab Menu Editor Dan MDI). Tujuan Pendeklarasian variabel, pembuatan subroutine fungsi & procedure pada modul, agar dikenali oleh seluruh form, Dan juga Untuk mempersingkat penulisan statement /rutine program, Serta memperpendek waktu pelaksanaan program. Jadi penggunaan modul ini sangat effective dan efisien sekali bila aplikasi . proyek yg kita buat terdiri dari banyak form.

Diatas disebutkan tujuan pendeklarasian pada Modul untuk mempersingkat penulisa code/program aplikasi yang dibuat karena Rutine program yg sama cukup dituliskan satu kali saja pada modul, kemudian dapat di gunakan/jalankan/panggil berulang-ulang kali diseluh form dalam satu project bila diperlukan. Dan tidak perlu menuliskan rutine program tersebut diseluruh form cukup satu kali saja pada modul.

Function dan prosedur yang dituliskan dimodul tersebut bila terjadi atau ada error/kesalahan yang harus perbaiki dari fungsi dan prosedur atau bagian rutine program, maka cukup memodifikasinya pada modul saja, dan tidak perlu melacaknya diseluruh form yang memanggilnya.

Modul juga dikembangkan untuk tujuan atau fungsi khusus. Modul dibuat secara general (umum). Pemrograman visual Dengan memiliki modul ini diharapkan pemrograman atau pemakai dapat dengan mudah dan cepat untuk mengembangkan aplikasi-aplikasi yang dibutuhkan. Pada visual basic dalam satu project modul dapat diciptakan lebih dari satu. Modul dalam pemrograman, umum dilakukan dan sangat diperlukan untuk mempermudah debugging dan pengembangan program.

7.2 DEKLARASI PADA MODUL

Pendeklarasian variabel, prosedur, fungsi atau UDF & koneksi pada Modul dapat menggunakan perintah atau keyword Public dan Private. Dimana penggunaan keywordnya, lihat tabel dibawah ini :

| Keyword | Digunakan pada |
|----------------|---|
| Public | Berlaku pada level modul |
| Sub | Berlaku pada level modul dan level form |
| Private | Berlaku pada level form |
| Dim | Berlaku pada level Form General dan procedure |
| Static | Berlaku pada level procedure |

Perbedaan deklarasi Public dgn Dim dan Static

Public

1. Deklarasi Publick harus dituliskan di modul hanya dikenal penulisannya di dalam modul
2. Semua Variabel, fungsi, procedure yang dideklarasikan dimodul harus dengan keyword Publick, kecuali subrutin cukup Sub saja, sama dengan diform. Hanya saja kalau yang dituliskan dimodul akan dikenal oleh seluruh form dalam satu project tersebut.
3. Cara memanggil fungsi, prosedur, namaVariabel dan subrutine tersebut pada form, cukup menuliskan nama fungsi/procedur, nama variabel dan nama subrutine tersebut, bila diperlukan dalam setiap form. jadi tidak perlu dideklarasikan lagi disetiap Form bila akan digunakan. Dan cukup sekali saja dideklarasikan pada modul.
3. Modul ini bertujuan untuk menyingkat penulisan program/code
4. Fungsi/subrutine/procedure/Variabel yang dideklarasikan pada Modul dikenal dengan Fungsi/subrutine/procedure/Variabel Public.

Dim

1. Keyword Dim digunakan untuk mendeklarasi nama variabel pada form dalam prosedur private dan general. Keyword Dim tidak dapat digunakan pada modul
2. Variabel yang dideklarasikan pada general disebut prosedur global karena dapat dikenal di seluruh prosedur private yang ada di satu form saja. Kalau variabel yang dideklarasikan dalam prosedur private hanya dapat dikenal didalam prosedur private itu saja dan tidak dikenal oleh prosedur private yang lainnya, karena dia disebut variabel private/lokal.

Static

1. Keyword Static digunakan untuk mendeklarasi nama variabel pada form dalam prosedur private dan tidak bisa digunakan pada general apalagi dimodul jelas jelas tidak akan bisa/dikenal. Keyword Dim hanya dapat digunakan untuk medeklarasi nama variabel pada prosedur priplate saja.
2. Variabel yang dideklarasikan dengan keyword static dikenal dengan variabel static

Subrutine (SUB)

1. **Keyword Sub digunakan untuk mendeklarasikan subrutine dari fungsi yang kita ciptakan sendiri (UDF) yaitu fungsi yang tidak terdapat dalam library VB. Keyword SUB bisa digunakan di form dan modul. Kalau yang di modul subrutine tersebut bisa dikenal diseluruh form dalam satu project.**
2. **Fungsi yang ada pada pustaka/library VB misalnya fungsi matematik yaitu : Cos, Sin, Roun, Abs, Dan seterusnya. Begitu juga fungsi string yang ada pada Pustaka/library VB yaitu : Mid, left, right, len Dan seterusnya. Dan selain itu kita bisa menciptakan fungsi/subrutine sendiri yang disebut dengan UDF (User Dified Function) untuk lebih jelasnya lihat contoh nya pada sub bab implementasi dibawah ini.**

7.3 PROSEDURE

Procedure digunakan untuk memadatkan tugas-tugas berulang ataupun proses proses yang digunakan bersama, seperti perhitungan yang sering dilakukan , text dan manipulasi kontrol, serta operasi database.

Keuntungan pemakaian procedure dalam program :

- Procedure memungkinkan untuk memecahkan program pada unit logika yang lebih kecil, dengan demikian program menjadi lebih terstruktur sehingga pencarian kesalahan mudah dilakukan. Dan juga dapat dengan mudah melakukan proses debug (membenarkan/memperbaiki kesalahan) daripada jika keseluruhan program tanpa menggunakan procedure.
- Procedure yang digunakan dalam suatu program dapat bertindak sebagai suatu blok yang dibangun untuk program lain dengan sedikit perubahan atau tanpa perubahan sama sekali.

Ada beberapa jenis procedure yang digunakan dalam Visual Basic :

- Sub procedure yang tidak mengembalikan nilai.
- Function procedure yang mengembalikan nilai.
- Event Prosedur adalah termasuk kategori/type dari Sub Prosedur
- General procedure adalah termasuk kategori/type dari Sub Prosedur
- Property procedure yang dapat mengembalikan nilai dan diisi nilai yang mengacu pada suatu objek.

7.3.1 SUB PROSEDURE

- Prosedur yang tidak akan mengembalikan suatu nilai
- Block kode yang di jalankan sebagai tanggapan atas terbentuknya kejadian (event). Dan harus dipanggil dalam event prosedur.
- Dapat ditempatkan pada standard **modul**
- Sub Prosedur terdiri dari dua type yaitu :
 - General Procedure**, procedure yang diaktifkan oleh aplikasi.

-**Event Procedure**, procedure yang diaktifkan oleh system sebagai respon terhadap event.

- General Procedure : bila dideklarasikan di modul dan harus diaktifkan oleh aplikasi.
 Kalau Event Procedure : Bila dideklarasikan pada Form, maka harus diaktifkan oleh system sebagai respon terhadap event.

Sintak Penulisan Sub Prosedur :

```
[Private|Public][Static]Sub namaprocedure(argument-argumen)
    pernyataan-pernyataan
End Sub
```

Atau

```
[PRIVATE] [PUBLIC] [STATIC] SUB NAMA PROSEDURE (PARAMATER)
    .....
END SUB
```

Setiap kali procedure dipanggil, maka pernyataan-pernyataan yang berada di antara Sub dan End Sub akan dijalankan. Argumen pada procedure adalah nilai yang akan dilewatkan saat pemanggilan procedure.

Contoh :

Sub Tengah yang dapat digunakan untuk menampilkan form ke tengah layar, dimana x adalah parameter yang merupakan form yang akan dibuat ke tengah layar.

```
Sub Tengah(x As Form)
    x.Top = (Screen.Height - x.Height) \ 2
    x.Left = (Screen.Width - x.Width) \ 2
End Sub
Private Sub Form_Load()
    Call Tengah(Me)
End Sub
```

7.3.2 FUNCTION PROSEDURE

Pada Visual Basic telah tersedia berbagai fungsi pustaka/library yang disebut fungsi bawaan seperti Sqr, Cos, dan Chr, tetapi fungsi- fungsi yang tersedia tersebut bersifat umum dan kadang-kadang tidak memenuhi kebutuhan programmer, untuk keperluan tersebut kita dapat menciptakan fungsi-fungsi sendiri yang dikenal dengan UDF yang sintaknya menggunakan default yang telah ditentukan oleh VB yaitu penggabungan keyword : **Function Procedure**.

Sintak Penulisan Function Prosedur :

```
[Private|Public][Static]Function namaprocedure
    (argument-argumen) [As type]
    statements
End Function
```

Contoh

Fungsi ciptaan untuk mengembalikan nama bulan dari suatu tanggal dalam bahasa Indonesia.

```
Function Bulan(x As Date)
    Dim sRet As String
```

```

Select Case Month(x)
  Case 1 : sRet = "Januari"
  Case 2 : sRet = "Februari"
  Case 3 : sRet = "Maret"
  Case 4 : sRet = "April"
  Case 5 : sRet = "Mei"
  Case 6 : sRet = "Juni"
  Case 7 : sRet = "Juli"
  Case 8 : sRet = "Agustus"
  Case 9 : sRet = "September"
  Case 10 : sRet = "Oktober"
  Case 11 : sRet = "Nopember"
  Case 12 : sRet = "Desember"
  Case Else
    sRet = "tidak sah"
End Select
Bulan = sRet
End Function

```

7.3 IMPLEMENTASI MODUL PADA MULTIFORM

Pada sub bab ini akan membahas implementasi fungsi UDF yaitu membuat fungsi parameter by Reference dan fungsi parameter by value serta fungsi Operator matematik pada modul. Dan juga membahas pendeklarian variable pada modul yaitu variable koneksi ke database Microsoft Accessw serta variable koneksi ke table (recordset)nya pada modul dengan menggunakan keyword Public, bila di level Form menggunakan keyword Dim.

Jadi bukan hanya deklarasi variable Coneksi database saja yang menggunakan keyword Public di Modul dan Dim pada level Form, tapi juga deklarasi variable lainnya. Untuk menggunakan Modul terlebih dulu tampilkan vie code dari Modul. Sekarang bahas terlebih dulu fungsi parameter by referensi dan fungsi parameter by referensi serta cara memunculkan modul yang akan digunakan.

7.3.1 Implementasi Fungsi Parameter by Value dan Fungsi Parameter By Referensi pada modul

•PARAMETER BY VALUE

- Pemanggilan fungsi parameter by value dilakukan dengan perintah "BYVAL"
- Pemanggilan parameter ini menandakan procedure yang menggunakan fungsi parameter ini hanya akan mengubah nilai variable salinan (variable di luar prosedure) dan tidak mengubah nilai variable asli (variable diluar procedure yang bersangkutan).

•PARAMETER BY REFERENSI

- Pemanggilan parameter by value dilakukan dengan perintah "BYREF"
- Prosedure yang menggunakan parameter ini hanya akan mengubah nilai variable salinan (Variabel diluar procedure)

•Contoh By Reference :

Sub REF(ByRef ANGKA As Integer)

```
ANGKA = ANGKA * 10
Form1.Print " ANGKA DI SUB PROSEDUR REF = " & ANGKA
End Sub
```

Dimana :

Panggilan pertama fungsi parameter **By Reference** akan menghasilkan :
10 X Angka = 50 karena variable Angka = 5 lihat pada procedure private sub CmdRef dipanggil sebanyak 3x, panggilan pertama ini hasilnya distore kan ke variable Angka.

Panggilan Kedua fungsi parameter **By Reference** akan menghasilkan :
10 X Angka = 500 karena Hasil panggilan pertama (yang distore ke variable angka) dikaliakn lagi dengan 10

Panggilan Ketiga fungsi parameter **By Reference** akan menghasilkan :
10 X Angka = 5000 begitulah seterusnya setiap pemanggilan nilainya akan selalu dikalikan dengan 10.

•Contoh By Value :

```
Sub VAL(ByVal ANGKA As Integer)
  ANGKA = ANGKA * 10
  Form1.Print " ANGKA DI SUB PROSEDUR VAL = " & ANGKA
End Sub
```

Dimana :

Panggilan pertama fungsi parameter by value akan menghasilkan :
10 X Angka = 50 karena variable Angka = 5 lihat pada procedure private sub CmdVal dipanggil sebanyak 2x, hasil pemanggilan pertama ini tidak di store kan ke variable angka

Panggilan Kedua fungsi parameter by value akan menghasilkan :
10 X Angka = 50 karena Hasil panggilan pertama (tidak distore ke variable angka) makanay hasil nya setiap pemanggilan akan sama terus.

•Contoh Program pemanggilan By Reference

```
Private Sub Cmdref_Click()
  Dim ANGKA As Integer
  Print
  Form1.Print " HSL FUNGSI REF"
  ANGKA = 5
  REF ANGKA 'Panggilan Pertama
  Form1.Print " ANGKA = " & ANGKA
  REF ANGKA 'Panggilan ke Dua
  Form1.Print " ANGKA = " & ANGKA
  REF ANGKA 'Panggilan ke Tiga
  Form1.Print " ANGKA = " & ANGKA
End Sub
```

•Contoh Program pemanggilan By Reference

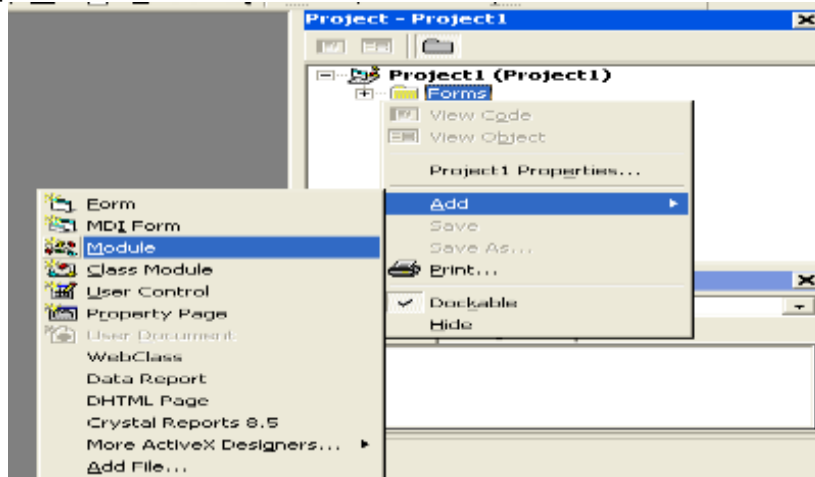
```
Private Sub Cmdval_Click()
```

```
Dim ANGKA As Integer
Print
Form1.Print " HSL FUNGSI VAL"
ANGKA = 5
VAL ANGKA 'Panggilan Pertama
Form1.Print " ANGKA = " & ANGKA
VAL ANGKA 'Panggilan ke Dua
Form1.Print " ANGKA = " & ANGKA
End Sub
```

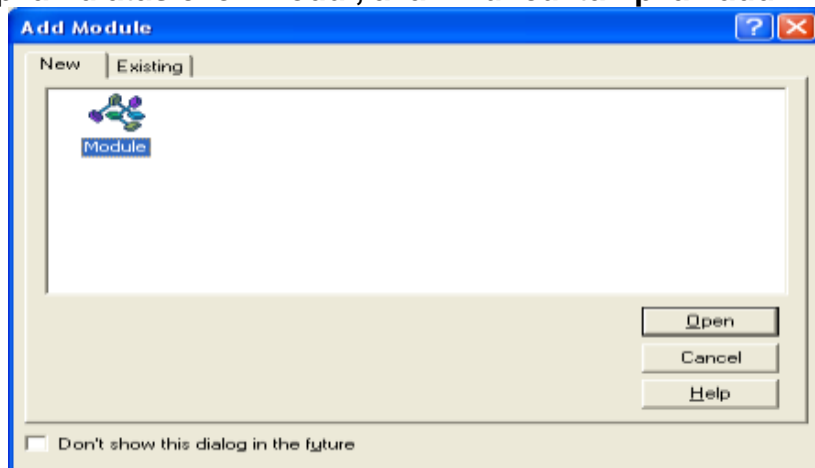
•MENAMPILKAN MODUL DAN DEKLARASI FUNGSI PAMETER BY REFERENSI & FUNGSI PAMETER BY VALUE

Cara memunculkan modul dan mendeklarasikan code pada modul, ikuti langkah-langkah berikut ini :

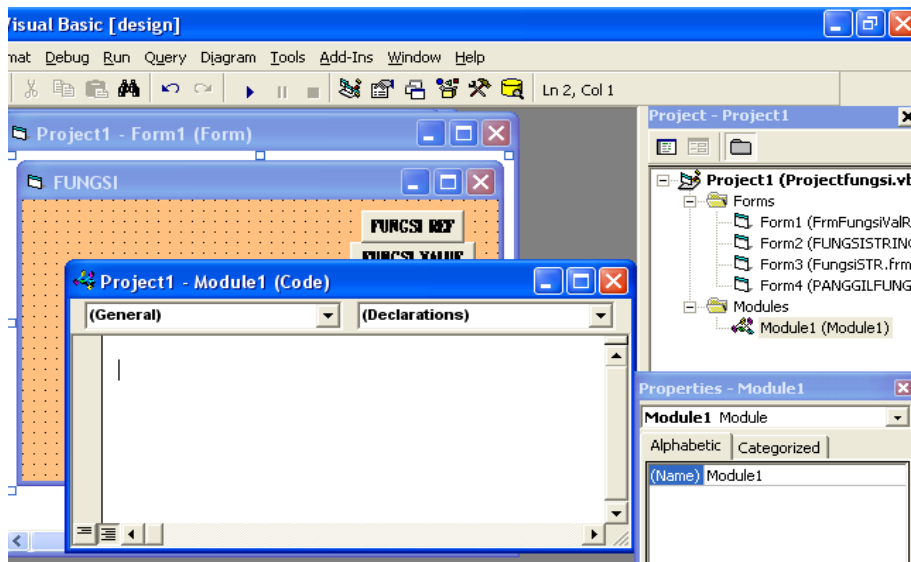
1. Click kanan mouse pada windows/jendela project | Pilih Add | Pilih Modul | lihat tampilan berikut :



2. Pada tampilan diatas click modul, akan muncul tampilan add modul,

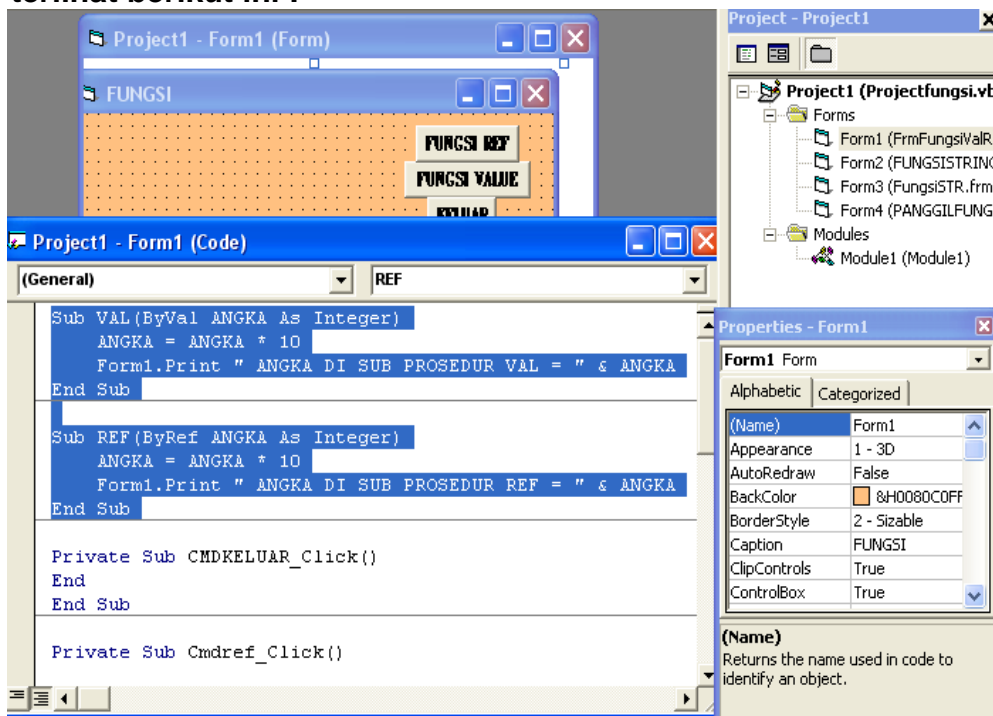


3. Pada tampilan diatas, click open, akan muncul jendela modul berikut :



Diatas anda lihat pada jendela project. Dibawah Project1 terdapat Forms (banyak form) dan Moduls (berarti modul bisa lebih dari satu modul untuk satu project, disini baru ada modul1). Click module1 tersebut dan click icon viewcode pada toolbar project explorer selanjutnya viewCode modul, siap untuk digunakan.

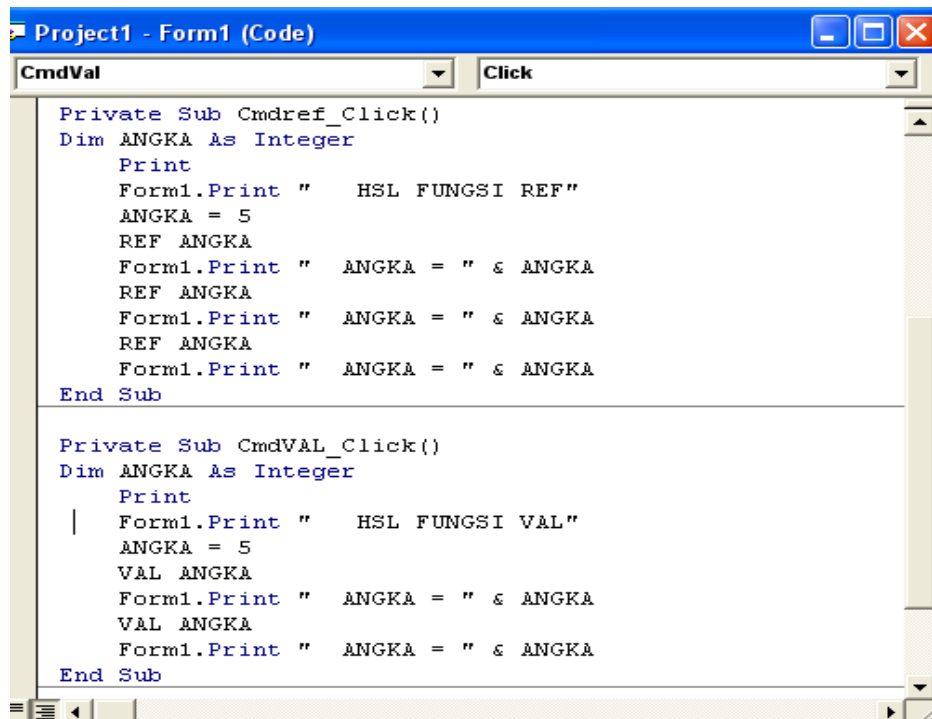
4. Kemudian Blok code/program yang ada di deklarasi general, lalu cut seperti terlihat berikut ini :



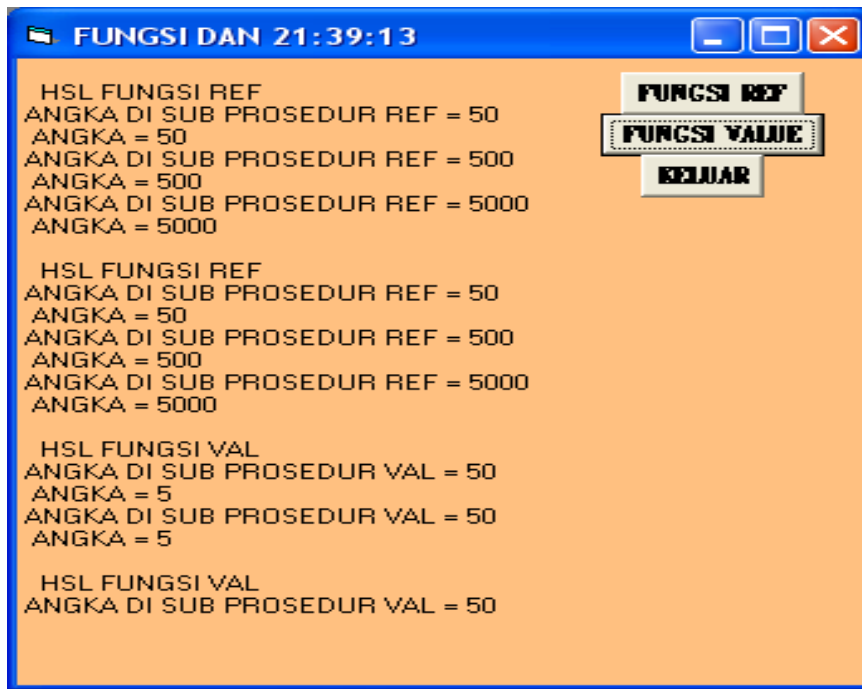
5. Dan Kemudian lakukan paste dalam viewcode modul hasilnya lihat berikut ini :



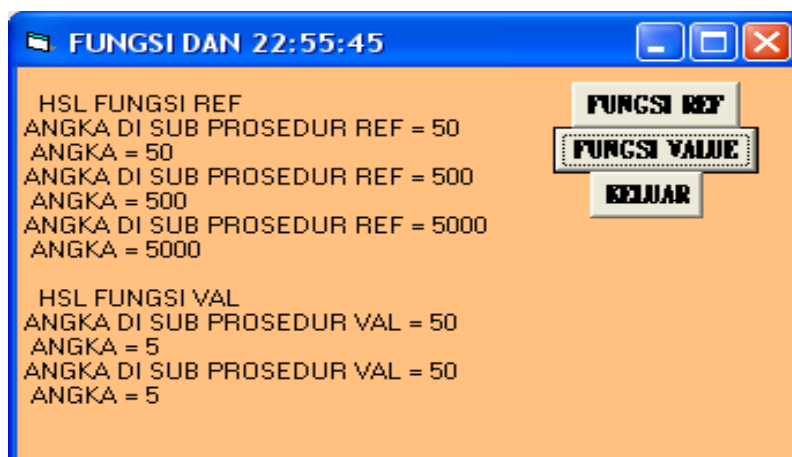
Pada procedure private yang dideklarkan pada level form memanggil nama Ref dan Val yang dibuat di modul, perhatikan code dibawah ini :



6. Coba jalankan form fungsi referenc dan value tersebut, dengan mengclick icon start (▶) pada toolbar. Perhatikan Hasilnya berikut ini :

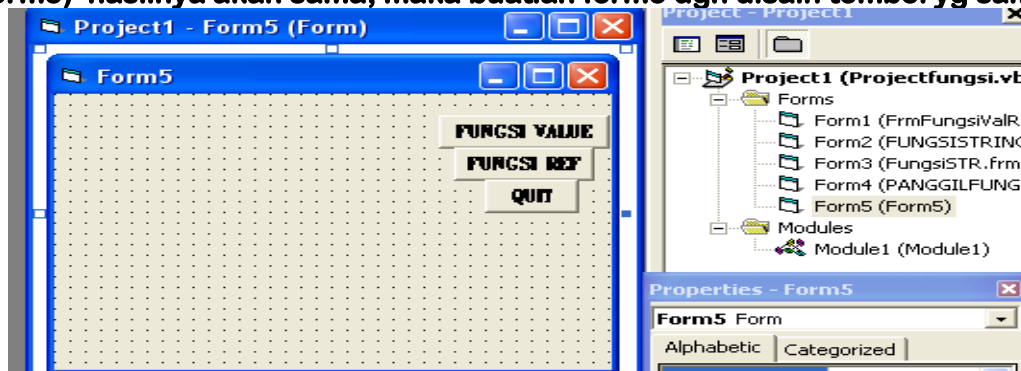


Dgn mengclick fungsi ref 2X & fungsi Value 2X

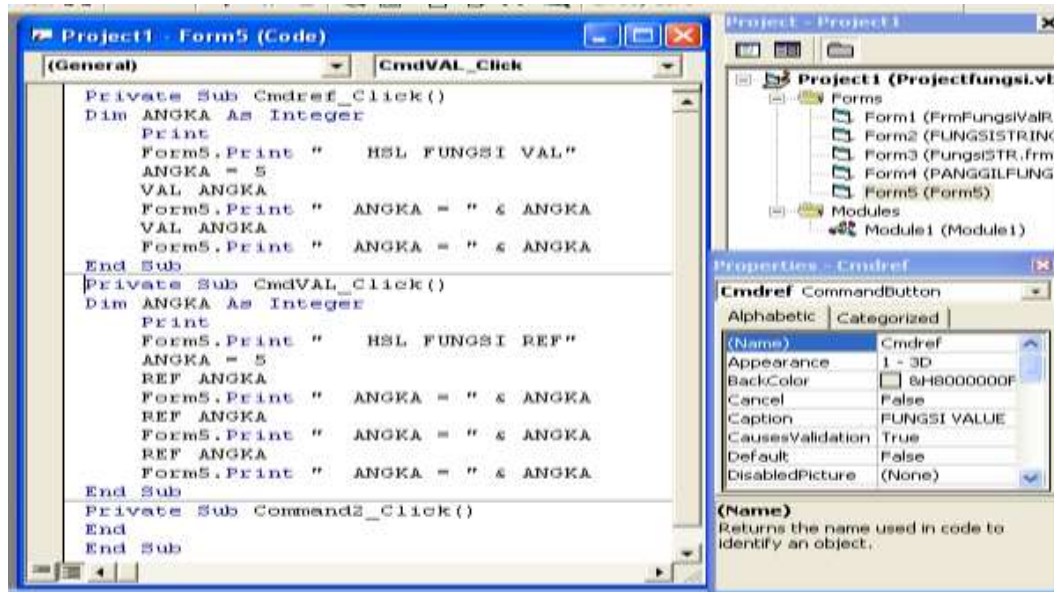


Dgn mengclick tombol fungsi ref 1X, tombol fungsi value 1X

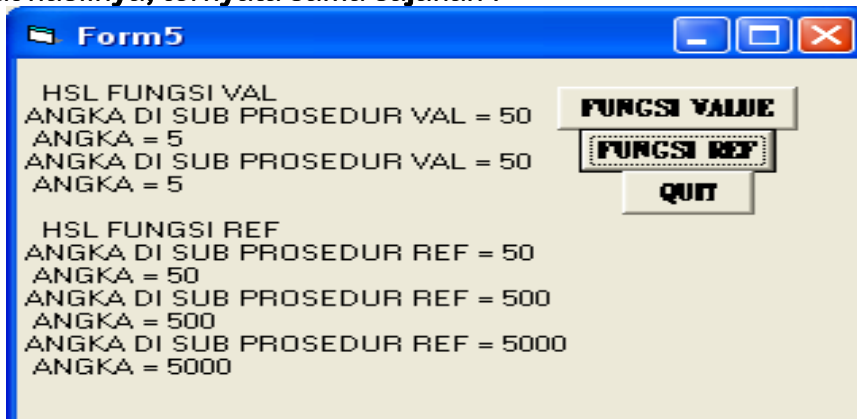
7. Sekarang coba tambah form baru lalu, panggil fungsi Ref dan Val tersebut pada baru (form5) hasilnya akan sama, maka buatlah form5 dgn disain tombol yg sama :



8. Kemudian copy code form1 yang terdapat dalam tombol (command) fungsi value dan tombol fungsi Ref ke tombol (command) fungsi value dan Tombol fungsi Ref yang terdapat pada form5, lihat hasilnya berikut ini :



9. Setelah itu coba jalankan form5 tersebut deng mengclick icon start (▶) pada toolbar lihat hasilnya, ternyata sama sajakan :



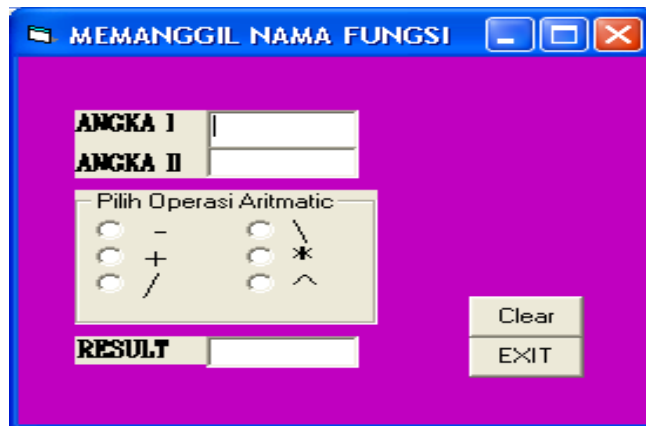
Dgn mengclick tombol fungsi ref 1X dan tombol Fungsi Value 1X

Catatan :

Disini terlihat dua buah form yaitu form1 dan form5. memanggil atau menjalankan fungsi reference dan value yang di deklarasikan dalam module1 hasilnya sama. Tapi bila deklarasi pada general form1, fungsi reference dan fungsi value tersebut tidak bisa dilaksanakan pada form5.

7.3.2 Implementasi Deklarasi Variabel pada modul

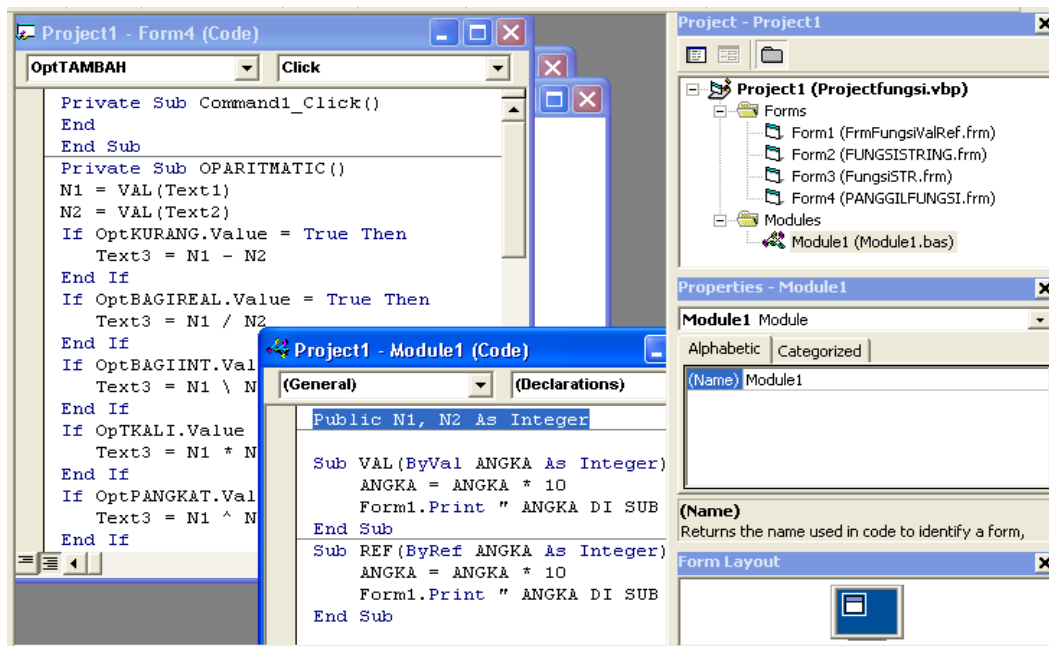
Pada Contoh ini, adalah membuat fungsi UDF yaitu membuat fungsi Operator matematik, kemudian mendeklarasikan variable yang digunakan untuk menampung operan atau data nya pada modul. Sekarang Bukalah project1 / project baru, Pada Jendela Form1 buatlah UI (User Interface). Gunakan Object Label, TextBox, Frame, OptionButton dan CommandButton Kemudian settinglah propertinya sesuai yang terlihat pada rancangan Form dibawah ini :



Buka Jendela Code dan pada bagian Code Editor ketikkan kode programnya sebagai berikut :

| | |
|---|--|
| <pre> Dim N1,N2 As Integer Private Sub OPARITMATIC() N1 = VAL(Text1) N2 = VAL(Text2) If OptKURANG.Value = True Then Text3 = N1 - N2 End If If OptBAGIREAL.Value = True Then Text3 = N1 / N2 End If If OptBAGIINT.Value = True Then Text3 = N1 \ N2 End If If OptTKALI.Value = True Then Text3 = N1 * N2 End If If OptPANGKAT.Value = True Then Text3 = N1 ^ N2 End If If OptTAMBAH.Value = True Then Text3 = N1 + N2 End If End Sub </pre> | <pre> Private Sub OptBAGIINT_Click() OPARITMATIC End Sub Private Sub OptBAGIREAL_Click() OPARITMATIC End Sub Private Sub OptTKALI_Click() OPARITMATIC End Sub Private Sub OptKURANG_Click() OPARITMATIC End Sub Private Sub OptPANGKAT_Click() OPARITMATIC End Sub Private Sub OptTAMBAH_Click() OPARITMATIC End Sub Private Sub Command1_Click() End End Sub </pre> |
|---|--|

Kemudian Deklarasi Variabel N1, N2 diatas dipindahkan ke Modul, ganti keyword Dim dengan Public, lihat berikut ini :



Keterangan untuk dilevel modul :

1. Disini hanya yang dapat dipindahkan ke modul hanya deklarasi variabel N1, N2 saja. Setelah dipindahkan ke modul ganti keyword dim dengan Public, kemudian hapuslah deklarasi variabel N1, N2 yang terdapat dilevel form tersebut atau bisa juga non aktifkan segaai berikut :

```
‘Dim N1,N2 As Integer
```

2. Untuk Fungsi OPARITMATIC yang dipanggil sebenarnya bisa diletakan atau dideklarasikan juga di modul, asalkan Tidak mengandung Object.
3. Dalam Fungsi OPARITMATIC yang dipanggil tersebut diatas, terdapat Object Text3 yaitu : `Text3 = N1/N2`, dan seterusnya.
4. Modul tidak mengenal nama Object kecuali Form.
Bila Fungsi OPARITMATIC tersebut, object Text3 diganti dengan nama Variabel yaitu : `N3 = N1/N2`, maka pada prosedur private pemanggil misalnya saja :
`Private Sub OptBAGIINT_Click()` dituliskan dilevel form
`OPARITMATIC`
`Text3=N3`
`End Sub`

Kemudian untuk Fungsi OPARITMATIC(), yaitu : **Private Sub OPARITMATIC()** dituliskan di modul dengan mengganti Keyword Private menjadi Public, maka penulisannya di modul menjadi **Public Sub OPARITMATIC()** atau cukup ditulis dengan keyword **Sub OPARITMATIC()**

5. Tantangan : Buatlah deklarasi Fungsi OPARITMATIC(), pada Level modul dan prosedur private pemanggil tuliskan di level Form, bernalar lah gimana seharusnya code/program tersebut. bila diganti instruksi `Text3 = N1/N2` dan seterusnya !.

Simpan Project1 dengan nama : **ProjectFungsi.vpb** dan **Form1** dengan nama : **LatForm01.frm**. Kemudian Coba jalankan dengan click icon start (▶) pada toolbar !. Lihat hasilnya berikut ini :



Gbr.1. Memanggil fungsi kali

Gbr.2. Memanggil fungsi kurang

Kemudian click tombol Clear, isikan lagi data operan pada **angka I** dan Pada **Angka II** dan panggilah fungsi operator aritmatik berikutnya :



Gbr.3. Memanggil fungsi kurang

Gbr.4 Memanggil fungsi tambah



Gbr.5. Memanggil fungsi bagiReal

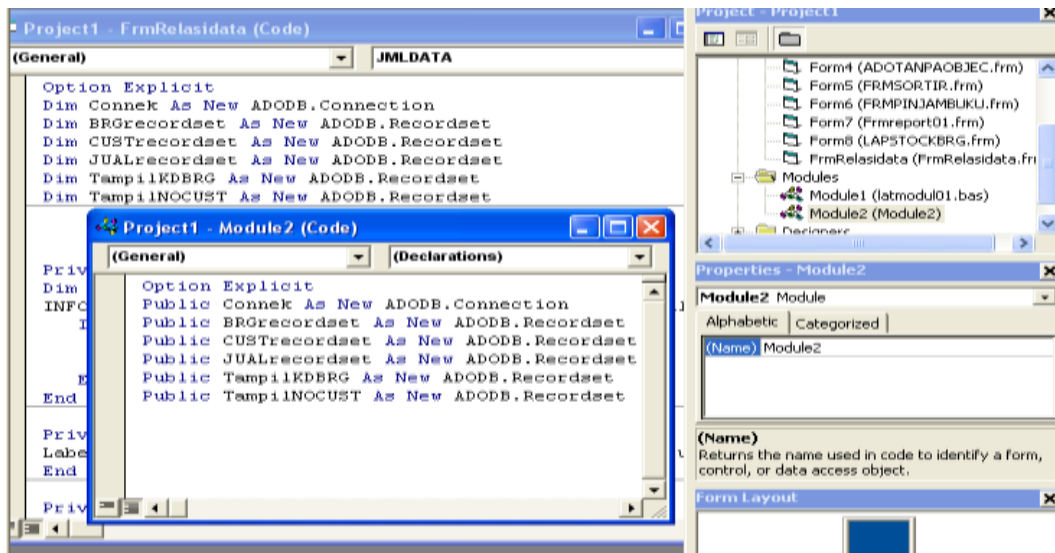
Gbr.6 Memanggil fungsi bagiInt

Caranya isikan nilai Operan / data pada object text **Angka I** dan pada object text **Angka II** | kemudian click **Result** hasilnya akan muncul pada **object text**

Result. Kemudian perhatikan hasilnya setiap operasi !

6.3.3 Implementasi Deklarasi Variabel koneksi Database pada Modul

Pada Contoh ini, adalah mendeklarasikan variable koneksi ke Database dan variable koneksi ke Recorset atau table pada Modul seperti berikut ini :



Perhatikan penulisan deklarasi variable diatas dari **General Level Form** sebelum dipindahkan ke modul :

Pada level form :

- ▶ Deklarasi variable digeneral menggunakan keyword **Dim**.
- ▶ Option Explicit : yang diletakan paling atas deklarasi fungsinya :
Setiap variable digunakan pada aplikasi nanti, harus sudah dideklarasikan bila tidak akan terdeteksi pada saat aplikasi di **Compile/run**, muncul kesalahan atau Error **“Variabel not found“**.

Pada level Modul :

- ▶ Pindahkan deklarasi variable yang ada digeneral dilevel form diatas ke modul dengan cara meng **Cut** seluruh deklarasi variable yang ada digeneral tersebut lalu **paste** pada viewcode modul.
- ▶ Termasuk Option Explicit juga ikut dipindahkan ke modul, fungsinya sama seperti di general level form.
- ▶ Kemudin ganti keyword **Dim** dengan **Public** seperti yang di lihat pada tampilan diatas.
- ▶ Setelah itu hapuslah semua deklarasi variabel koneksi di general level form tersebut yang telah dindahkan ke Modul.

Catatan :

- ◆ Pada sub bab ini kita hanya melihat deklasi variabel pada mudul multiform
- ◆ Untuk aplikasi data base tersebut secara lengkap yang bisa dilaksanakan silahkan baca juga buku : **“Pemrograman Microsoft visual Basic 6.0 dengan Database Sql Server”**.

- ◆ Pada buku : “Pemrograman Microsoft visual Basic 6.0 dengan Database Sql Server”. juga dibahas koneksi Visual basic ke database Sql Server serta Pemrograman Basisdata atau Struktur Query Languagey (SQL) dan banyak lagi bahasan yang lainnya tentang basisdata Sql server.

7.4 FUNGSI

FUNGSI adalah : Sebuah rutine program yang mampu menerima berbagai nilai atau data dan memberikan hasil tertentu. Selain itu dalam visual basic banyak sekali fungsi-fungsi yang sudah disediakan dalam library/pustaka file visual Basic, yang memiliki tujuan tertentu, tinggal dipanggil nama fungsi tsb. bila kita perlukan otomatis akan memberikan hasil, diantaranya : fungsi tanggal, string, aritmatic, fungsi error dan lain lain. Serta UDF, ByVal, ByReference, yang telah dibahas diatas.

Pada sub bab fungsi ini, hanya akan melihat fungsi Pustaka atau library dan fungsi error.

7.4.1 FUNGSI TANGGAL DAN JAM

TANGGAL dan JAM

Merupakan fungsi yang digunakan untuk manipulasi nilai bertipe date maupun Time. Fungsi Time dan Date dapat dilihat pada table dibawah ini berserta keterangannya :

| FUNGSI | KETERANGAN |
|------------|---|
| Now | Menampilkan tanggal dan jam saat ini |
| Date | Menampilkan tanggal saat ini |
| DateSerial | Mengubah nila numeric menjadi btk tanggal |
| DateDiff | Menghitung interval antara 2 tanggal |
| Time | Menampilkan Waktu saat ini |
| Timer | Menampilkan jumlah detik |
| Timeserial | Mengubah nilai numeric menjadi btk jam |

FUNGSI STRING

Merupakan fungsi yang digunakan untuk manipulasi nilai bertipe string, yaitu berisi Teks, angka atau symbol khusus.

Pada table dibwh ini dapat dilihat fungsi String berserta keterangannya :

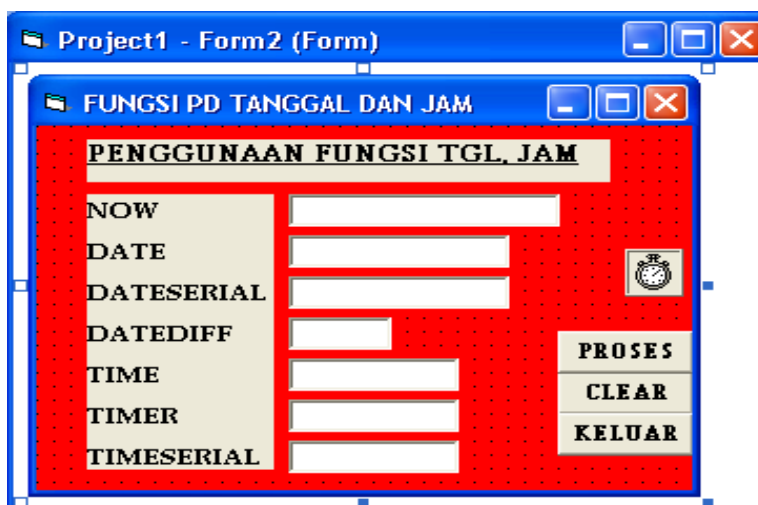
| FUNGSI | PERINTAH | KETERANGAN | HASIL |
|----------------|------------------|--|-------|
| Asc(Str) | Asc("A") | Mengubah nilai string menjadi ascii | 65 |
| CHR(int) | Char(65) | Mengubah nilai ascii menjadi string | A |
| LCase(Str) | LCase("FINUX") | Mengubah string menjadi huruf kecil | Finux |
| UCase(str) | UCase("finux") | Mengubah string menjadi huruf besar | FINUX |
| Left(Str,int) | Left("Finux",3) | Mengambil huruf sebanyak int dari kiri | Fin |
| Right(str,int) | Right("Finux",2) | Mengambil huruf sebanyak int dari kanan | Ux |
| Mid(str,n1,n2) | Mid("Finux",3,2) | Mengambil huruf dari kiri dimulai dari karaktern1, sebanyak n2 | |
| StrReverse | StrReverse(str) | Membalik huruf besar | xuniF |

| | | | |
|------------|-----------------|--|--------|
| LTrim(str) | LTrim(" Finux") | Memotong spasi dr kiri karakter | Finux |
| RTrim(str) | RTrim("Finux ") | Memotong spasi dari kanan karakter | Finux |
| Len(str) | Len(" Finux") | Menghitung jumlah huruf termasuk spasi | 7 |
| STR(int) | Str(1500) | Merubah nilai numeric menjadi string | "1500" |

7.4.2 IMPLEMENTASI FUNGSI DATE & TIME SERTA STRING

Fungsi Tanggal dan Jam :

Bukalah project1 / **ProjectFungsi.vpb**, Tambahkan form baru (Form2). Pada Jendela Form2 buatlah UI (User Interface). Gunakan Object Label, TextBox, dan CommandButton serta object timer. Kemudian settinglah propertinya sesuai yang terlihat pada rancangan Form dibawah ini :



Buka Jendela Code dan pada bagian Code Editor ketikkan kode programnya sebagai berikut : Code untuk Run/Melaksanakan rancangan form diatas :

```

Private Sub CMDPROSES_Click()
    Text1.Text = Now
    Text2.Text = Date
    Text3.Text = DateSerial(1998, 1, 16)
    Text4.Text = DateDiff("D", 16, 23)
    Text5.Text = Time
    Text6.Text = Timer
    Text7.Text = TimeSerial(18, 45, 38)
End Sub

Private Sub CMDCLEAR_Click()
    Text1.Text = ""
    Text2.Text = ""
    Text3.Text = ""
    Text4.Text = ""
    Text5.Text = ""
    Text6.Text = ""
    Text7.Text = ""
End Sub

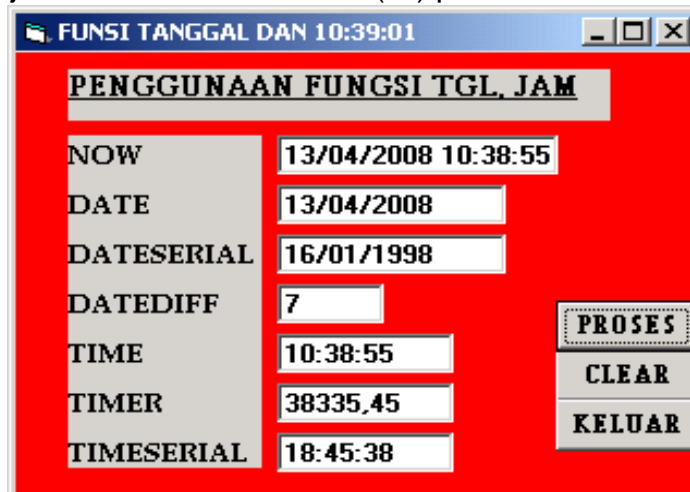
Private Sub CMDKELUAR_Click()
End
    
```

```

End Sub
Private Sub Command1_Click()
    Text1.SETFOCUS
End Sub

Private Sub Timer1_Timer()
    Form2.Caption = "FUNSI TANGGAL DAN " & Time
End Sub
    
```

Simpan **Form2** dengan **nama** : LatForm02.frm. Kemudian Click MenuProject | Project1 Properties | lalu klik tab General. Gantilah Startup Object-nya menjadi Form2. Coba jalankan Click icon start (▶) pada toolbar !. Lihat hasilnya berikut ini :

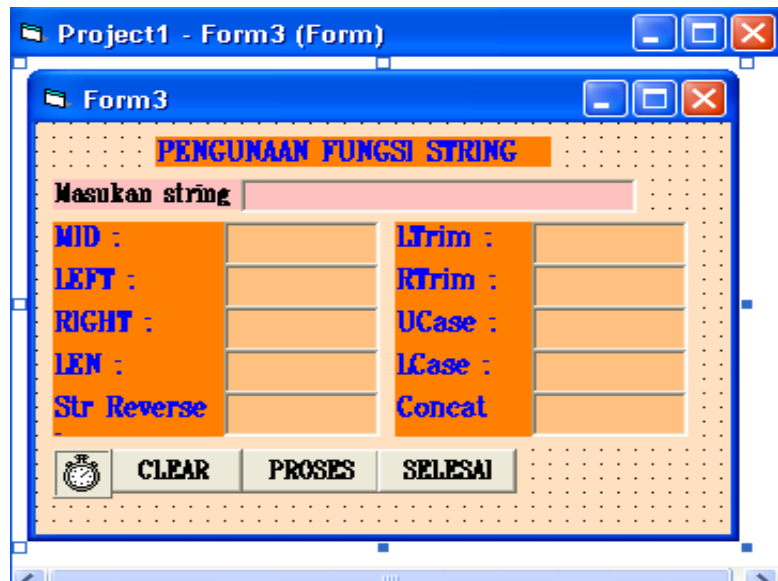


Kemudian click tombol proses dan perhatikan hasilnya setiap operasi lalu baca code nya !.

Data yang digunakan untuk proses disini adalah yang aktif pada object Timer, jadi tidak perlu memasukan data pada kasus ini.

Fungsi String

Bukalah project1 / **ProjectFungsi.vpb**, Tambahkan form baru (Form3). Pada Jendela Form3 buatlah UI (User Interface). Gunakan Object Label, TextBox, dan CommandButton serta object timer. Kemudian settinglah propertynya sesuai yang terlihat pada rancangan Form dibawah ini :



Buka Jendela Code dan pada bagian Code Editor ketikkan kode programnya sebagai berikut : Code untuk Run/Melaksanakan rancangan form diatas :

```

Private Sub CMDPROS_Click()
    Text1.Text = Mid(Txtstr, 2, 3)
    Text2.Text = Right(Txtstr, 2)
    Text3.Text = Left(Txtstr, 2)
    Text4.Text = Len(Txtstr)
    Text5.Text = StrReverse(Txtstr)
    Text6.Text = UCase(Txtstr)
    Text7.Text = LCase(Txtstr)
    Text8.Text = LTrim(Txtstr)
    Text9.Text = RTrim(Txtstr)
    Text10.Text = Right(Txtstr, 2) + Left(Txtstr, 2)
    Txtstr.SetFocus
End Sub

Private Sub Timer1_Timer()
    Form3.Caption = "FUNGSI STRING DAN " & Time
End Sub

Private Sub CMDKELUAR_Click()
    End
End Sub

Private Sub Command1_Click()
    Text1.SETFOCUS
End Sub

```

Simpan **Form3** dengan **nama** : LatForm03.frm. Kemudian Click MenuProject | Project1 Properties | lalu klik tab General. Gantilah Startup Object-nya menjadi Form3.

Coba jalankan Click icon start (▶) pada toolbar !. Lihat hasilnya berikut ini :



Kemudian isikanlah String JAKARTA pada text Masukan string. Dan click tombol proses dan perhatikan hasilnya setiap operasi lalu baca code nya ! Bila ingin coba denga string lain clicklah tombol clear, ualangi lagi masukan data string baru.

7.5 LATHAN SOAL KASUS

1. Tantangan

Buatlah Project baru, dan ciptakan fungsi UDF, membuat fungsi, string, perhitungan atau fungsi yang lainnya sendiri, cobalah anda membuatnya dalam multiform (lebih dari satu form) serta lakukanlah peng code annya menggunakan modu !

2. Diberikan Setting propertie Object Form1 Sbb:

| OBJECT | PROPERTIE | | |
|----------------|------------|-------------|------------|
| | CAPTION | TEXT | NAME |
| FRAME1 | INPUT DATA | | |
| LABEL1 | R1 : | | |
| LABEL2 | R2 : | | |
| LABEL3 | R3 : | | |
| TEXTBOX1 | | DIKOSONGKAN | TXTR1 |
| TEXTBOX2 | | DIKOSANGKAN | TXTR2 |
| TEXTBOX3 | | DIKOSONGKAN | TXTR3 |
| FRAM2 | HASIL | | |
| COMMANDBUTTON1 | SERI | | |
| COMMANDBUTTON2 | PARALEL | | |
| TEXTBOX4 | | DIKOSANGKAN | TXTSERI |
| TEXTBOX5 | | DIKOSONGKAN | TXTPARALEL |
| COMMANDBUTTON3 | DATA BARU | | |
| COMMANDBUTTON4 | SELESAI | | |

Pertanyaan :

1. Buatlah disain form1 dari setting propertie diatas !
2. Buatlah kode programnya pada :

a. **Tombol Seri (CommandButon1) : berisikan code prog :**

utk menghitung nilai tahanan seri = $R1 + R2 + R3$

Apabila di click tombol seri muncul :

- nilai seri pada object textseri dan nilai 0 pada object txtparalel.
- judul pada form berubah menjadi :MENGHITUNG NILAI TAHANAN SERI.
- Backcolor form berubah menjadi Green

b. **Tombol Paralel (CommandButon2) : berisikan code prog. :**

Utk menghitung nilai tahanan paralel = $1/R1 + 1/R2 + 1/R3$

Apabila di click tombol paralel muncul

- nilai paralel pada object txtparalel, dan nilai 0 pada object txtseri
- Judul form berubah menjadi :MENGHITUNG NILAI TAHANAN PARALEL
- Backcolor form berubah menjadi : Yellow.

c. **Tombol data baru (CommandButon3) : berisikan code prog. :**

- Utk mengosongkan object txtR1, txtR2, txtR3, txtseri, dan txtparalel.

Kursor aktif pada object txtR1.

- Judul form berubah menjadi : "INPUT DATA TAHANAN BARU".

- Backcolor form berubah menjadi : Blue.

d. **Tombol Selesai (CommandButon4) : berisikan code prog. :**

utk mengakhir execution program secara permanen.

3. Buatlah fungsi matematik yang terdiri tiga buah form dengan disain masing-masing form adalah seperti terlihat dibawah ini : (gunakan object Label, text, dan commandbutton, kemudian settinglah propertie nya sesuai dengan rancangan masing-masing form tersebut)

Kumudian buatlah codenya pada tombol :

PROSES :

Carilah fungsi matematik sebanyak-banyaknya pada buku pelajaran matematika anda, gunakan untuk menuliskan code nya, gunakanlah deklarasi variable untuk menampung bilangan1 dan bilangan2 serta untuk menampung hasil perhitungan, Dan coba anda buat code yang bisa ditempatkan pada level modul.

LANJUT :

CODE yang digunakan untuk menampilkan form berikutnya !.

CLEAR :

Code yang digunakan untuk mengosongkan semua isi object TextBox

DATA BARU :

Code untuk menampilkan form pertama yaitu tempat memasukan data bilangan1 dan data bialangan2.

Latihan Deklarasari level Modul

PENGUNAAN FUNGSI MATEMATIK

Masukan Bil1 SINGLE :

Masukan Bil2 INTEGER :

ABS(BIL1) :

ROUND(BIL1,2) :

SQR(BIL2) :

INT(BIL1) :

BIL2 MOD 3 :

CLEAR PROSES LANJUT !

Gbr.1 form Fungsi matematika

DEKLARASI LEVEL MODUL

**PENGUNAAN FUNGSI SUDUT
LANJUTAN DARI FUNGSI MATEMATIK**

LOG(BIL2) :

COS(BIL2) :

SIN(BIL2) :

TGN(BIL2) :

INT(BIL2) :

CLEAR PROSES LANJUT !

Gbr.2 form Fungsi Sudut Trigonometri

DEKLARASI LEVEL MODUL

**PENGUNAAN FUNGSI COMPUTASI
LANJUTAN DARI FUNGSI MATEMATIK**

HEX(BIL1) :

OCT(BIL1) :

ASCII(BIL2) :

CHAR(BIL2) :

INT(BIL2) :

CLEAR PROSES DATA BARU

Gbr.1 Form Fungsi Computasi