

MODUL KOMPUTER APLIKASI SIPIL 1

(VISUAL BASIC)

Daftar Isi

[1. MENGENAL VISUAL BASIC 2](#_Toc257306991)

[2. DASAR-DASAR VISUAL BASIC 6.0 6](#_Toc257306992)

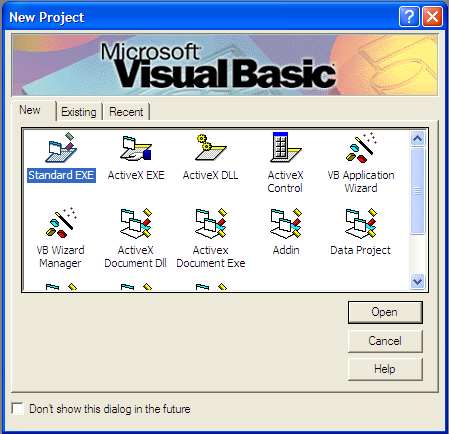
[3. KONTROL PADA VISUAL BASIC 10](#_Toc257306993)

[4. STRUKTUR KEPUTUSAN 13](#_Toc257306994)

[5. STRUKTUR PENGULANGAN 19](#_Toc257306996)

# MENGENAL VISUAL BASIC

**Start >> All Programs >> Microsoft Visual Basic 6**, akan muncul tampilan sebagaimana berikut, lalu pilihlah **Standard.Exe >>Open**

****

Setelah itu akan muncul tampilan berikut ini, lalu kita akan mulai mengenal bagian-bagian dari IDE (*Integrated Development Enviroment*)

Keterangan:

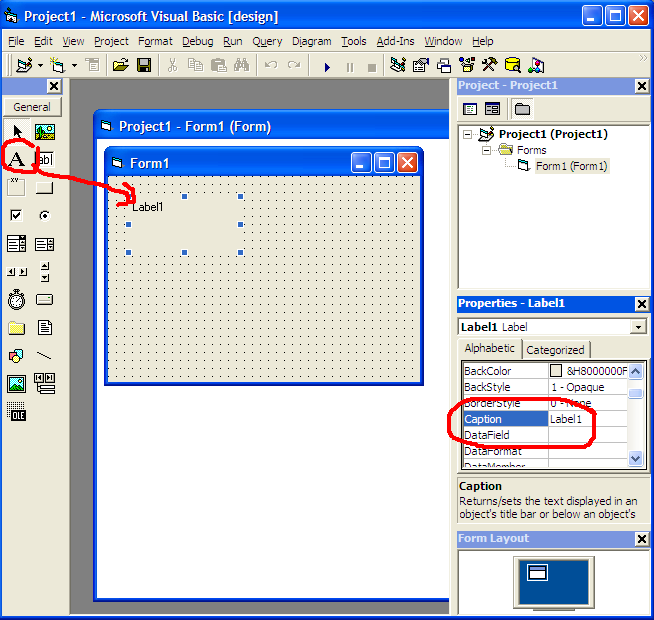
1. *Baris Menu*, digunakan untuk memilih tugas-tugas tertentu seperti menyimpan project, membuka project, dll.
2. *Main Toolbar*, digunakan untuk melakukan tugas-tugas tertentu dengan cepat.
3. *Jendela Proyek*, jendela ini berisi gambaran dari semua modul yang terdapat dalam aplikasi anda. Anda dapat menggunakan icon Toggle Folders untuk menampilkan modul-modul dalam jendela tersebut secara di group atau berurut berdasarkan nama. Anda dapat menggunakan **Ctrl+R** untuk menampilkan jendela project ataupun menggunakan project explorer.
4. *Jendela Form Designer***,** jendela ini merupakan tempat anda untuk merancang user interface dari aplikasi anda. Jadi jendela ini merupakan kanvas bagi seorang pelukis.
5. *Jendela Toolbar*, jendela ini berisi komponen-komponen yang dapat anda gunakan untuk mengembangkan user interface.
6. *Jendela Code*, merupakan tempat bagi anda untuk menulis koding. Anda dapat menampilkan jendela ini dengan menggunakan kombinasi tombol **Shift+F7**.
7. *Jendela Properties*, merupakan daftar property-properti object yang sedang terpilih, contoh anda dapat mengubah warna tulisan (foreground) dan warna belakang (background). Anda dapat menggunakan F4 untuk menampilkan jendela property.
8. *Jendela Form Layout*, akan menunjukkan bagaimana form bersangkutan ditampilkan ketika dijalankan.

Jika jendela-jendela tersebut tidak ada, anda dapat memunculkannya dengan **Menu >> View** dan pilih:

1. Project Explorer (Ctrl+R)
2. Properties Windows (F4)
3. Property Pages (Shift+F4), dsb.

**Membuat Program Pertama**

Agar lebih mudah, sekarang kita akan membuat program pertama:

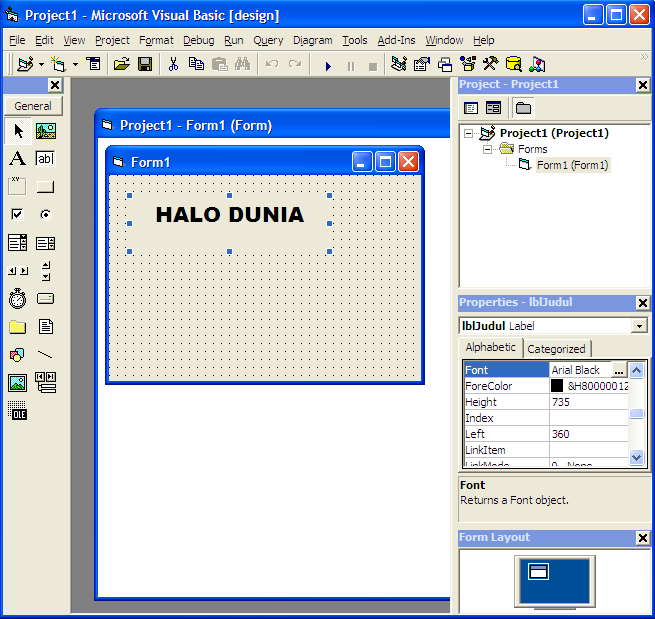


Pilihlah Label pada toolbox dengan simbol , kemudian drag ke jendela form design, kemudian pada jendela properties ubahlah property Caption untuk mengubah text yang tertulis pada layar.

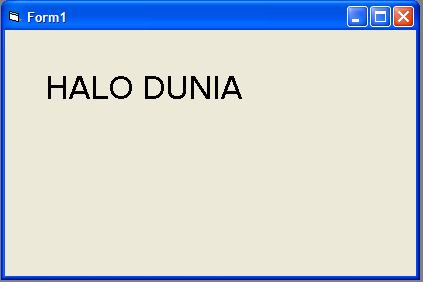
Berikut ini adalah property yang yang kita setting:

|  |  |  |
| --- | --- | --- |
| **Nama Property** | **Value** | **Keterangan** |
| Name | lblJudul | Nama dari Object |
| Alignment | 2-Center | Untuk mengatur agar teks rata tengah |
| Caption | HALO DUNIA | Untuk mengatur teks yang terlihat pada layar |
| Font | Pilih jenis tulisan | Untuk mengatur jenis tulisan, ukuran dan sebagainya |

Sehingga akan menjadi tampilan sbb:

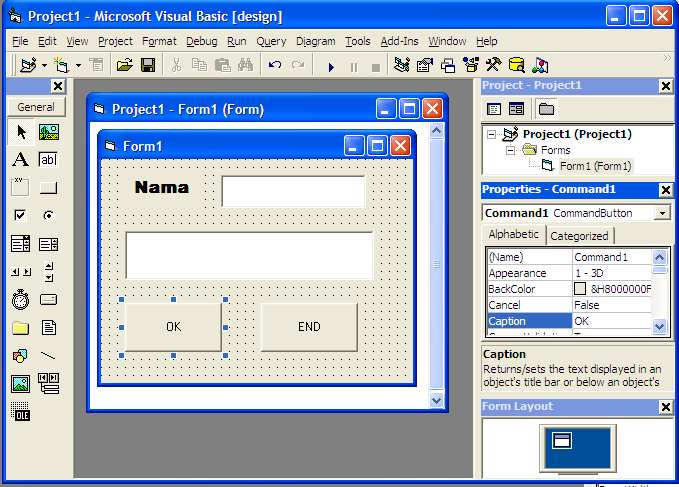


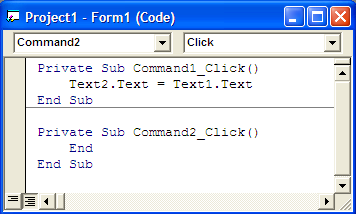
Hasil tampilannya setelah anda jalankan:

****

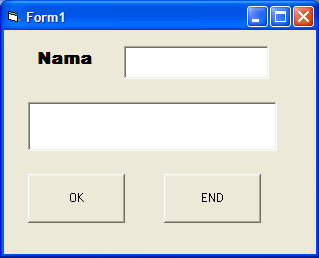
**OK, Jika anda sudah paham maka kita bisa lanjut ke Program-2 berikut ini.**

Buatlah design form yang mengandung Label, Text box dan Command Button seperti dibawah ini:

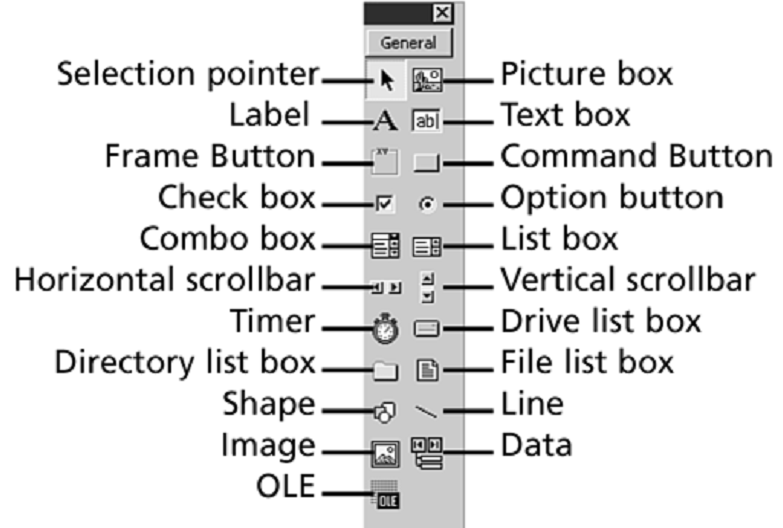
****



Hasil programnya setelah di jalankan:



**TOOLBOX:**



Keterangan:

| **Nama Tombol** | **Fungsi** |
| --- | --- |
| Picture Box | Kontrol untuk menampilkan objek gambar, seperti suatu container yang dapat menerima output/kontrol lain |
| Label | Kontrol untuk menampilkan teks |
| Text Box | Kontrol untuk memasukkan atau mengubah data input teks |
| Frame | Kontrol untuk mengelompokkan kontrol didalam suatu frame |
| Check box | Kontrol untuk menunjukkan suatu kondisi True atau False, atau untuk menampilkan suatu pilihan ganda saat pemakai memilih kotak cek tersebut lebih dari satu |
| Option Button | Kontrol untuk menampilkan beberapa pilihan yang hanya dapat dipilih salah satu oleh pemakai |
| Combo Box | Kontrol kombinasi dari list box dan text box, dimana pemakai dapat memilih item dari daftar atau memasukkan nilai pada bagian kotak teks (text box) |
| List Box | Kontrol untuk menampilkan daftar item yang dapat dipilih salah satu |
| Horisontal  Scroll Bar | Kontrol untuk menunjukkan posisi yang berlaku pada suatu skala, atau sebagai media input/ indicator kecepatan |
| Vertikal Scroll Bar | Kontrol menunjukkan posisi yang berlaku pada suatu skala atau sebagai media input/ indicator kecepatan |
| Timer | Kontrol untuk membangun penghitung waktu berdasarkan nilai interval tertentu. Kontrol ini tidak tampak saat aplikasi dijalankan |
| Drive List Box | Kontrol untuk menampilkan daftar disk drive |
| Directory List Box | Kontrol untuk menampilkan direktori dan path |
| FileListBox | Kontrol untuk menampilkan daftar nama file |
| Shape | Kontrol untuk menggambarkan variasi bentuk pola pada form. Anda dapat memilih pola kotak (rectangle), bujur sangkar (square), bujur sangkar tumpul (rounded square), oval, atau lingkaran (circle) |
| Line | Kontrol untuk membentuk objek garis pada form |
| Image | Kontrol untuk menampilkan objek gambar dari file bitmap (.BMP), ikon (.ICO), metafile (.WMF) pada form. |
| Data | Kontrol untuk mengakses data dalam database |
| OLE | Kontrol untuk menghubungkan objek dari aplikasi lain ke dalam Visual Basic |

# DASAR-DASAR VISUAL BASIC 6.0

* 1. **Variabel**

Variabel adalah sebuah tempat untuk menampung data dalam memori komputer sesuai dengan jenis data yang digunakan. Untuk membuat sebuah variabel terdapat ketentuan sebagai berikut:

* Harus dimulai dengan suatu huruf
* Tidak boleh mengandung titik, spasi atau special karakter lainnya, untuk memisahkan antar huruf digunakan tanda *underscore* ( \_ )
* Tidak lebih dari 255 huruf
* Nama dari kontrol, forms, classes dan module tidak melebihi 40 karakter
* Tidak boleh menggunakan keyword yang sudah ada pada Visual Basic
* Tidak membedakan huruf besar dan huruf kecil (no case sensitive)

Setiap variable yang dideklarasikan dalam pemograman visual diberikan sebuah maxifier yang mengatur jangkauan kehidupan dari variable tersebut. modifier dalam visual basic adalah

|  |  |
| --- | --- |
| **Keyword** | **Digunakan pada** |
| Public | Berlaku pada level modul |
| Private | Berlaku pada level modul |
| Dim | Berlaku pada level modul dan level procedure |
| Static | Berlaku pada level procedure |

Contoh penulisan variabel:

|  |  |
| --- | --- |
| **PENULISAN YANG BENAR** | **PENULISAN YANG SALAH** |
| Namamahasiswa | Nama Mahasiswa |
| Gergerkalong21 | 21Gegerkalong |
| Harga\_total | Harga.total |

* 1. **Konstanta**

Konstanta adalah variabel yang nilai datanya bersifat tetap dan tidak bisa diubah.

Jadi konstanta adalah juga variabel bedanya adalah pada nilai yang disimpannya. Jika nilai datanya sepanjang program berjalan tidak berubah-ubah, maka sebuah varibel lebih baik diperlakukan sebagai konstanta.

Pada sebuah kode program, biasanya nilai data dari konstanta diberikan langsung di bagian deklarasi konstanta. Sedangkan untuk variabel biasanya hanya ditentukan nama variabel dan tipe datanya tanpa isian nilai data.

Aturan penamaan variabel juga berlaku untuk penamaan konstanta., demikian juga aturan penetapan tipe data.

Contoh jika kita membuat program perhitungan matematika yang menggunakan nilai phi (3.14159) yang mungkin akan muncul dibanyak tempat pada kode program, kita dapat membuat phi sebagai konstanta.

Penggunaan konstanta phi akan lebih memudahkan penulisan kode program dibanding harus mengetikkan nilai 3.14159 berulang-ulang.

* 1. **Jenis Tipe Data**

Setiap data yang digunakan dalam Visual Basic memiliki tipe datanya masing-masing. Tipe data akan menentukan bentuk dan ukuran dari sebuah data.

|  |  |  |
| --- | --- | --- |
| **Tipe Data** | **Ukuran Storage** | **Jangkauan** |
| Byte | 1 Byte | 0 s/d 255 |
| Boolean | 2 Byte | True atau False |
| Integer | 2 Byte | -32768 s/d 32767 |
| Long | 4 Byte | -2,147,483,648 s/d 2,147,483,647 |
| Single | 4 Byte | -3.40282e38 s/d -1.401296e-45 (-) |
|  |  | 1.401296e-45 s/d 3.402823e38 (+) |
| Double | 8 Byte | -1,797691348623e308 s/d -.940654844127e324(-) |
|  |  | 4.940654844127e324 s/d 1,797691348623e308 (+) |
| Currency | 8 Byte | -922,337,203,685,477.5808 s/d 922,337,203,685,477.5807 |
| Decimal | 14 Byte | 7.92E+028 |
| Date | 8 Byte | 1 Januari 100 s/d 31 Desember 9999 |
| Object | 4 Byte | Mengacu pada objek tertentu |
| String | Panjang dari string | 1 sampai lebih kurang 65400 |
| Variant | 16 Byte | Sembarang angka sampai jangkauan jenis double atau jenis string |

* 1. **Operator**

Visual Basic menyediakan operator aritmatika, komparasi dan logika, salah satu hal yang harus dipahami adalah tata urut dari masing-masing operator tersebut sehingga mampu membuat ekspresi yang akan menghasilkan nilai benar.

*Operator aritmatika*

|  |  |
| --- | --- |
| **Nama Operator** | **Tanda Operator** |
| Pangkat | ^ |
| Negatif | - |
| Kali dan Bagi | \*, / |
| Pembagian Bulat | \ |
| Sisa Bagi | Modulus |
| Tambah dan Kurang | +,- |
| Penggabungan string | & |

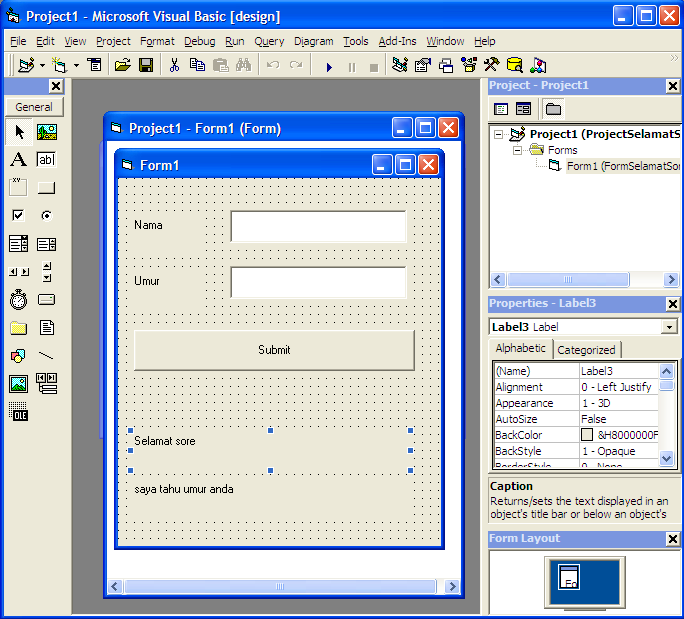
*Operator Komparasi*

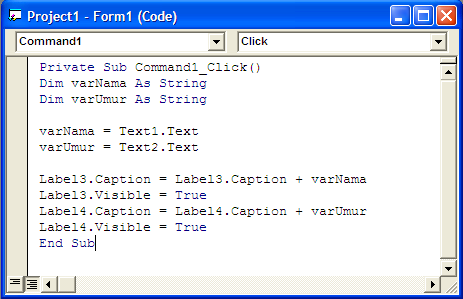
|  |  |
| --- | --- |
| **Nama Operator** | **Tanda Operator** |
| Sama (=) | = |
| Tidak sama | <> |
| Kurang dari | < |
| Lebih dari | > |
| Kurang dari atau sama | <= |
| Lebih dari atau sama | >= |
| Like | Like |

*Operator Logika*

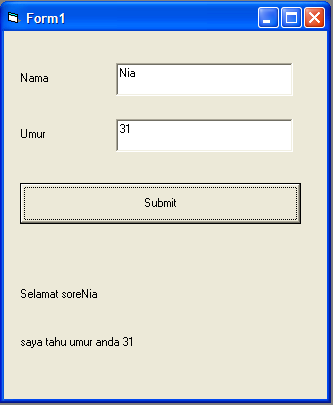
|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Operator** | **Tanda Operator** | **Nama Operator** | **Tanda Operator** |
| Not | NOT | Or | OR |
| And | AND | Xor | XOR |

**Contoh Program-3 membuat nama variabel:**





|  |  |  |
| --- | --- | --- |
| **Kontrol** | **Nama** | **Caption/Text** |
| Label1 | Label1 | Nama |
| Label2 | Label2 | Umur |
| Label3 | Label3 | Selamat Sore |
| Label4 | Label4 | Saya tahu umur anda |
| Text1 | Text1 | - |
| Text2 | Text2 | - |
| Command1 | Command1 | Submit |



“Semua nama kontrol disini belum dirubah, anda bisa mencoba merubah-rubah nama kontrol di properties sebagai latihan di rumah”

Soo… yang harus anda diperhatikan!

* Pertama-tama anda perlu mendeklarasikan variable terlebih dahulu, Mendeklarasikan variable:

Dari contoh diatas: Dim varNama As String

* + Nama variable : varNama
  + Tipe data : String
* Kemudian anda perlu menyimpan variabel tersebut dengan operasi penugasan:

varNama = Text1.Text

* artinya berapapun nilai yang diinputkan pada text box Text1 akan tercopy ke varNama
* Kemudian panggil data variabel yang sudah disimpan
* Label3.Caption= Label3.Caption + varNama

# KONTROL PADA VISUAL BASIC

Kontrol merupakan bagian terpenting dari program Visual Basic yang berfungsi untuk menyusun sebuah program aplikasi.

Kontrol adalah sebuah komponen atau objek yang diletakkan dalam form. Setiap kontrol adalah bagian yang berdiri sendiri tetapi saling berhubungan satu sama lain sehingga membentuk program aplikasi.

Contoh dari kontrol Visual Basic adalah Text Box, Rectangle, Command Button dan beberapa kontrol lain yang dapat ditempatkan pada form untuk menampilkan data, menjalankan suatu aksi, atau membuat form lebih mudah dibaca.

**Kategori Kontrol**

Kontrol pada Visual Basic dibagi menjadi tiga kategori sbb:

* Kontrol Intrinsic 🡪 contoh: Command Button dan Frame. Kontrol-kontrol tersebut terdapat didalam Visual Basic. Kontrol intrinsic selalu berada didalam jendela toolbox, tidak seperti kontrol ActiveX dan objek-objek tambahan lainnya yang dapat ditambahkan dan dihapus dari bagian toolbox.
* Kontrol ActiveX 🡪 merupakan kontrol aktif yang aktif sebagai file terpisah denga ekstensi .OCX. yang termasuk dalam kontrol ActiveX adalah kontrol-kontrol yang tersedia dalam semua versi visual Basic, contohnya DataCombo dan Data List serta kontrol-kontrol yang tersedia dalam versi Profesional Edition dan Enterprise Edition seperti Listview, Toolbar, Animation, dan Tabbed Dialog.

**Nama Kontrol**

Anda mungkin tetap ingin menggunakan nama default, akan tetapi jika anda mempunyai beberapa kontrol dengan tipe sama, anda dapat mengubah nama kontrol tersebut dengan nama yang lebih deskriptif.

Ketentuan untuk memberi nama objek form dan kontrol adalah sebagai berikut:

* Nama harus diawali dengan karakter.
* Nama terdiri dari karakter, angka dan garis bawah”\_”.
* Nama tidak boleh mengandung karakter khusus, misalnya titik, koma, slash dan lain-lain.
* Panjang maksimal nama adalah 40 karakter.

**Nilai Sebuah Kontrol**

Semua kontrol mempunyai sebuah property yang dapat Anda gunakan untuk menampung atau menerima nilai hanya dengan mengacu pada kontrol tersebut tanpa menggunakan property Name.

Tabel berikut menampung daftar dari property Value yang sesuai dengan masing-masing kontrol:

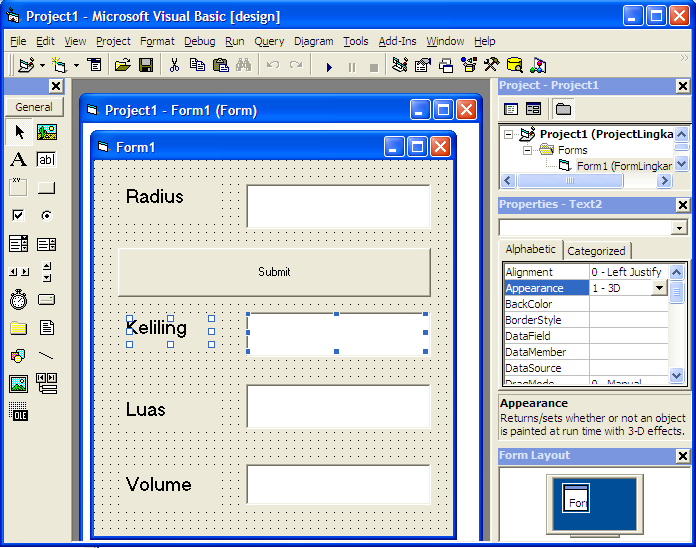
|  |  |
| --- | --- |
| **Kontrol** | **Nilai** |
| CheckBox | Value |
| Combo box | Text |
| Command button | Value |
| Command dialog | Action |
| Data | Caption |
| DataCombo | Text |
| DataGrid | Text |
| DataList | Text |
| Directory list box | Path |
| File list box | FileName |
| FlexGrid | Text |
| Frame | Caption |
| Horizontal scroll bar | Value |
| Image | Picture |
| Label | Caption |
| Line | Visible |
| List box | Text |
| Option button | Value |
| Picture box | Picture |
| Shape | Shape |
| Text box | Text |
| Timer | Enable |
| Vertical scroll bar | Value |

**Memberi Nama Kontrol**

Secara default, Visual Basic 6.0 menyediakan 21 kontrol yang ditampung dalam satu batang toolbar yang bernama Toolbox. Selain 21 kontrol default tersebut, sebenarnya masih banyak kontrol lain yang dapat digunakan untk menyusun sebuah program aplikasi. Hal tersebut mungkin akan membingungkan anda selaku pembuat program. Untuk itu disarankan memberi nama awalan untuk masing-masing kontrol dalam membuat sebuah program aplikasi.

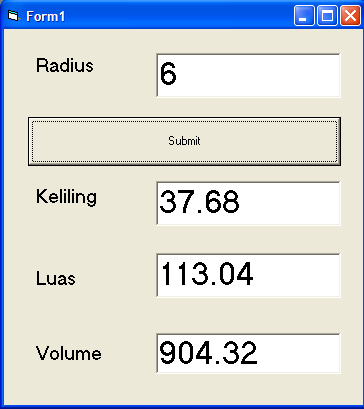
|  |  |  |
| --- | --- | --- |
| **Tipe Kontrol** | **Awalan** | **Contoh** |
| 3D Panel | pnl | pnlKelompok |
| ADO Data | ado | adoPenjualan |
| Animated Button | ani | aniBola |
| CheckBox | chk | chkTampilan |
| Combo box | cbo | cboKota |
| Command button | cmd |  |
| Command dialog | dlg |  |
| Communications | com |  |
| Data | dat |  |
| DataCombo | dbcbo |  |
| DataGrid | dgd |  |
| DataList | dbl |  |
| Data Repeater | drp |  |
| Date Picker | dtp |  |
| Directory list box | dir |  |
| File list box | fil |  |
| Frame | fra |  |
| Horizontal scroll bar | hsb |  |
| Image | img |  |
| Label | lbl | lblJudul |
| Line | lin |  |
| List box | lst |  |
| Option button | opt |  |
| Picture box | pic |  |
| Shape | shp |  |
| Text box | txt | txtNama |
| Timer | tmr |  |
| Vertical scroll bar | vsb |  |

**Contoh Program-4 menggunakan operator aritmatika dan konstanta:**





Hasil setelah di jalankan:



Perhatikan pada penulisan program:

Konstanta 🡪 phi = 3.14

# STRUKTUR KEPUTUSAN

**IF\_THEN**

Ekspresi berkondisi digunakan pada blok pernyataan khusus yang disebut dengan struktur keputusan, dan fungsinya mengendalikan apakah pernyataan lain pada program anda dijalankannya dan bagaimana urut-urutan dijalankannya.

Struktur keputusan **If\_Then** mengizinkan anda mengevaluasi sebuah kondisi pada program dan melakukan tindakan berdasarkan hasilnya.

Cara penulisannya:

Pernyataan **If\_Then\_Else** terdiri dari 3 bagian:

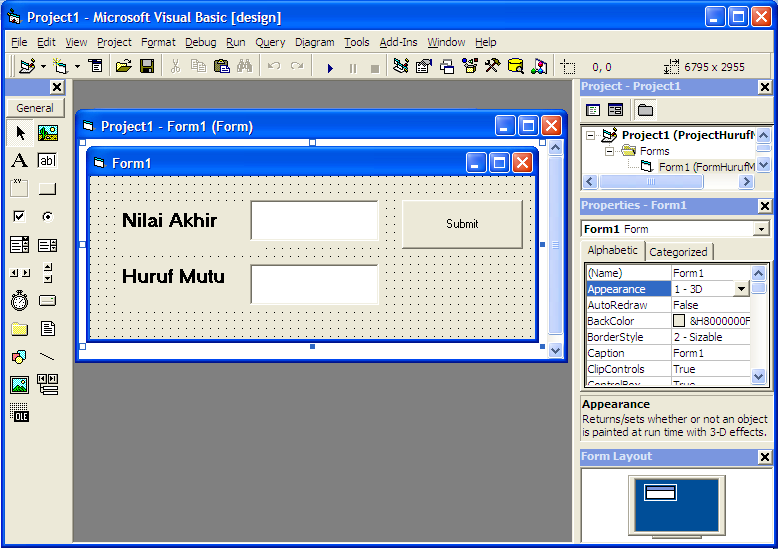
|  |  |
| --- | --- |
| Bagian | Keterangan |
| Kondisi | Kondisi dapat terdiri dari satu jenis ekspresi atau lebih, yaitu ekspresi numeric atau ekspresi string yang digunakan untuk menguji kebenaran data dengan menghasilkan dua logika benar (True) atau salah (False) |
| Statement | Statement bersifat opsional/ pilihan, tetapi menjadi wajib digunakan apabila dalam pernyataan If tidak ada pilihan Else. Satu atau lebih statement dipisahkan penulisannya dengan tanda titik dua (:)  Statement hanya akan dieksekusi jika kondisi bernilai benar (True) |

**Program-5:**

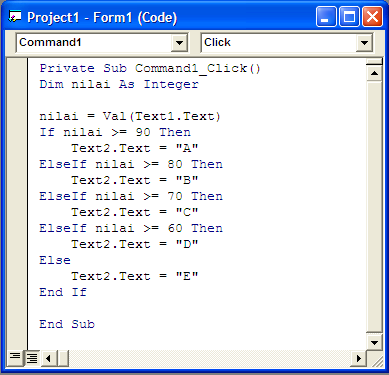
Anda diminta memasukkan nilai akhir dimana sesuai dengan ketentuan sbb:

|  |  |
| --- | --- |
| **Nilai Akhir** | **Huruf Mutu** |
| 90-100 | A |
| 80-89 | B |
| 70-79 | C |
| 60-69 | D |
| 0-59 | E |

Buatlah pada Jendela Form seperti dibawah ini:



Klik ganda pada tombol Command sehingga dapat ditulis Code-nya sbb:



**Contoh program-6 menggunakan If…Then:**

Buatlah program untuk menghitung jumlah barang yang dibeli!

Jawab:

1. Buatlah design form seperti gambar dibawah ini yang terdiri dari label, text box, dan command button



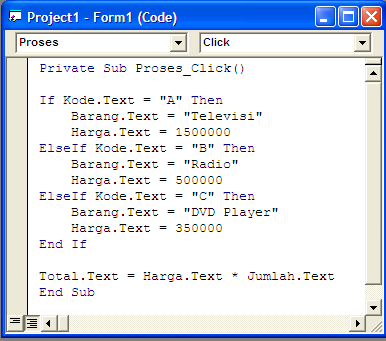
1. Ubahlah property masing-masing kontrol dengan ketentuan sbb:

|  |  |  |
| --- | --- | --- |
| **Kontrol** | **Nama** | **Caption/Text** |
| Text1 | Pembeli | Pembeli |
| Text2 | Kode | Kode |
| Text3 | Jumlah | Jumlah |
| Text4 | Barang | Barang |
| Text5 | Harga | Harga |
| Text6 | Total | Total |
| Command1 | Proses | PROSES |

1. Ketentuan perhitungan untuk kontrol Proses adalah sebagai berikut:

* Barang diisi dengan Televisi jika Kode bernilai A, diisi Radio jika bernilai B, dan diisi DVD Player jika kode bernilai C
* Harga diisi 1500000 jika kode bernilai A, diisi 500000 jika bernilai B, diisi 350000 jika bernilai C
* Total diisi dengan hasil perhitungan antara nilai Harga dan Jumlah

Jika sudah selesai maka klik dua kali pada kontrol Proses dan ketik kode program berikut sesuai dengan ketentuan diatas:



**Program-7 Belajar Menggunakan Kontrol Frame dan Option Button**

Anda bisa memodifikasi contoh sebelumnya dengan mengganti kontrol label dan text box untuk Kode dengan kontrol option button

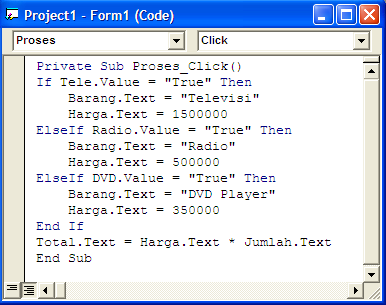
1. Tambahkan kontrol Frame dalam Form dan isilah dengan tiga buah kontrol Option Button didalam kontrol Frame



1. Ubahlah property masing-masing kontrol dengan ketentuan sbb:

|  |  |  |
| --- | --- | --- |
| **Kontrol** | **Nama** | **Caption/Text** |
| Frame | Kode | - |
| Option1 | Tele | Televisi |
| Option2 | Radio | Radio |
| Option3 | DVD | DVD Player |

1. Klik dua kali kontrol proses, dan ubah kode program tersebut dengan ketentuan proses berikut:



Setelah di run, maka hasilnya menjadi:



**SELECT\_CASE**

Pernyataan ini biasanya digunakan untuk mengganti penggunaan **If-Then-Else**. **Select Case** memiliki fungsi yang sama dengan If\_Then hanya saja penggalan pernyataan akan terlihat lebih terstruktur, sehingga coding akan lebih mudah untuk dibaca. **Select\_Case** dapat digunakan untuk melakukan pemilihan kondisi yang jumlahnya banyak atau bertingkat. Berikut ini adala bentuk penulisan dari pernyataan **Select Case**:

Keterangan:

|  |  |
| --- | --- |
| **Bagian** | **Keterangan** |
| Kondisi | Kondisi dapat berupa ekspresi numerik atau ekspresi string |
| Ekspresi-n | Ekspresi-n dapat menggunakan ekspresi **To** dan operator **Is.**   * Kata kunci To menentukan range nilai * Menggunakan kata kunci Is dengan operator pembanding (kecuali Is dan Like) untuk menentukan range nilai |
| Statement-n | Statement-n bersifat opsional.  Satu atau lebih statement yang akan dieksekusi jika kondisi sesuai dengan bagian Ekspresi-n |

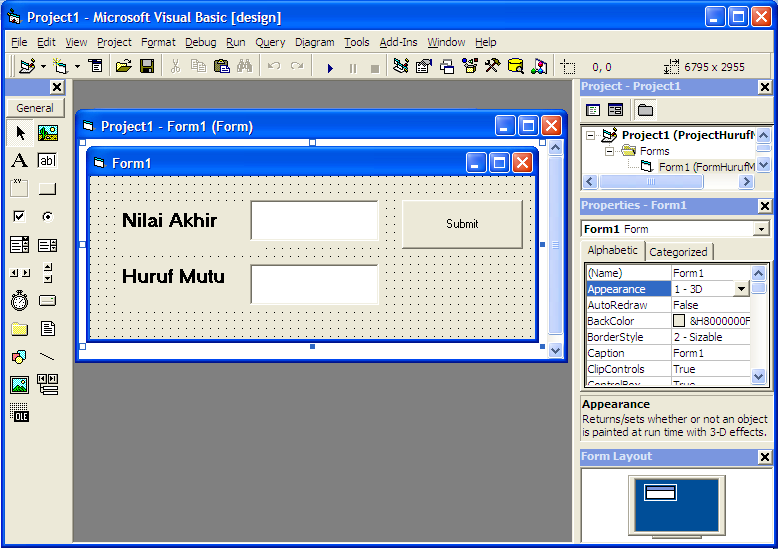
Anda dapat menggunakan ekspresi ganda dalam setiap case. Sebagai contoh, perhatikan penulisan kode program berikut:

Operator pembanding Is tidak sama dengan kata kunci Is yang digunakan dalam pernyataan **Select\_Case**

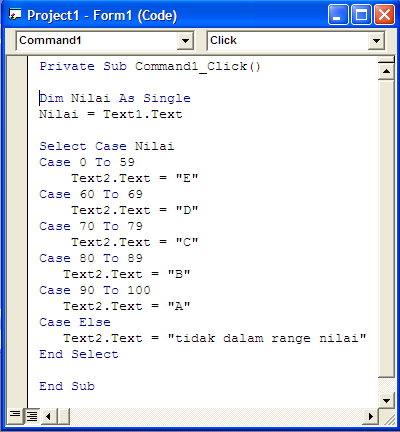
**Program-8**:

Marilah kita buat program menggunakan perintah **Select\_Case** menggantikan perintah **If\_Else** seperti pada contoh Program-5:

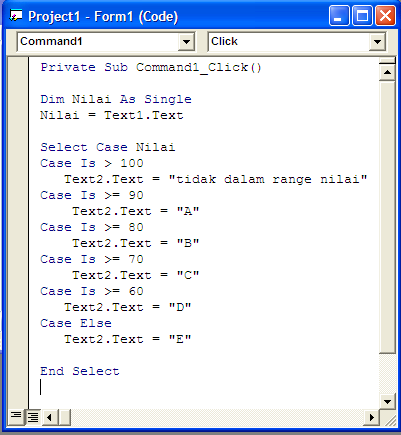
|  |  |
| --- | --- |
| **Nilai Akhir** | **Huruf Mutu** |
| 90-100 | A |
| 80-89 | B |
| 70-79 | C |
| 60-69 | D |
| 0-59 | E |



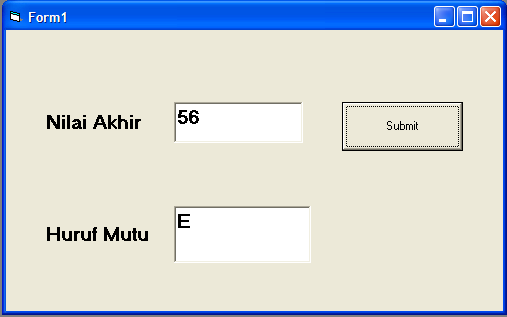
* **Dengan menggunakan ekspresi To**



* **Dengan menggunakan ekspresi Is**



Klik tombol Start pada Toolbar untuk menjalankan program.

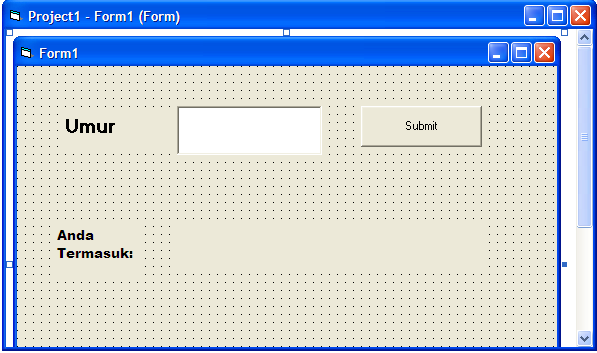


**Latihan**

Kasus:

Buatlah program untuk menampilkan tulisan kategori setelah inputan umur diberikan, dengan ketentuan sbb:

|  |  |
| --- | --- |
| **Usia (Tahun)** | **Kategori** |
| >=60 | Lansia |
| 20-59 | Dewasa |
| 12-20 | Remaja |
| 6-12 | Anak-anak |
| 0 sd 5 | Balita |



# 5.STRUKTUR PENGULANGAN

Struktur pengulangan (loop) digunakan untuk mengulang sebagian dari suatu rutin (bagian dari suatu kode program tertentu) berdasarkan suatu kondisi tertentu. Dengan demikian, Anda tidak perlu menulis ulang rutin tersebut sebanyak pengulangan yang diinginkan.

Pengulangan akan dilakukan selama kondisi tersebut memenuhi syarat (bernilai ***True***), apabila kondisi tersebut sudah tidak memenuhi syarat, maka pengulangan akan dihentikan.

* **DO\_WHILE**

Pengulangan **Do\_While** merupakan pengulangan yang dilakukan secara berulang-ulang selama (While) suatu kondisi memenuhi syarat (bernilai ***True***) sampai kondisi tersebut tidak memenuhi syarat.

Bentuk penulisan dari pengulangan Do\_While adalah:

Atau anda dapat menggunakan bentuk penulisan berikut:

Keterangan:

* *Condition*: kondisi bersifat opsional. Bisa berupa ekspresi numerik atau string yang menghasilkan dua nilai logika true atau false
* *Statement*: satu atau lebih statement yang akan diulang selama bernilai True
* **DO\_UNTIL**

Pengulangan **Do\_Until** dapat dilakukan sebagai kebalikan dari pengulangan yang menggunakan Do\_While. Do\_Until digunakan untuk melakukan proses pengulangan selama kondisi yang dibandingkan tidak memenuhi syarat (bernilai ***False***). Pengulangan ini akan dilakukan secara berulang-ulang sampai (until) kondisi memenuhi syarat (bernilai ***True***).

Atau anda dapat menggunakan bentuk penulisan berikut:

Dim **Nama\_variabel** As **Type**

Contoh:

Menampilkan bilangan bulat ganjil dari satu sampai dengan 11, ini merupakan looping dengan akumulator yang dapat dituliskan sebagaimana berikut:

Bilangan = 1

Do

List1.AddItem bilangan

Bilangan = bilangan +2

Loop until bilangan>11

* **FOR-NEXT**

Pengulangan **For\_Next** digunakan untuk mengulangi suatu perintah dalam jumlah yang telah ditentukan. Dengan For\_Next, anda dapat menentukan berapa kali pengulangan yang akan dilakukan.

Bentuk penulisannya:

**If** *condition1* **Then**

*Statement executed if condition1 is True*

**ElseIf** *condition2* **Then**

*Statement executed if condition2 is True*

*[Additional ElseIf clauses and statements can be placed here]*

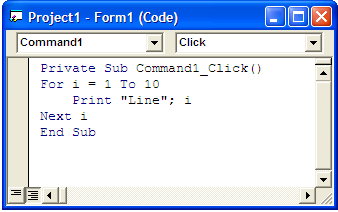
**Else**

*Statement executed none of the conditions is True*

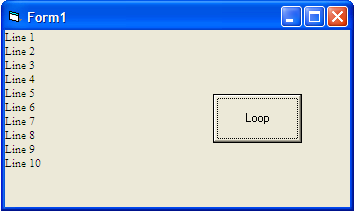
**End If**

Contoh program menampilkan informasi dengan loop For-Next

1. Pada Visual Basic, bukalah sebuah project baru
2. Gunakan kontrol CommandButton untuk membuat sebuah tombol perintah disebelah kanan form.
3. Ganti nama kontrol Command menjadi “Loop”.
4. Klik dua kali pada jendela Form, ubahlah Font menjadi Times New Roman (anda bisa saja menggunakan font apa saja, tetapi font TrueType adalah yang terbaik karena bisa ditampilkan dengan berbagai ukuran dan akan tampak sama di layar maupun saat dicetak)
5. Ubahlah property AutoRedraw menjadi True
6. Klik ganda pada CommandButton lalu ketik pernyataan program berikut ini:



1. Klik tombol start pada Toolbar



* **WHILE-WEND**

Pengulangan dengan While-Wend adalah bentuk pengulangan terkendali dengan proses akan diulang selama kondisi yang diberikan masih terpenuhi.

Bentuk penulisan dari pengulangan While\_Wend adalah:

**Select Case** *Kondisi*

**Case** *Ekspresi-n*

[*Statement-n*]

**Case Else**

[*Statement\_else*]

**End Select**

Variable control pada jenis looping tidak otomatis bertambah/berkurang seperti pada looping dengan For\_Next, sehingga penambahan atau pengurangan nilai dilakukan dengan menggunakan akumulator yang dituliskan dengan:

**var=var+step**

Meskipun hal ini bukan suatu keharusan. Karena tidak setiap looping membutuhkan akumulator, variable yang terus menerus bertambah atau berkurang.

Contoh:

Menampilkan bilangan bulat ganjil dari satu sampai dengan 11, ini merupakan looping dengan akumulator yang dapat dituliskan sebagai berikut:

bilangan =1

While bilangan <= 11

List.AddItem bilangan

Bilangan = bilangan + 2

Wend

**Latihan:**

1. Buatlah program pengulangan untuk menampilkan hasil penjumlahan dari rumus deret yang dituliskan dengan:

Private Sub Command1\_Click()

S = 0

n = 5

For x = 1 To n

S = S + (x - 1) ^ 2

Text1.Text = S

Next x

End Sub

**Latihan**

1. Buatlah program untuk menampilkan barisan N buah bilangan yang dimulai dengan 1, bilangan kedua berselisih 1 dari bilangan pertama, bilangan kedua berselisih 2 dari bilangan kedua, …. Bilangan ke n beselisih (n-1)

1 2 4 7 11 16 22 …

1. Buatllah program untuk menampilkan barisan bilangan N bilangan sebagai berikut:

1 -2 3 -4 5 -1 2 -3 4 -5 1 -2 3 -4 5

1. Membuat program untuk menghitung nilai n! apabila nilai n telah diinputkan maka dapat diketahui nilai factorial bilangan tersebut!

* **PENGULANGAN BERTINGKAT**

Pada contoh-contoh sebelumnya hanya dilakukan satu pengulangan. Padahal dalam banyak aplikasi pemograman menggunakan pengulangan lebih dari satu misalkan pada pengolahan matriks, diperlukan pengulangan untuk baris dan pengulangan untuk kolom pada setiap baris. Pengulangan yang ada didalam pengulangan inilah yang disebut dengan pengulangan bertingkat.

Pengulangan bertingkat secara umum dapat dituliskan dengan bentuk penulisan sbb:

**Case**  1 To 4, 7 To 9, 11, 13, Is>Angka\_Max

Contoh 1:

Menampilkan n barus dari barisan bilangan 1 2 3 4 5 sebagai berikut:

**1 2 3 4 5**

**1 2 3 4 5**

**1 2 3 4 5**

**………**

Dapat ditulis dengan:

Dim N As Integer

N = 4

For baris = 1 To N

kal = ""

For i = 1 To 5

kal = kal & Str(i)

Next i

List1.AddItem kal

Next baris

Contoh 2:

Menampilkan 5 baris dari kombinasi bilangan 1 2 3 4 5 sebagai berikut:

**1**

**1 2**

**1 2 3**

**1 2 3 4**

**1 2 3 4 5**

Perhatikan bahwa pada baris ke 1 hasilnya adalah dari 1 sampai dengan 1, baris ke 2 hasilnya adalah dari 1 sampai 2, pada baris ke 3 hasilnya adalah dari 1 sampai dengan 3. Sehingga dapat dikatakan untuk baris ke i hasilnya adalah dari 1 sampai dengan i. dan dapat dituliskan sebagaimana berikut:

For i = 1 To 5

kal = " "

For j = 1 To i

kal = kal & Str(j)

Next j

List1.AddItem kal

Next i

Contoh 3:

Menampilkan n barus dari barisan bilangan 1 2 3 4 5 sebagai berikut:

**1 2 3 4 5**

**5 4 3 2 1**

**1 2 3 4 5**

**5 4 3 2 1**

**………**

perhatikan bahwa pada baris ganjil hasilnya 1 2 3 4 5 dan pada baris genap adalah 5 4 3 2 1, maka perlu untuk mengetahui apakah baris itu baris genap/ganjil untuk menentukan nilai awal. Dapat ditulis dengan:

N = 10

For baris = 1 To N

If baris Mod 2 = 0 Then

awal = 5: akhir = 1

langkah = -1

Else

awal = 1: akhir = 5

langkah = 1

End If

kal = ""

For i = awal To akhir Step langkah

kal = kal & Str(i)

Next i

List1.AddItem kal

Next baris