

Menggunakan DAO (Data Access Objects) Kode

Tutorial ini menjelaskan bagaimana Anda dapat menggunakan kode DAO secara langsung dalam aplikasi VB6 Anda bukannya terikat ke kontrol Visual Basic.

Mengapa Gunakan Kode Daripada Data Control?

Keuntungan dari menggunakan kontrol data adalah bahwa Anda bisa menyatukan bentuk padat entri data tanpa menulis kode VB banyak. Metode ini bekerja dengan baik untuk kecil, satu kali proyek yang perlu diselesaikan cepat.

Kerugian dari menggunakan kontrol data adalah bahwa setelah proyek selesai, tidak selalu mudah untuk memodifikasi bentuk entri data atau mengadaptasi bentuk jadi untuk proyek lain entri data. Juga, bentuk dibangun menggunakan kontrol data tidak selalu mudah untuk debug atau mempertahankan karena sebagian tindakan berjalan di dalam data kontrol itu sendiri. Jika Anda berpikir proyek Anda perlu diubah atau dipertahankan oleh programmer lain, kontrol data mungkin tidak menjadi pilihan terbaik Anda.

Keuntungan dari menggunakan kode VB lengkap untuk menghasilkan bentuk entri data adalah bahwa Anda memiliki total kontrol atas semua aspek dari proses. Anda memutuskan kapan harus membuka database dan recordset, dan Anda mengontrol operasi membaca dan menulis juga. Kemampuan ini dapat menjadi keuntungan nyata dalam multiuser (file sharing) di mana pengaturan peningkatan lalu lintas dapat menyebabkan mengunci konflik dalam program-program yang menggunakan kontrol data. Keuntungan lain dari menggunakan kode VB untuk bentuk entri data Anda adalah bahwa Anda dapat membuat kode generik yang dapat Anda kembali dalam semua proyek database Anda. Bila Anda memiliki sepenuhnya debugged set rutinitas entri data, Anda dapat dengan cepat membuat bentuk-bentuk baru tanpa coding tambahan banyak. Karena bentuk bergantung pada rutinitas generik, mereka juga mudah untuk memodifikasi dan mempertahankan di masa depan.

Kelemahan utama untuk menggunakan VB kode untuk menciptakan bentuk-bentuk entri data adalah bahwa Anda harus menangani semua proses sendiri,

Anda dapat mengasumsikan apa-apa. Misalnya, menemukan dan memperbarui rekor tunggal dalam tabel data mengharuskan Anda account untuk semua proses berikut:

- Membuka database
- Membuka recordset
- Menemukan data yang diminta
- Memuat kontrol masukan dari recordset
- Penanganan semua tindakan pengguna selama proses entri data
- Menulis kontrol diperbarui kembali ke recordset

Anda juga membutuhkan suatu cara bagi pengguna untuk menelusuri data. Dalam memberikan kontrol data, Anda menggunakan VCR-gaya panah navigasi.

Meskipun ini tanggung jawab tambahan, menulis data Anda bentuk entri dengan kode VB memberikan Anda kontrol lebih besar atas proses dan dapat menghasilkan bentuk yang mudah untuk kedua programmer dan pengguna untuk menangani. Bahkan meskipun Anda harus melakukan sedikit baik coding untuk menciptakan rutinitas baru manajemen data, rutinitas ini sering dapat digunakan kembali dalam proyek-proyek masa depan dengan minimal re-coding.

Contoh database : **employee.MDB**

Dokumen ini menunjukkan Anda bagaimana untuk memproses sebuah database Access menggunakan kode saja. Database bernama "EMPLOYEE.MDB", dan didasarkan pada file karyawan sampel yang digunakan dalam tutorial pada file sekuensial. EMPLOYEE.MDB berisi tiga tabel: EmpMast, DeptMast, dan JobMast. Tabel yang terstruktur sebagai berikut:

1. Tabel : **EmpMast**

Field Name	Data Type	Comments
EmpNbr	AutoNumber	PK. Unik mengidentifikasi masing-masing karyawan dalam database.
EmpFirst	Text (50)	Nama depan karyawan
EmpLast	Text (50)	Nama belakang karyawan
DeptNbr	Number (Long Integer)	FK untuk PK dari tabel DeptMast. Mengidentifikasi departemen karyawan yang bekerja masuk
JobNbr	Number (Long Integer)	FK untuk PK dari tabel DeptMast. Mengidentifikasi departemen karyawan yang bekerja masuk.
HireDate	Date/Time	Tanggal karyawan dipekerjakan.
HrlyRate	Number (Single)	Tingkat jam kerja karyawan

SchedHrs	Number (Single)	Jumlah jam kerja karyawan per minggu.
----------	-----------------	---------------------------------------

2. Tabel : DeptMast

Field Name	Data Type	Comments
DeptNbr	Number (Long Integer)	PK; unik mengidentifikasi masing-masing departemen dalam database. Indeks PK dinamai idxDeptNbrPK dalam antarmuka akses (lihat di bawah).
DeptName	Text (50)	Nama departemen. Sebuah indeks non-unik didirikan pada bidang ini, dan indeks ini berganti nama idxDeptName .
Location	Text (50)	Lokasi departemen (bisa membangun, suite nomor, lantai, dll)

3. Tabel : JobMast

Field Name	Data Type	Comments
JobNbr	AutoNumber	PK; unik mengidentifikasi setiap pekerjaan dalam database. Indeks PK berganti nama menjadi idxJobNbrPK .
JobTitle	Text (50)	Jabatan (deskripsi). Sebuah indeks non-unik didirikan pada bidang ini, dan indeks ini dinamakan idxJobTitle .
MinRate	Number (Single)	Tingkat per jam minimum yang seseorang bekerja di posisi ini biasanya dibayar.
AvgRate	Number (Single)	Tingkat rata-rata per jam bahwa seseorang bekerja di posisi ini biasanya dibayar.
MaxRate	Number (Single)	Tingkat per jam maksimum yang seseorang bekerja di posisi ini biasanya dibayar.

How to Rename Indexes in the Access UI

DeptNbr is designated as the Primary Key field.

The "Yes (Duplicates OK)" option was selected for the DeptName field, which sets up a non-unique index for that field.

DeptMast : Table	
Field Name	Data Type
DeptNbr	Number
DeptName	Text
Location	Text

Field Properties

General | Lookup

Field Size: 50

Format:

Input Mask:

Caption:

Default Value:

Validation Rule:

Validation Text:

Required: No

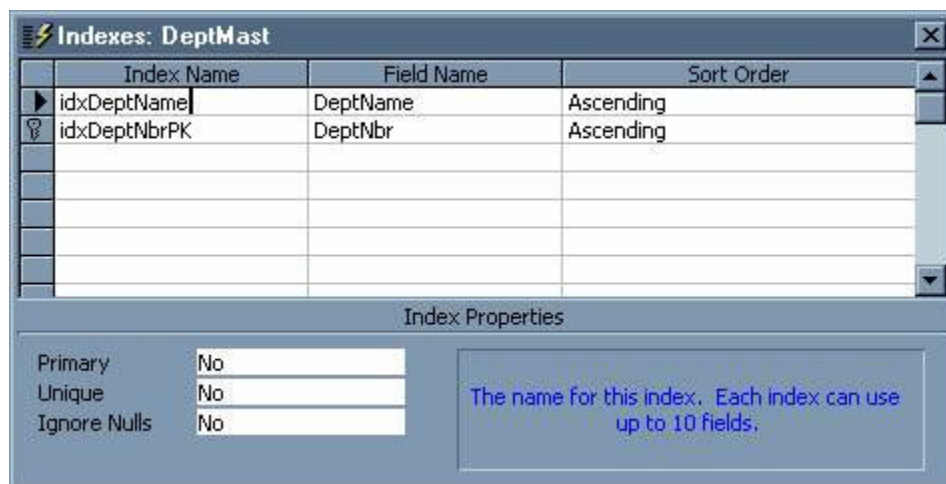
Allow Zero Length: No

Indexed: Yes (Duplicates OK)

Untuk memeriksa informasi yang terkait dengan indeks yang Anda tetapkan untuk meja, klik ikon Indeks (terlihat seperti petir) pada toolbar Akses.



Mengklik ikon Indeks menyebabkan kotak dialog Indeks (ditampilkan di bawah) yang akan ditampilkan. Perhatikan bahwa di bawah kolom "**Nama Indeks**", saya berganti nama menjadi indeks. Nama Indeks untuk field DeptNbr diubah dari "**primaryKey**" untuk "**idxDeptNbrPK**" (default Akses indeks nama untuk field primary key adalah "**primaryKey**") dan Indeks Nama untuk field DeptName diubah dari "**DeptName**" untuk "**idxDeptName**" (default Akses indeks nama untuk non-primer field kunci adalah sama dengan field Name).



The DAO Object Model

Sebuah bagian penting dari belajar bagaimana aplikasi program database Access untuk mendapatkan pengetahuan kerja dari model (Data Access Objects) objek DAO, ditampilkan pada halaman berikut. Model objek adalah representasi, atau peta konseptual, fungsi obyek dalam hal hirarki objek. Objek dalam model objek dikatakan "terbuka", yang berarti bahwa mereka adalah item yang dapat diprogram atau dikendalikan.

Objek dalam model objek yang disusun dalam berbagai tingkatan. Anda dapat menganggap tingkat ini seperti tingkatan dalam suatu hierarki. Tier paling atas dalam model objek DAO database JET mesin itu sendiri (DBEngine). Tingkat kedua terdiri dari kategorisasi tingkat tinggi objek. Yang ketiga, tingkatan keempat dan kelima, dll termasuk berbagai objek yang berbeda yang digunakan untuk mengakses fungsi yang kedua-tier benda mengandung. Anda melintasi tingkatan untuk menemukan objek yang ingin Anda gunakan.

Sekelompok objek serupa dapat dikombinasikan dalam hirarki sebagai koleksi. Secara umum, nama jamak ("Database", "Recordset", dll) koleksi dan nama-nama tunggal ("Database", "Recordset", dll) adalah obyek dengan koleksi.

Collection Syntax

Untuk mengacu pada suatu objek dalam koleksi, sintaksnya adalah:

```
CollectionName.Item("Key")
```

Atau :

```
CollectionName.Item(index)
```

Item adalah metode standar untuk semua koleksi, sehingga **item** selalu dapat dijatuhkan dari sintaks di atas, mengurangi ke :

```
CollectionName("Key")
```

Atau :

```
CollectionName(index)
```

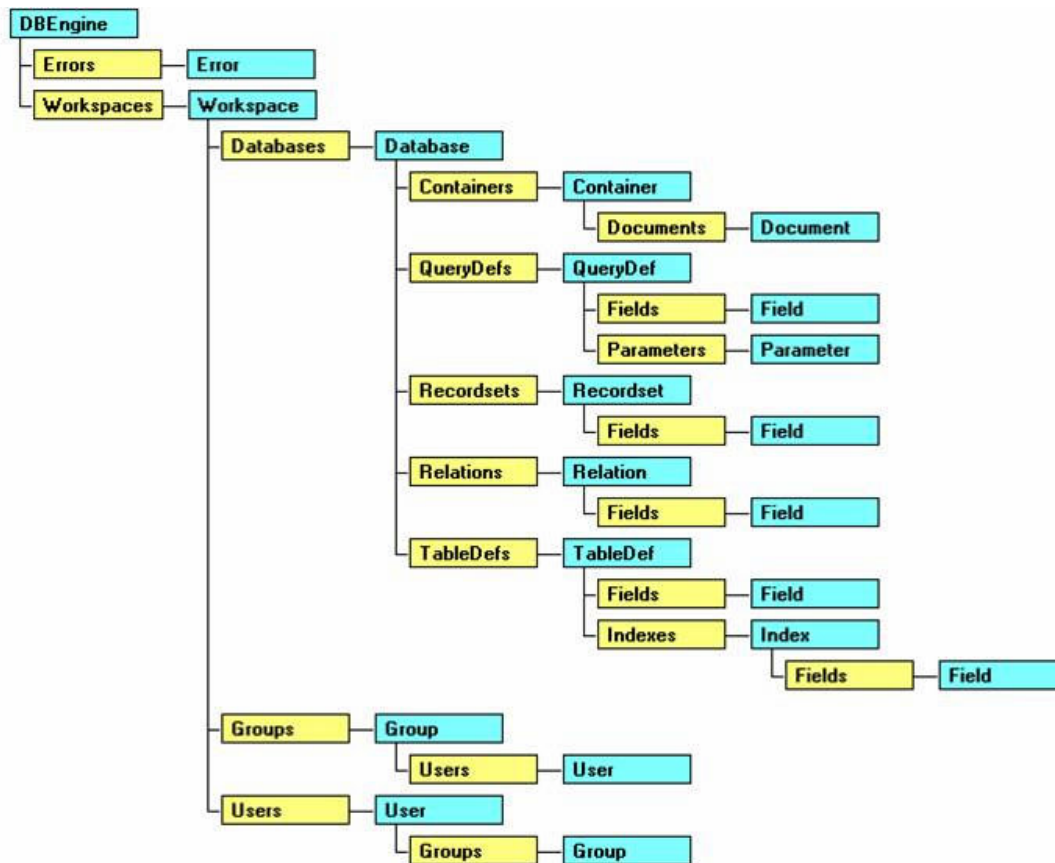
Sebuah objek kunci adalah sebuah string yang secara unik mengidentifikasi objek yang dalam koleksi. Misalnya, bidang DeptNbr dalam koleksi bidang tabel dapat disebut sebagai Fields ("DeptNbr"). Indeks mengacu pada posisi ordinal objek dalam koleksi, yang bisa berubah jika item yang ditambahkan ke atau dikeluarkan dari koleksi. Jika bidang DeptNbr adalah bidang pertama dalam koleksi Fields, itu bisa disebut sebagai Fields (0). (**Catatan:** Hal ini umumnya lebih baik untuk menggunakan kunci daripada indeks untuk mengakses item dalam koleksi Sejak indeks untuk item dalam koleksi dapat berubah, penggunaannya akan terbatas pada situasi seperti di mana Anda perulangan melalui item satu per satu, mungkin untuk menampilkan

informasi tentang setiap item, atau hanya untuk memverifikasi apa yang item dalam koleksi.)

Dalam melihat model objek DAO, Anda melihat baik jumlah koleksi dan benda-benda. Masing-masing obyek memiliki berbagai metode dan properti. Namun, dalam aplikasi VB / DAO biasa, Anda hanya akan perlu untuk berurusan dengan segelintir dari koleksi dan benda-benda. Kemungkinan besar, Anda akan menggunakan objek Workspace (tidak langsung), obyek database, dan Recordset, TableDefs, dan koleksi mungkin QueryDefs. Koleksi dan benda-benda yang digunakan dalam proyek sampel disajikan dalam dokumen ini, bersama dengan sifat mereka dan metode akan dibahas karena mereka ditemui, seperti yang kita memeriksa proses yang terjadi dalam setiap bentuk proyek.

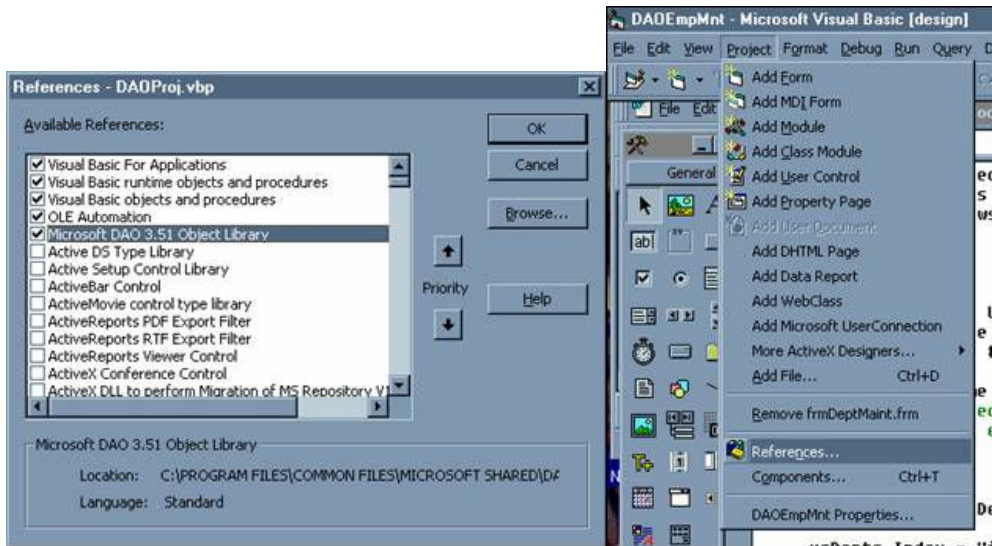
Referensi yang baik untuk masing-masing item dapat ditemukan dalam sistem bantuan Akses. Selain itu, ada sejumlah buku referensi yang baik tentang JET / DAO. Salah satu buku terbaik pada subjek adalah "***Panduan Microsoft Jet Database Engine Programmer***" oleh ***Dan Haught*** dan ***Jim Ferguson*** (Microsoft Press, 1997).

The DAO (Data Access Objects) Object Model



Using DAO in Your VB Project

VB Project yang akan memproses Access-gaya (JET) database harus mencakup referensi ke Microsoft DAO 3.51 Object Library. Untuk menyertakan referensi ini, pergi ke menu dan pilih Project VB Referensi. Dari kotak dialog yang muncul, periksa referensi (ditampilkan di bawah):



Catatan: Bila Anda menggunakan kontrol data, pengaturan referensi ini tidak diperlukan (VB akan mengatur secara otomatis).

The Sample Project (prjDAODemo)

Proyek sampel berisi tujuh bentuk dan satu modul standar, seperti yang dirangkum di bawah ini :

Forms:

Nama	Deskripsi
frmSplash	Splash screen
frmMainMenu	Menu utama, atau "switchboard" layar
frmHelp	Bentuk di mana file bantuan ditampilkan
frmDeptMaint	Digunakan untuk menambah, mengubah, atau menghapus record dari tabel DeptMast
frmJobMaint	Digunakan untuk menambah, mengubah, atau menghapus record dari tabel JobMast
frmEmpMaint	Digunakan untuk menambah, mengubah, atau menghapus record dari tabel EmpMast
frmReportMenu	Memungkinkan pengguna untuk memilih sebuah laporan (Crystal) yang akan dicetak

Modules:

Nama	Deskripsi
frmSplash	Standar modul yang berisi variabel publik, salam, dan fungsi.

modCommon

Kita akan melihat setiap bentuk dalam proyek sampel pada gilirannya, tapi pertama, mari kita lihat kode yang ada di modul standar (tercatat sedikit lebih jauh di bawah). Pada bagian deklarasi umum, Anda melihat pernyataan

```
Public gobjEmpDB As Database
```

"*Database*" adalah jenis tertentu dari variabel object. Setelah Microsoft DAO 3.51 Object Library telah dimasukkan dalam proyek Anda, variabel didefinisikan sebagai salah satu jenis objek hadir dalam model objek DAO (Database, TableDef, Recordset, Field, dll) dapat digunakan dalam proyek.

Selain rutin CenterForm dan fungsi GetAppPath, ada dua database yang terkait rutinitas, satu untuk membuka database dan satu untuk menutupnya. Dalam proyek ini, setiap bentuk yang menggunakan database panggilan sub OpenEmpDatabase ketika form tersebut dimuat dan panggilan sub CloseEmpDatabase ketika form dibongkar.

Method OpenDatabase

Para OpenEmpDatabase Sub berisi pernyataan tunggal:

```
Set gobjEmpDB = OpenDatabase(GetAppPath() & "EMPLOYEE.MDB")
```

Ingat bahwa gobjEmpDB dinyatakan sebagai variabel "Database", yang merupakan jenis khusus dari objek variabel. Tidak seperti VB variabel standar seperti "Integer" dan "String", variabel objek harus diinisialisasi dengan pernyataan Set. Pernyataan Set menetapkan referensi yang valid ke variabel objek.

OpenDatabase adalah metode dari objek Workspace dalam koleksi Workspace (dan tentu saja Workspace adalah kumpulan dari objek DBEngine). Sebuah objek Workspace ada untuk setiap sesi aktif dari mesin database Jet. Sesi melukiskan urutan operasi yang dilakukan oleh MS Jet. Sesi dimulai ketika pengguna log on dan berakhir ketika pengguna log off. Bila aplikasi Anda DAO mulai dijalankan, objek Workspace pertama dari koleksi Workspace (dirujuk

sebagai Workspace (0)) sudah dibuat. Ruang kerja (0) juga disebut sebagai ruang kerja standar.

Karena objek DBEngine dan default item yang Workspace (Workspace (0)) secara otomatis tersedia untuk aplikasi yang mencakup referensi ke Microsoft DAO 3.51 Library Obyek dan tidak perlu secara eksplisit direferensikan ketika menggunakan "OpenDatabase", meskipun mereka dapat , seperti dalam pernyataan berikut:

```
Set gobjEmpDB = DBEngine.Workspaces(0).OpenDatabase(GetAppPath() &
"EMPLOYEE.MDB")
```

Metode OpenDatabase memerlukan argumen string mengevaluasi dengan nama file database yang akan digunakan, diikuti oleh argumen opsional tidak digunakan dalam contoh di atas. Ini membuka file database diminta dan mengembalikan sebuah referensi obyek sesuai dengan variabel Database. Pernyataan di atas membuka file database EMPLOYEES.MDB, terletak di jalan yang sama seperti program VB, dan set up gobjEmpDB variabel untuk referensi database ini.

The Database Object

Setelah Anda memiliki referensi yang valid ke database melalui metode OpenDatabase, maka Anda dapat menggunakan berbagai metode dan properti dari objek database dalam kode, dengan menggunakan variabel objek Database. Metode yang umum digunakan dengan objek Database Execute, Tutup, dan OpenRecordset. Metode OpenRecordset adalah jauh metode yang paling umum digunakan dari objek Database dan digunakan secara bebas dalam beberapa bentuk proyek ini. OpenRecordset akan dibahas sedikit kemudian, ketika form-form yang terlihat.

Metode Execute digunakan untuk menjalankan pernyataan SQL yang mengembalikan baris tidak pada database. Hal ini pada dasarnya termasuk non-SELECT query: query DDL seperti CREATE TABLE atau DROP TABLE, dan permintaan tindakan seperti UPDATE atau DELETE. Metode Jalankan membutuhkan argumen string yang berisi pernyataan SQL yang benar.

Sebagai contoh, pernyataan berikut akan memberikan semua karyawan meningkatkan 10%:

```
gobjEmpDB.Execute "UPDATE EmpMast " _  
& "SET HrlyRate = HrlyRate * 1.1"
```

Catatan: metode Execute, sementara cukup berguna, tidak digunakan dalam proyek sampel.

Metode **Close** hanya menutup database dan dikodekan sebagai ***gobjEmpDB.close*** (lihat Sub CloseEmpDatabase).

Kata Kunci Yang Tidak Berarti Apa-apa

Ketika Anda selesai menggunakan variabel object (baik itu objek database variabel, variabel objek recordset, atau jenis lain dari variabel object), Anda harus menetapkan bahwa variabel objek untuk Tidak ada kata kunci VB, yang disassociates variabel objek dari yang sebenarnya objek dan melepaskan memori dan sumber daya sistem yang berhubungan dengan objek yang mengacu variabel. Jadi, ketika database ditutup, variabel objek database harus di set ke Nothing, seperti dalam:

```
Set gobjEmpDB = Nothing
```

Code for the standard module modCommon:

```
Option Explicit

Public gobjEmpDB As Database

Public gintHelpFileNbr As Integer

'-----

Public Sub OpenEmpDatabase()

'-----

Set gobjEmpDB = OpenDatabase(GetAppPath() & "EMPLOYEE.MDB")

End Sub

'-----

Public Sub CloseEmpDatabase()

'-----

gobjEmpDB.Close

Set gobjEmpDB = Nothing

End Sub

'-----

Public Sub CenterForm(pobjForm As Form)

'-----

With pobjForm

.Top = (Screen.Height - .Height) / 2

.Left = (Screen.Width - .Width) / 2

End With

End Sub

'-----

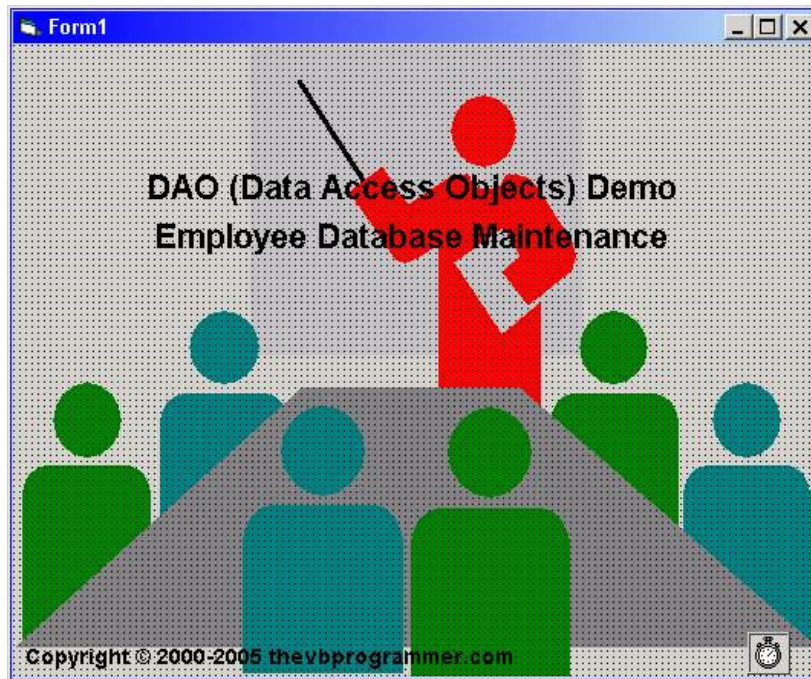
Public Function GetAppPath() As String

'-----

GetAppPath = IIf(Right$(App.Path, 1) = "\", App.Path, App.Path & "\")

End Function
```

Screen Splash (frmSplash):



Kode untuk frmSplash:

```
Option Explicit

'-----

Private Sub Form_Load()

'-----

CenterForm Me

End Sub

'-----

Private Sub tmrSplash_Timer()

'-----

tmrSplash.Enabled = False

frmMainMenu.Show

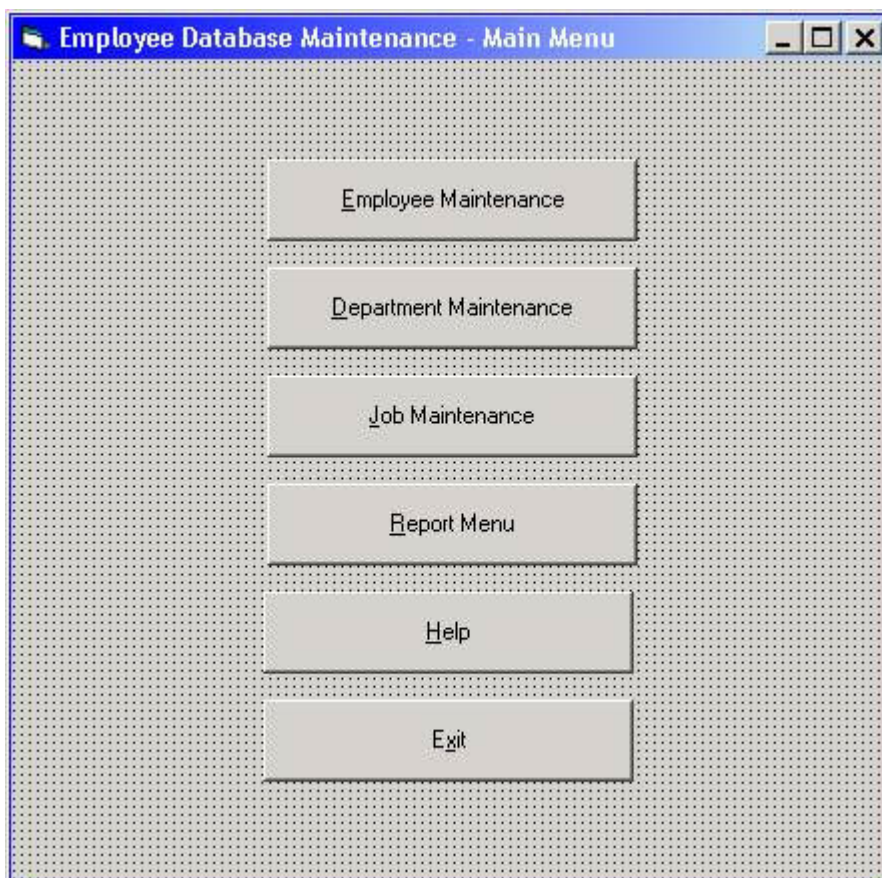
Unload Me

End Sub
```

Layar Menu Utama (frmMainMenu):

Menu utama, atau "**switchboard**" layar, bernama "**frmMainMenu**" ditampilkan di bawah. Tidak ada pengolahan database yang terkait dalam bentuk ini, baik. Ini berisi sebuah array dari enam tombol perintah, bernama **cmdMainMenuOpt**, diindeks 0 sampai 5.

Dalam melihat kode di **cmdMainMenuOpt_Click**, Anda dapat melihat bahwa untuk tombol perintah 0 sampai 3, bentuk yang sesuai ditampilkan. Untuk tombol perintah 4 (Bantuan), sebuah **ShowHelpForm** bernama sub disebut. **Sub ShowHelpForm** juga disebut bila pengguna menekan tombol F1 (lihat **event prosedur Form_KeyDown**). Ingatlah bahwa dalam rangka untuk acara Form_KeyDown untuk bekerja, Anda harus mengatur properti KeyPreview bentuk untuk Benar. Di sub ShowHelpForm, Anda melihat bahwa gintHelpFileNbr variabel publik di set ke 1. Alasan untuk ini akan dijelaskan lebih lanjut di bawah ini, ketika kita melihat bentuk Bantuan. Tombol Keluar (cmdMainMenuOpt (5)) hanya berakhir aplikasi.



Kode untuk frmMainMenu:

```
Option Explicit

'-----

Private Sub Form_Load()

'-----

CenterForm Me

End Sub

'-----

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)

'-----

If KeyCode = vbKeyF1 Then

ShowHelpForm

End If

End Sub

'-----

Private Sub cmdMainMenuOpt_Click(Index As Integer)

'-----

Select Case Index

Case 0

frmEmpMaint.Show vbModal

Case 1

frmDeptMaint.Show vbModal

Case 2

frmJobMaint.Show vbModal

Case 3

frmReportMenu.Show vbModal

Case 4

ShowHelpForm
```

```

Case 5

End

End Select

End Sub

'-----

Private Sub ShowHelpForm()

'-----

gintHelpFileNbr = 1

frmHelp.Show vbModal

End Sub

```

Layar Bantuan (frmHelp):

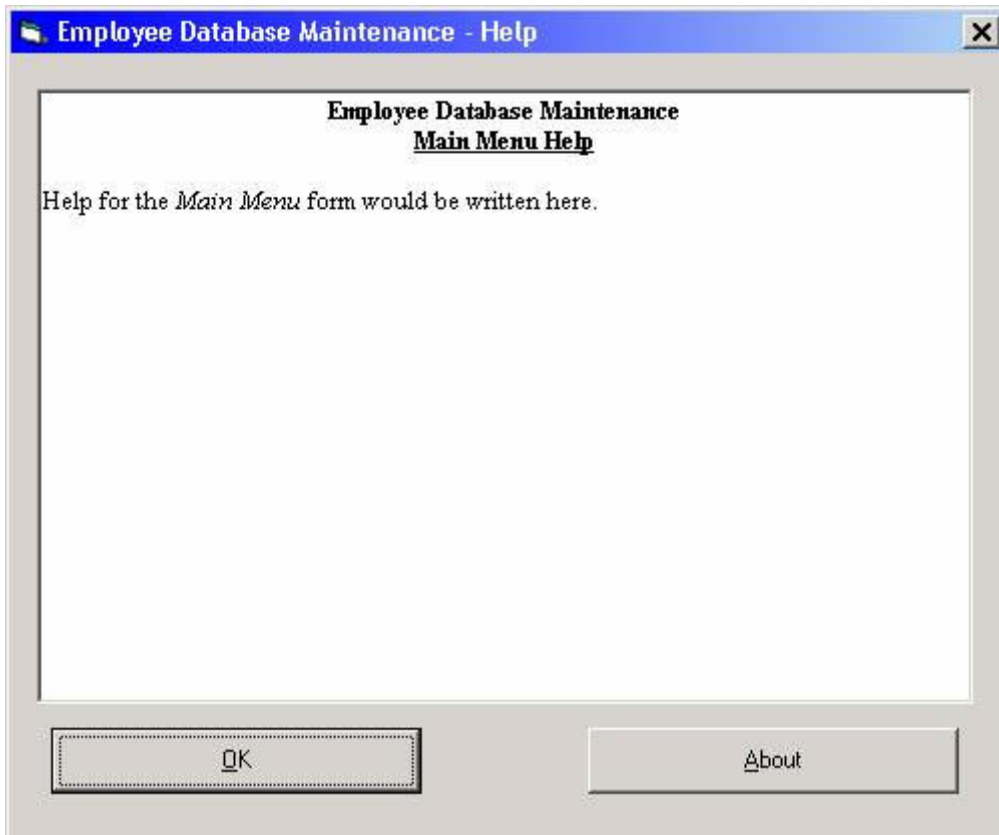
Layar Bantuan, bernama "frmHelp", menampilkan sebuah file bantuan yang sesuai dalam textbox kaya. Ada empat file bantuan untuk aplikasi bernama EDMHELP1.DOC, EDMHELP2.DOC, EDMHELP3.DOC, dan EDMHELP4.DOC. (. Meskipun ekstensi mereka DOC, file-file yang disimpan dalam format Rich Text) File-file ini berisi bantuan untuk bentuk Main Menu, bentuk Karyawan Pemeliharaan, bentuk Departemen Pemeliharaan, dan bentuk Ayub Pemeliharaan, masing-masing (sebenarnya, itulah yang mereka harus mengandung, mereka benar-benar hanya berisi satu atau dua kalimat, untuk tujuan aplikasi ini sampel).

Seperti yang Anda lihat dalam kode untuk membentuk Menu Utama, gintHelpFileNbr variabel publik ditetapkan sebelum menampilkan frmHelp. Bentuk frmHelp kemudian menggunakan nomor yang di acara Form_Load untuk menentukan file help untuk menampilkan dalam kotak teks kaya. Perhatikan bagaimana baris ini dalam acara Form_Load menghasilkan nama file sesuai:

```
strHelpFileName = GetAppPath & "EDMHELP" & gintHelpFileNbr & ".DOC"
```


Tombol Tentang menampilkan "Tentang" kotak pesan. Tombol OK membongkar bentuk, sehingga kembali kontrol ke bentuk menelepon.

Sebuah tampilan run-time serta kode untuk frmHelp ditunjukkan di bawah ini:



Kode untuk frmHelp:

```
Option Explicit

'-----

Private Sub Form_Load()

'-----

Dim strHelpFileName As String

CenterForm Me

strHelpFileName = GetAppPath & "EDMHELP" & gintHelpFileNbr & ".DOC"

rtbHelp.LoadFile strHelpFileName, rtfRTF

End Sub

'-----

Private Sub cmdOK_Click()

'-----

Unload Me

End Sub

'-----

Private Sub cmdAbout_Click()

'-----

MsgBox "DAO (Data Access Objects) Demo" & vbNewLine _

& "Employee Database Maintenance" & vbNewLine _

& "Copyright " & Chr$(169)

& " 2000-2005 thevbprogrammer.com", _ vbInformation, _ "About"

End Sub
```

Form-Form Data Entry (frmDeptMaint, frmJobMaint, frmEmpMaint)

Pada form entri data untuk aplikasi demo, pengguna dapat menelusuri melalui catatan dalam tabel (satu record per layar) menggunakan "Record Pertama", "Record Sebelumnya", "Next Record" dan "terakhir Record" tombol. Selain itu, pengguna dapat melakukan pencarian untuk melompat ke record tertentu berdasarkan kriteria masuk. Pengguna dapat menambah, memperbarui, atau menghapus merekam menggunakan "Record Tambah", "Record Update", dan "Hapus Record" tombol masing-masing.

Awalnya, pengguna dapat "melihat tapi tidak menyentuh" data, karena mereka melakukan berbagai bergerak atau operasi pencarian. Ketika pengguna inisiat add atau update, kontrol (umumnya textboxes) untuk bidang data serta "Simpan", "Undo" dan "Batal" tombol menjadi diaktifkan, semua tombol lain menjadi non-aktif.

Sebagai pengguna memasukkan atau memodifikasi data dalam kontrol, field-dengan-data field validasi dilakukan. Ini berarti bahwa pengguna tidak dapat pindah ke kontrol tertentu sampai semua kontrol sebelum lulus mengedit (misalnya, jika membentuk berisi nama depan, nama belakang, dan field alamat, pengguna tidak bisa bergerak ke bidang nama belakang sampai nama field pertama masuk dan pengguna tidak bisa bergerak ke field alamat sampai nama field terakhir yang dimasukkan).

Sebagaimana ditunjukkan di atas, ketika menambah atau update berlangsung, pengguna memiliki tiga pilihan yang tersedia melalui "Simpan", "Undo" dan "Batal" tombol:

Jika pengguna mengklik tombol "Terima", asalkan semua entri lulus validasi, catatan tersebut akan ditambahkan atau diperbarui dalam tabel database dan bentuk kontrol diatur kembali ke negara asal mereka (textboxes dan Terima, Undo, dan tombol Batal dinonaktifkan, semua tombol lainnya diaktifkan).

Jika pengguna mengklik "Undo" tombol sebagai gantinya, kontrol pada formulir adalah re-diisi dengan bidang yang sesuai dari catatan saat ini (jika melakukan update) atau kontrol pada form dihapus (jika melakukan menambahkan).

Bentuknya tetap dalam keadaan di mana pengguna dapat menambahkan atau memodifikasi data.

Jika pengguna mengklik "Batal", bidang catatan sebelumnya saat ini kembali ditugaskan untuk boks teks yang sesuai mereka di bentuk, dan, seperti dengan tombol "Simpan", bentuk kontrol diatur kembali ke negara asal mereka (textboxes dan Simpan, Undo, dan tombol Batal dinonaktifkan, semua tombol lainnya diaktifkan).

Perlu dicatat bahwa pendekatan entri data dilaksanakan oleh aplikasi demo seperti dijelaskan di atas adalah pendekatan ketat yang mencoba untuk mengontrol tindakan pengguna untuk tingkatan yang tertinggi. Pendekatan ini mungkin cocok dalam beberapa kasus tetapi tidak dalam yang lain â € "itu tergantung pada jenis aplikasi serta apa yang pengguna yang akrab dengan. Jika kita adalah untuk "melonggarkan" hal-hal sedikit, validasi field-field-oleh bisa dihilangkan dan validasi akan dilakukan hanya ketika rekaman adalah untuk diselamatkan (cara ini pengguna dapat bergerak pada formulir; melompat bidang kemudian kembali kepada mereka, dll). Jika kita adalah untuk melonggarkan hal-hal lebih lanjut, kita bisa menghilangkan "dua negara" pendekatan antara browsing dan memperbarui â € "dengan kata lain, memiliki semua bidang terbuka sepanjang waktu dan tidak membuat pengguna eksplisit klik tombol untuk memulai pembaruan. Terlepas dari pendekatan yang digunakan, aplikasi Anda akan perlu untuk mendeteksi ketika pengguna telah membuat perubahan dan menjamin integritas data.

Pelaksana Validasi Tiap Field

Dalam proses mengimplementasikan field-field dengan validasi seperti dijelaskan di atas, poin-poin berikut harus dipertimbangkan :

- Semua kontrol pada form Anda harus memiliki *tabindex* properti mereka disetel sehingga urutan tabbing benar.
- Properti *MaxLength* harus ditetapkan untuk semua textboxes mana yang sesuai.

- Bentuk variabel integer tingkatan form untuk menyimpan properti tabindex dari field yang sedang aktif harus dikenalkan atau di deklarasikan (dalam aplikasi sampel disebut `intCurrTabIndex`).
- Sebuah variabel Boolean tingkatan form yang menunjukkan apakah ada atau tidak kesalahan validasi terjadi harus dikenalkan atau di deklarasikan (dalam aplikasi sampel disebut `blnValidationError`).
- Sebuah rutin (Sub) yang berisi kode validasi untuk semua field enterable harus dikodekan (di aplikasi sampel Sub ini disebut `ValidateAllFields`).
- Logika untuk `GotFocus`, menekan tombol (jika perlu), dan event Perubahan harus dikodekan untuk setiap textbox, logika untuk acara Validasi harus kode untuk textbox terakhir.
-

Logika dasar kode di setiap peristiwa adalah sebagai berikut:

1. GotFocus:

- 1) Mengatur `intCurrTabIndex` ke properti `tabindex` textbox ini.
- 2) Panggil rutin `ValidateAllFields`.
- 3) Jika ada kesalahan validasi (`mblnValidationError = True`), `exit now` (melewatkan dua pernyataan berikutnya).
- 4) Mengatur properti `SelStart` textbox ini untuk 0.
- 5) Mengatur property `SelLength` textbox ini untuk panjang dari teks saat ini dalam textbox ini.

Catatan: Ketiga di awal pernyataan di atas dapat diabaikan dalam hal `GotFocus` dari textbox pertama yang divalidasi. Ketiga pernyataan tersebut pada dasarnya berkata kepada pengguna "Anda tidak bisa datang ke sini sampai Anda sudah berurusan dengan kesalahan sebelumnya." Atau " You can't come here until you've dealt with your previous errors." Laporan 4 dan 5 menyebabkan teks di textbox yang akan disorot ketika menerima fokus.

2. KeyPress:

Menempatkan kode opsional di sini untuk menyaring keluar karakter yang tidak diinginkan dimasukkan oleh pengguna (misalnya, jika bidang ini hanya berisi digit angka, Anda dapat menyaring apa pun). Anda juga dapat mengubah karakter yang dimasukkan oleh pengguna (konversi yang paling umum dilakukan di sini adalah untuk mengkonversi semua karakter huruf kecil dimasukkan oleh pengguna ke huruf besar).

3. Change:

Jika panjang teks dalam bidang ini adalah sama dengan properti MaxLength, maka mengatur fokus ke bidang enterable berikutnya. (Ini menyediakan fitur *auto-tab* untuk kolom form)

4. Validate :

Diperkenalkan di VB6, cara Validasi dapat digunakan dalam hubungannya dengan properti CausesValidation untuk mencegah kontrol dari kehilangan fokus sampai kriteria tertentu terpenuhi. Acara Validasi hanya terjadi ketika kontrol yang akan menerima fokus telah properti CausesValidation diatur ke Benar.

Acara Validasi cocok dalam skema masuk field-by-data field untuk bidang terakhir yang divalidasi. (Dalam aplikasi sampel, acara Validasi dikodekan untuk bidang entri data terakhir, dan tombol Simpan memiliki properti CausesValidation diatur ke Benar. Ketika pengguna tab keluar dari field terakhir dan ke tombol Simpan, yang Validasi kebakaran acara untuk field terakhir Jika peristiwa Validasi menganggap bahwa field belum melewati mengedit, parameter Batal dari acara Validasi diset ke True, menyebabkan fokus untuk tetap di field itu).

Catatan: Beberapa sumber daya menganjurkan penggunaan acara Validasi untuk semua bidang yang akan divalidasi, hal ini tidak dilakukan dalam aplikasi sampel karena acara Validasi akan api apakah Anda sedang bergerak mundur atau maju melalui ladang pada form â€ "niat dalam aplikasi sampel adalah untuk memvalidasi hanya ketika bergerak maju.

✚ Logika dasar (dalam pseudocode) untuk Sub ValidateAllFields adalah sebagai berikut:



5.