

Mengetahui Keseluruhan Informasi Sebuah File dengan Visual Basic 6.0

FIRSTI MANAH ASRI

firsty@linuxmail.org

Lisensi Dokumen:

Copyright © 2003-2006 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Sedikit saya ingin berbagi ilmu dengan rekan pembaca semua. Saya sedang mendalami pemrograman dengan Visual Basic, dan rasanya sangat senang sekali bila membagikan kode program yang telah saya pelajari kepada rekan pembaca semua ☺.

Ok's, langsung saja ke inti permasalahan, yakni membuat sebuah aplikasi untuk mengetahui keseluruhan informasi sebuah file. Aplikasi ini akan memberikan informasi kepada pengguna terhadap sebuah file yang berekstensi *.exe, *.dll, dan *.ocx. Hal ini bertujuan agar pengguna biasa bisa mengetahui informasi yang dimiliki oleh sebuah file tersebut sehingga dapat digunakan sebagai landasan dan pendukung sebuah teori terhadap sebuah file.

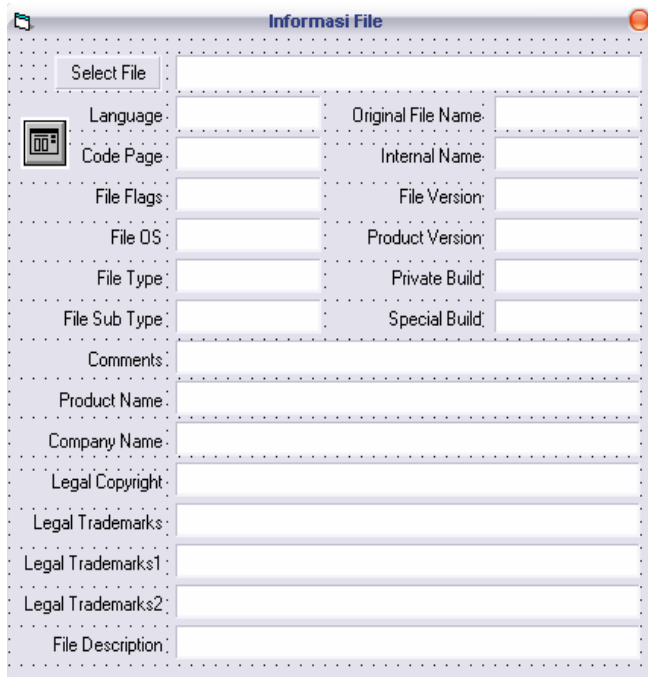
Beberapa informasi yang bisa ditampilkan oleh aplikasi ini adalah :

- Language
- Original File Name
- Code Page
- Internal Name
- File flags
- File Version
- File OS
- Product Version
- File Type
- Private Build
- File Sub Type
- Special Build
- Comments
- Product Name
- Company Name
- Legal Copyright
- Legal Trademarks

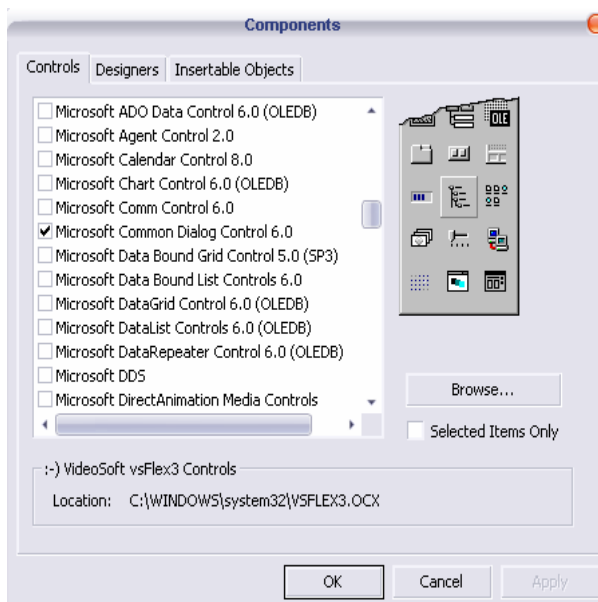
- Legal Trademarks1
- Legal Trademarks2
- File Description

Langkah-langkah membuat aplikasi ini adalah :

Sebelumnya buka Visual Basic 6.0 dan buat sebuah *project* dengan sebuah form standar. Letakkan *common dialog*, *button*, *label* dan *textbox* seperti gambar berikut :



Untuk menampilkan *common dialog*, klik menu **Project > Components** (atau tekan Ctrl + t). Pastikan *Microsoft Common Dialog Control 6.0* dicentang.



Kemudian masuk ke jendela kode dan ketikkan listing program berikut :

```
Option Explicit
Private Sub cmdSelectFile_Click()
Dim strFile As String
Dim strFilter As String

    '
    ' Clear fields.
    '
    txtFileName.Text = ""
    Call pClearLabels

    strFilter = _
    "Executable files (*.exe;*.dll;*.ocx)|*.exe;*.dll;*.ocx|" & _
    "Drivers (*.sys;*.drv;*.fnt)|*.sys;*.drv;*.fnt|" & _
    "All files (*.*)|*.*"

    '
    ' Select the file whose version information
    ' you want to view.
    '
    If Not fGetFile(strFile, strFilter, "Select a File") Then
        Exit Sub
    End If

    '
    txtFileName.Text = "VBE6.exe (module in memory)"
    '
    Call GetVersionInfo("VBE6", False)

    txtFileName.Text = strFile
    Call GetVersionInfo(txtFileName.Text, True)
End Sub
Private Sub GetVersionInfo(strName As String, _
    Optional fIsFileName As Boolean = True)
Dim vi As VersionInfo

    On Error GoTo HandleErrors

    Set vi = New VersionInfo

    With vi
        If fIsFileName Then
            .FileName = strName
        Else
            .ModuleName = strName
        End If

        lblInfo(0).Caption = .Language
        lblInfo(1).Caption = .CodePage
        lblInfo(2).Caption = .Comments
        lblInfo(3).Caption = .FileFlags
        lblInfo(4).Caption = .FileOS
        lblInfo(5).Caption = .FileType
        lblInfo(6).Caption = .FileSubType
        lblInfo(7).Caption = .ProductName
        lblInfo(8).Caption = .OriginalFileName
    End With
End Sub
```

```
        lblInfo(9).Caption = .InternalName
        lblInfo(10).Caption = .CompanyName
        lblInfo(11).Caption = .LegalCopyright
        lblInfo(12).Caption = .LegalTrademarks
        lblInfo(13).Caption = .LegalTrademarks1
        lblInfo(14).Caption = .LegalTrademarks2
        lblInfo(15).Caption = .FileDescription
        lblInfo(16).Caption = .FileVersion
        lblInfo(17).Caption = .ProductVersion
        lblInfo(18).Caption = .PrivateBuild
        lblInfo(19).Caption = .SpecialBuild
    End With

ExitHere:
    Set vi = Nothing
    Exit Sub

HandleErrors:
    MsgBox "Error: " & Err.Description & " (" & Err.Number & ")"
    Resume ExitHere
End Sub

Public Function fGetFile(strPath As String, strFilter As String, _
    strTitle As String) As Boolean
    '
    ' Let the user point to the folder containing the
    ' template .dbf files.
    '
    On Error GoTo ErrorHandler
    fGetFile = True
    strPath = ""
    '
    ' Display the file open dialog.
    '
    With CommonDialog
        .FileName = ""
        .CancelError = True
        .DialogTitle = "Select the " & strTitle & " file"
        .DefaultExt = ""
        .Flags = cdIOFNFileMustExist Or cdIOFNNoLongNames Or
cdIOFNPathMustExist Or cdIOFNHideReadOnly
        .Filter = strFilter & " |" & strFilter
        .FilterIndex = 1
        .ShowOpen
        strPath = .FileName
    End With
    Exit Function

ErrorHandler:
    fGetFile = False
End Function

Private Sub pClearLabels()
    Dim lbl As Label

    For Each lbl In lblInfo
        lbl.Caption = ""
    Next

End Sub
```

Langkah selanjutnya adalah membuat sebuah *module* yang akan digunakan nantinya pada pengembangan aplikasi ini. Pada *module* yang akan dibuat, akan terdapat variabel yang bersifat global atau dapat diakses dari form utama pada program aplikasi dan beberapa *procedure* yang juga bisa diakses dari form utama.

Untuk menambahkan module, klik **Project > Add Class Module**, atau klik kanan pada bagian project, klik **Add > Class Module**. Setelah itu akan tampil kotak dialog **Add Class Module** tab **New**. Klik **Open**.

Ketikkan listing program berikut pada *class module* yang telah dibuat sebelumnya:

```
Option Explicit
Private Const adhClassName = "VersionInfo"
Private Const adhErrBase = vbObjectError + 19560516
Private Const adhErrNoModuleHandle = adhErrBase + 1
Private Const adhErrNoModule = adhErrBase + 2
Private Const adhErrNoFileName = adhErrBase + 3
Private Const adhErrNoVersionInfoSize = adhErrBase + 4
Private Const adhErrNoVersionInfo = adhErrBase + 5
Private Const adhErrNoLangCPInfo = adhErrBase + 6
Private Const adhNoFixedFileInfo = adhErrBase + 7
Private Const adhErrUnknown = adhErrBase + 8

Public Enum adhLanguageValue
    ' lvArabic = 1025                'Arabic
    ' lvBulgarian = 1026            'Bulgarian
    ' lvCatalan = 1027              'Catalan
    ' lvTraditionalChinese = 1028   'Traditional Chinese
    ' lvCzech = 1029                'Czech
    ' lvDanish = 1030               'Danish
    lvGerman = 1031                 'German
    ' lvGreek = 1032                'Greek
    lvUSEnglish = 1033              'U.S. English
    ' lvCastilianSpanish = 1034      'Castilian Spanish
    ' lvFinnish = 1035              'Finnish
    lvFrench = 1036                 'French
    ' lvHebrew = 1037               'Hebrew
    ' lvHungarian = 1038            'Hungarian
    ' lvIcelandic = 1039            'Icelandic
    lvItalian = 1040                'Italian
    ' lvJapanese = 1041             'Japanese
    ' lvKorean = 1042               'Korean
    ' lvDutch = 1043                'Dutch
    ' lvNorwegianBokmål = 1044       'Norwegian - Bokmål
    ' lvPolish = 1045               'Polish
    ' lvBrazilianPortuguese = 1046   'Brazilian Portuguese
    ' lvRhaetoRomanic = 1047         'Rhaeto-Romanic
    ' lvRomanian = 1048              'Romanian
    ' lvRussian = 1049              'Russian
    ' lvCroatoSerbian = 1050         'Croato-Serbian (Latin)
    ' lvSlovak = 1051               'Slovak
    ' lvAlbanian = 1052             'Albanian
    ' lvSwedish = 1053              'Swedish
    ' lvThai = 1054                 'Thai
    ' lvTurkish = 1055              'Turkish
    ' lvUrdu = 1056                 'Urdu
    ' lvBahasa = 1057               'Bahasa
    ' lvSimplifiedChinese = 2052    'Simplified Chinese
```

```

    ' lvSwissGerman = 2055
lvUKEnglish = 2057
lvMexicanSpanish = 2058
    ' lvBelgianFrench = 2060
    ' lvSwissItalian = 2064
    ' lvBelgianDutch = 2067
    ' lvNorwegianNynorsk = 2068
    ' lvPortuguese = 2070
    ' lvSerboCroatian = 2074
    ' lvCanadianFrench = 3084
    ' lvSwissFrench = 4108
End Enum

Private Type VS_FIXEDFILEINFO
    dwSignature As Long
    dwStrucVersionl As Integer
    dwStrucVersionh As Integer
    dwFileVersionMSl As Integer
    dwFileVersionMSh As Integer
    dwFileVersionLSl As Integer
    dwFileVersionLSh As Integer
    dwProductVersionMSl As Integer
    dwProductVersionMSh As Integer
    dwProductVersionLSl As Integer
    dwProductVersionLSh As Integer
    dwFileFlagsMask As Long
    dwFileFlags As Long
VFF_ PRERELEASE
    dwFileOS As Long
    dwFileType As Long
    dwFileSubtype As Long
    dwFileDateMS As Long
    dwFileDateLS As Long
End Type

Public Enum adhFileFlags
    VS_FF_DEBUG = &H1&
    VS_FF_PRERELEASE = &H2&
    VS_FF_PATCHED = &H4&
    VS_FF_PRIVATEBUILD = &H8&
    VS_FF_SPECIALBUILD = &H20&
End Enum

Public Enum adhTargetOS
    VOS_UNKNOWN = &H0&
    VOS__BASE = &H0&
    VOS__WINDOWS16 = &H1&
    VOS__PM16 = &H2&
    VOS__PM32 = &H3&
    VOS__WINDOWS32 = &H4&
    VOS_DOS = &H10000
    VOS_OS216 = &H20000
    VOS_OS232 = &H30000
    VOS_NT = &H40000

    ' Combinations of previous values.
    VOS_DOS_WINDOWS16 = &H10001
    VOS_DOS_WINDOWS32 = &H10004
    VOS_NT_WINDOWS32 = &H40004

```

```
VOS_OS216_PM16 = &H20002
VOS_OS232_PM32 = &H30003
End Enum

Public Enum adhFileType
    VFT_UNKNOWN = &H0&
    VFT_APP = &H1&
    VFT_DLL = &H2&
    VFT_DRV = &H3&
    VFT_FONT = &H4&
    VFT_VXD = &H5&
    VFT_STATIC_LIB = &H7&
End Enum

Public Enum adhFileSubType
    VFT2_UNKNOWN = &H0&
    VFT2_DRV_COMM = &HA&
    VFT2_DRV_PRINTER = &H1&
    VFT2_DRV_KEYBOARD = &H2&
    VFT2_DRV_LANGUAGE = &H3&
    VFT2_DRV_DISPLAY = &H4&
    VFT2_DRV_MOUSE = &H5&
    VFT2_DRV_NETWORK = &H6&
    VFT2_DRV_SYSTEM = &H7&
    VFT2_DRV_INSTALLABLE = &H8&
    VFT2_DRV_SOUND = &H9&
    VFT2_DRV_INPUTMETHOD = &HB&

    VFT2_FONT_RASTER = &H1&
    VFT2_FONT_TRUETYPE = &H3&
    VFT2_FONT_VECTOR = &H2&
End Enum

'
' API Declarations
'

Private Declare Function GetModuleFileName _
    Lib "kernel32" Alias "GetModuleFileNameA" _
    (ByVal hModule As Long, ByVal lpFileName As String, _
    ByVal nSize As Long) As Long

Private Declare Function GetModuleHandle _
    Lib "kernel32" Alias "GetModuleHandleA" _
    (ByVal lpModuleName As String) As Long

Private Declare Function GetFileVersionInfoSize _
    Lib "version.dll" Alias "GetFileVersionInfoSizeA" _
    (ByVal lptstrFilename As String, lpdwHandle As Long) As Long

Private Declare Function GetFileVersionInfo _
    Lib "version.dll" Alias "GetFileVersionInfoA" _
    (ByVal lptstrFilename As String, ByVal dwHandle As Long, _
    ByVal dwLen As Long, lpData As Any) As Long

Private Declare Function VerQueryValue _
    Lib "version.dll" Alias "VerQueryValueA" _
    (pBlock As Any, ByVal lpSubBlock As String, _
    lpplpBuffer As Any, puLen As Long) As Long
```

```
Private Declare Function lstrcpy _
    Lib "kernel32" Alias "lstrcpyA" _
    (ByVal lpString1 As String, ByVal lpString2 As Long) As Long

Private Declare Sub RtlMoveMemory _
    Lib "kernel32" _
    (Destination As Any, ByVal Source As Any, ByVal Length As Long)

'
' Private variables
'
Private mstrModuleName As String
Private mstrFileName As String

Private mabytBuffer() As Byte

Private mstrLangCP As String

Private mffi As VS_FIXEDFILEINFO

Public Property Let FileName(Value As String)

    mstrFileName = Value

    Call GetLanguageInfo
End Property

Public Property Get FileName() As String

    FileName = mstrFileName

End Property

Public Property Let ModuleName(Value As String)

    mstrModuleName = Value

    Call GetLanguageInfo
End Property

Public Property Get ModuleName() As String

    ModuleName = mstrModuleName

End Property

Public Property Get Language() As Long

    If Len(mstrLangCP) = 8 Then
        Language = Val("&H" & Left$(mstrLangCP, 4))
    End If
End Property

Public Property Get CodePage() As Long

    If Len(mstrLangCP) = 8 Then
        CodePage = Val("&H" & Right$(mstrLangCP, 4))
    End If
```



```
End Property

Public Property Get CompanyName() As String
    CompanyName = GetValue("CompanyName")
End Property

Public Property Get FileDescription() As String
    FileDescription = GetValue("FileDescription")
End Property

Public Property Get FileVersion() As String
    FileVersion = GetValue("FileVersion")
End Property

Public Property Get InternalName() As String
    InternalName = GetValue("InternalName")
End Property

Public Property Get LegalCopyright() As String
    LegalCopyright = GetValue("LegalCopyright")
End Property

Public Property Get OriginalFileName() As String
    OriginalFileName = GetValue("OriginalFileName")
End Property

Public Property Get ProductName() As String
    ProductName = GetValue("ProductName")
End Property

Public Property Get ProductVersion() As String
    ProductVersion = GetValue("ProductVersion")
End Property

Public Property Get LegalTrademarks() As String
    LegalTrademarks = GetValue("LegalTrademarks")
End Property

Public Property Get LegalTrademarks1() As String
```

```
        LegalTrademarks1 = GetValue("LegalTrademarks1")
End Property

Public Property Get LegalTrademarks2() As String
    LegalTrademarks2 = GetValue("LegalTrademarks2")
End Property

Public Property Get Comments() As String
    Comments = GetValue("Comments")
End Property

Public Property Get PrivateBuild() As String
    PrivateBuild = GetValue("PrivateBuild")
End Property

Public Property Get SpecialBuild() As String
    SpecialBuild = GetValue("SpecialBuild")
End Property

Public Property Get FileFlagsRaw() As adhFileFlags
    FileFlagsRaw = mffi.dwFileFlags And mffi.dwFileFlagsMask
End Property

Public Property Get FileFlags() As String
Dim strOut As String
Dim lngTemp As adhFileFlags

    lngTemp = mffi.dwFileFlags And mffi.dwFileFlagsMask

    If lngTemp And VS_FF_DEBUG Then strOut = strOut & "Debug "
    If lngTemp And VS_FF_PATCHED Then strOut = strOut & "Patched "
    If lngTemp And VS_FF_PRERELEASE Then strOut = strOut & "Pre-
release "
    If lngTemp And VS_FF_PRIVATEBUILD Then strOut = strOut & "Private
"
    If lngTemp And VS_FF_SPECIALBUILD Then strOut = strOut & "Special
"

    FileFlags = strOut
End Property

Public Property Get FileOSRaw() As adhTargetOS
    FileOSRaw = mffi.dwFileOS
End Property

Public Property Get FileOS() As String
```

```
Dim lngTemp As Long
Dim strOut As String
lngTemp = mffi.dwFileOS

If lngTemp And VOS__WINDOWS16 Then
    strOut = "Win16 API"
ElseIf lngTemp And VOS__PM16 Then
    strOut = "PM16 API"
ElseIf lngTemp And VOS__PM32 Then
    strOut = "PM32 API"
ElseIf lngTemp And VOS__WINDOWS32 Then
    strOut = "Win32 API"
Else
    strOut = "Unknown API"
End If

If lngTemp And VOS_DOS Then
    strOut = strOut & " on DOS"
ElseIf lngTemp And VOS_OS216 Then
    strOut = strOut & " on OS2/16"
ElseIf lngTemp And VOS_OS232 Then
    strOut = strOut & " on OS2/32"
ElseIf lngTemp = VOS_NT Then
    strOut = strOut & " on Windows NT"
End If

FileOS = strOut
End Property

Public Property Get FileTypeRaw() As adhFileType

    FileTypeRaw = mffi.dwFileType

End Property

Public Property Get FileType() As String
Dim strOut As String

    Select Case mffi.dwFileType
        Case VFT_APP
            strOut = "Application"
        Case VFT_DLL
            strOut = "DLL"
        Case VFT_DRV
            strOut = "Driver"
        Case VFT_FONT
            strOut = "Font"
        Case VFT_STATIC_LIB
            strOut = "Static-link library"
        Case VFT_UNKNOWN
            strOut = "Unknown"
        Case VFT_VXD
            strOut = "VXD"
    End Select
    FileType = strOut
End Property

Public Property Get FileSubTypeRaw() As adhFileSubType
```

```
FileSubTypeRaw = mffi.dwFileSubtype

End Property

Public Property Get FileSubType() As String
Dim strOut As String

    strOut = "Unknown"

    Select Case mffi.dwFileType
        Case VFT_APP
        Case VFT_DLL
        Case VFT_DRV
            Select Case mffi.dwFileSubtype
                Case VFT2_DRV_COMM
                    strOut = "Communications"
                Case VFT2_DRV_DISPLAY
                    strOut = "Display"
                Case VFT2_DRV_INPUTMETHOD
                    strOut = "Input Method"
                Case VFT2_DRV_INSTALLABLE
                    strOut = "Installable"
                Case VFT2_DRV_KEYBOARD
                    strOut = "Keyboard"
                Case VFT2_DRV_LANGUAGE
                    strOut = "Language"
                Case VFT2_DRV_MOUSE
                    strOut = "Mouse"
                Case VFT2_DRV_NETWORK
                    strOut = "Network"
                Case VFT2_DRV_PRINTER
                    strOut = "Printer"
                Case VFT2_DRV_SOUND
                    strOut = "Sound"
                Case VFT2_DRV_SYSTEM
                    strOut = "System"
                Case VFT2_UNKNOWN
                    strOut = "Unknown"
            End Select
        Case VFT_FONT
            Select Case mffi.dwFileSubtype
                Case VFT2_FONT_RASTER
                    strOut = "Raster"
                Case VFT2_FONT_TRUETYPE
                    strOut = "TrueType"
                Case VFT2_FONT_VECTOR
                    strOut = "Vector"
                Case VFT2_UNKNOWN
                    strOut = "Unknown"
            End Select
        Case VFT_STATIC_LIB
        Case VFT_UNKNOWN
        Case VFT_VXD

            strOut = mffi.dwFileSubtype
    End Select
    FileSubType = strOut
End Property
```

```
Private Function GetValue(strItem As String) As String
Dim strTemp          As String
Dim lngVerPointer As Long
Dim lngBufferLen  As Long
Dim strResult      As String

    On Error GoTo HandleErrors

    strTemp = "\StringFileInfo\" & mstrLangCP & "\" & strItem
    If VerQueryValue(mabytBuffer(0), strTemp, _
        lngVerPointer, lngBufferLen) <> 0 Then
        strResult = Space$(lngBufferLen - 1)
        Call lstrcpy(strResult, lngVerPointer)
        strResult = adhTrimNull(strResult)
    End If

ExitHere:
    GetValue = strResult
    Exit Function

HandleErrors:
    strResult = vbNullString
    Resume ExitHere
End Function

Private Sub GetLanguageInfo()
Dim hInst          As Long
Dim lngValue       As Long
Dim lngBufferLen  As Long
Dim lngVerPointer As Long
Dim abyTemp()      As Byte
    If Len(mstrModuleName) > 0 Then
        hInst = GetModuleHandle(mstrModuleName)

        If hInst = 0 Then
            Err.Raise adhcErrNoModuleHandle, _
                adhcClassName, "Unable to retrieve module handle."
        End If

        mstrFileName = Space(260)
        If GetModuleFileName(hInst, mstrFileName, Len(mstrFileName)) =
0 Then

            Err.Raise adhcErrNoModule, _
                adhcClassName, "Unable to retrieve module file name."
        End If
    End If
    If Len(mstrFileName) = 0 Then
        Err.Raise adhcErrNoFileName, adhcClassName, _
            "Either you haven't specified a valid file name, or the
module you specified can't be associated with a file."
    End If

    lngBufferLen = GetFileVersionInfoSize(mstrFileName, lngValue)
    If lngBufferLen = 0 Then
        Err.Raise adhcErrNoVersionInfoSize, adhcClassName, _
            "Unable to retrieve the version info size. Perhaps the module
name or file name is invalid."
    End If
```

```
ReDim mabytBuffer(0 To lngBufferLen - 1)
If GetFileVersionInfo(mstrFileName, 0, lngBufferLen,
mabytBuffer(0)) = 0 Then
    Err.Raise adhcErrNoVersionInfo, adhcClassName, _
        "Unable to retrieve version information."
End If

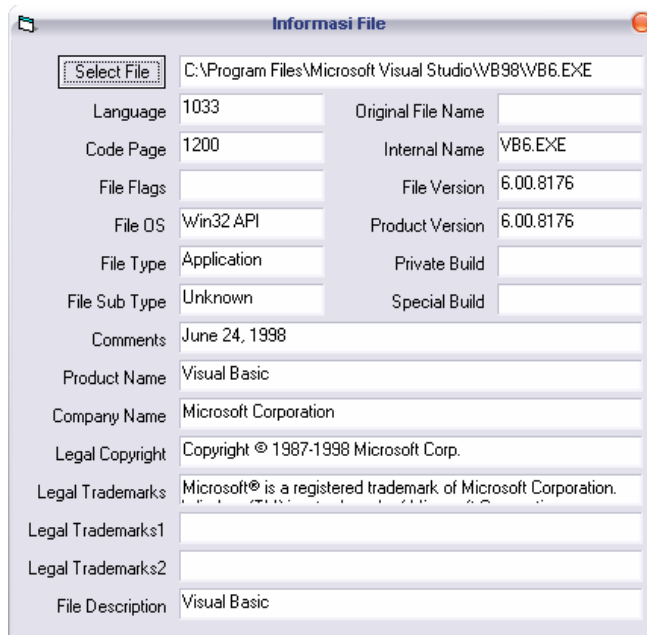
If VerQueryValue(mabytBuffer(0), "\VarFileInfo\Translation", _
    lngVerPointer, lngBufferLen) = 0 Then
    Err.Raise adhcErrNoLangCPInfo, adhcClassName, _
        "Unable to retrieve language and code page information."
End If

ReDim abyTemp(0 To lngBufferLen - 1)
Call RtlMoveMemory(abyTemp(0), lngVerPointer, lngBufferLen)
mstrLangCP = _
    ZeroPad(Hex$(abyTemp(1)), 2) & _
    ZeroPad(Hex$(abyTemp(0)), 2) & _
    ZeroPad(Hex$(abyTemp(3)), 2) & _
    ZeroPad(Hex$(abyTemp(2)), 2)

If mstrLangCP = "00000000" Then
    mstrLangCP = "040904E4"
End If

If VerQueryValue(mabytBuffer(0), "\", _
    lngVerPointer, lngBufferLen) = 0 Then
    Err.Raise adhcNoFixedFileInfo, adhcClassName, _
        "Unable to retrieve fixed file version information."
End If
Call RtlMoveMemory(mffi, lngVerPointer, lngBufferLen)
End Sub
Private Function adhTrimNull(strVal As String) As String
Dim intPos As Integer
intPos = InStr(1, strVal, vbNullChar)
Select Case intPos
    Case Is > 1
        adhTrimNull = Left$(strVal, intPos - 1)
    Case 0
        adhTrimNull = strVal
    Case 1
        adhTrimNull = vbNullString
End Select
End Function
Private Function ZeroPad(strValue As String, intLen As String) As
String
    ZeroPad = Right$(String(intLen, "0") & strValue, intLen)
End Function
```

Langkah terakhir adalah dengan menjalankan program tersebut (klik F5), dan hasil akhirnya adalah sebagai berikut :



Selamat Mencoba!!! ☺ ☺

Biografi Penulis



Firsti Manah Asri. Dilahirkan di Batang pada tanggal 16 Nopember 1985. Menamatkan SMU pada tahun 2003, dan kemudian melanjutkan studinya di Institut Sains dan Teknologi AKPRIND (ISTA) Yogyakarta. Sekarang sedang duduk di bangku perkuliahan dan mengambil jurusan Teknik Informatika. Mulai mengenal komputer pada umur 15 tahun dan pada saat itu tertarik pada Sistem Operasi Windows. Untuk memperdalam ilmu yang yang diperoleh lewat bangku perkuliahan, Penulis menjadi Assisten pada sebuah Laboratorium Komputer di kampusnya. Saat ini sedang mendalami pemrograman dengan Visual Basic, jaringan komputer (khususnya Wireless), serta database.

Informasi lebih lanjut mengenai penulis ini bisa didapat melalui :

firsty@linuxmail.org