

**PEMBUATAN ANTIVIRUS YANG MENDUKUNG
KARAKTER-KARAKTER *UNICODE* MENGGUNAKAN
VISUAL BASIC 6.0**

Yosafat Kristianto / 0827029

Jurusan Sistem Komputer, Fakultas Teknik, Universitas Kristen Maranatha
Jalan Prof. Drg. Surya Sumantri 65
Bandung 40164, Indonesia

ABSTRAK

Sebagai pengguna komputer tentunya pernah mengalami masalah yang di akibatkan oleh *malware* seperti virus. Penggunaan antivirus merupakan cara untuk melawan virus tersebut. Virus sekarang telah berkembang dengan teknologi dan serangan yang lebih mematikan. Begitu juga dengan antivirusnya yang ikut berkembang untuk melawan virus-virus mematikan. Sekarang virus telah mampu menggunakan karakter *Unicode* agar tidak terdeteksi oleh antivirus, khususnya antivirus lokal. Masih banyak antivirus lokal yang belum mendukung pendeteksian virus yang menggunakan karakter *Unicode*. Oleh karena itu dalam tugas akhir ini dibuatlah antivirus lokal yang dapat mendeteksi virus yang menggunakan karakter-karakter *Unicode*.

***Making of Antivirus Supporting Unicode Characters
Using Visual Basic 6.0***

Yosafat Kristianto / 0827029

Jurusan Sistem Komputer, Fakultas Teknik, Universitas Kristen Maranatha
Jalan Prof. Drg. Surya Sumantri 65
Bandung 40164, Indonesia

ABSTRACT

A computer user is likely to encounter a problem caused by malwares such as virus. Antivirus is a way to fight viruses. Today's virus has evolved with technology and much more deadly attack. Likewise, antiviruses evolved to fight harmful viruses. Nowadays, viruses are use Unicode characters to avoid antivirus scanner, especially local antivirus. Most of the local antivirus have no support for Unicode enabled viruses. This project's objective is to create a local antivirus capable of detecting Unicode enabled viruses.

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	vii
DAFTAR TABEL	ix
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	2
1.3 Tujuan	2
1.4 Pembatasan Masalah.....	2
1.5 Sistematika Penulisan	2
BAB II LANDASAN TEORI	4
2.1 Virus	4
2.2 Antivirus	8
2.3 Karakter <i>Unicode</i>	10
2.4 <i>Checksum</i> dan <i>Hash</i>	14
2.6 Algoritma MD5	16
2.6 Microsoft Visual Basic	19
BAB III PERANCANGAN DAN IMPLEMENTASI	22
3.1 Deskripsi Singkat	22

3.2 Perancangan Tampilan	28
3.3 Penambahan komponen dalam Microsoft Visual Basic 6.0.....	30
3.4 Desain <i>form</i> dalam Visual Basic	31
BAB IV DATA PENGAMATAN	35
4.1 Antivirus Mandalawangi	35
4.2 Percobaan Algoritma MD5	37
4.3 Percobaan memindai <i>folder</i> yang berisi virus	39
4.4 Percobaan pemindaian yang berisi virus menggunakan	
antivirus lain	41
BAB V KESIMPULAN DAN SARAN	47
5.1 Kesimpulan.....	47
5.2 Saran.....	47
DAFTAR PUSTAKA	48
Lampiran <i>Source Code</i>	A-1

DAFTAR GAMBAR

Gambar 2.1 contoh <i>hashing</i> beberapa buah pesan dengan panjang berbeda.....	15
Gambar 2.2 Pembuatan <i>message digest</i> dengan algoritma MD5.....	16
Gambar 2.3 Pengolahan blok 512 <i>bit</i> (Proses H_{MD5}).....	19
Gambar 2.4 Jendela Kode.....	21
Gambar 2.5 Jendela Kode.....	21
Gambar 3.1 <i>Flowchart</i> perancangan antivirus.....	22
Gambar 3.1 <i>Flowchart</i> perancangan antivirus (lanjutan).....	23
Gambar 3.2 <i>Flowchart</i> MD5.....	26
Gambar 3.3 Rancangan Tampilan <i>Form</i> Utama.....	28
Gambar 3.4 Rancangan <i>Form</i> Pilihan.....	29
Gambar 3.5 Desain <i>form</i> utama.....	31
Gambar 3.6 Desain <i>form option</i>	33
Gambar 4.1 Form utama Antivirus Mandalawangi.....	35
Gambar 4.2 Form pilihan pada Antivirus Mandalawangi.....	36
Gambar 4.3 Nilai <i>hash</i> “test” menggunakan Visual Basic.....	38
Gambar 4.4 <i>Folder</i> yang berisi virus.....	39
Gambar 4.5 Hasil virus yang di deteksi setelah di pindai.....	40
Gambar 4.6 Hasil pemindaian menggunakan MSE.....	41
Gambar 4.7 Hasil pemindaian menggunakan MSE.....	42
Gambar 4.8 Hasil pemindaian menggunakan MSE.....	43
Gambar 4.9 Hasil pemindaian menggunakan MSE.....	44

Gambar 4.10 Hasil pemindaian menggunakan MSE.....	45
Gambar 4.11 Hasil pemindaian menggunakan Smadv.....	46

DAFTAR TABEL

Tabel 2.1 Perbedaan <i>worm</i> dan virus.....	8
Tabel 3.1 Keterangan <i>form</i>	32
Tabel 3.2 Keterangan <i>form</i> (Lanjutan).....	33
Tabel 3.3 Keterangan <i>form option</i>	34
Tabel 4.1 Tabel Nilai <i>Hash</i>	37
Tabel 4.1 Tabel Nilai <i>Hash</i> (lanjutan).....	38
Tabel 4.2 Uji coba <i>folder</i> berisi virus dengan tiga antivirus.....	47

LAMPIRAN

Lampiran source code


```
Private Sub cmdCekMD5_Click()
```

```
frmMD5.Show 1, Me
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
lstResult.AddItem "Selamat Datang di Antivirus Mandalawangi!"
```

```
lstResult.AddItem "Copyright © 2010 - 2011, Mandalawangi Soft"
```

```
lstScan.AddItem Environ$("windir") & "\*.*)"
```

```
BERHENTI = True
```

```
BacaDatabase App.path & "\database.db"
```

```
End Sub
```

```
Private Sub cmdAdd_Click()
```

```
UniDialog1.FolderMessage = "Select a path :"
```

```
UniDialog1.ShowFolder
```

```
End Sub
```

```
Private Sub cmdRemove_Click()
```

```
On Error Resume Next
```

```
Static count As Integer
```

```
For count = 1 To lstScan.ListCount
```

```
If lstScan.Selected(count - 1) = True Then lstScan.RemoveItem (count - 1)
```

```
Next
```

```
End Sub
```

```
Private Sub Command1_Click()

Static count As Byte

If lstScan.ListCount = 0 Then Exit Sub

If Command1.Caption = "Mulai &Pindai" Then

    BERHENTI = False

    PosisiScan (True)

    Command1.Caption = "Batal &Pindai"

    lstResult.Clear

    For count = 0 To lstScan.ListCount - 1

        If BERHENTI = True Then Exit For

        lstResult.AddItem "[Scanning File ...]"

        lstResult.AddItem " "

        lstResult.AddItem " "

        ScanFolder RemoveFromRight(lstScan.List(count), 4), lblFile, lstResult

    Next

    Command1.Caption = "Mulai &Pindai"

    PosisiScan (False)

    BERHENTI = True

    MsgBox "Scan is finished !", vbInformation, "MY Antivirus"

Else

    BERHENTI = True

    Command1.Caption = "Mulai &Pindai"

    PosisiScan (False)

End If
```

End Sub

Private Sub Command2_click()

frmOption.Show 1, Me

End Sub

Private Sub Command5_click()

Unload Me

End Sub

Private Sub Form_Resize()

On Error Resume Next

Me.Height = 8760

Me.Width = 9700

End Sub

Private Sub Form_Unload(Cancel As Integer)

If BERHENTI = False Then Cancel = 1

End Sub

Private Sub UniDialog1_FolderSelect(ByVal path As String)

If path <> "" Then

lstScan.AddItem path & "*.*)"

End If

End Sub

```
Private Function AddSlash(sPath As String) As String
```

```
    If Right(sPath, 1) = "\" Then
```

```
        AddSlash = sPath
```

```
    Else
```

```
        AddSlash = sPath & "\"
```

```
    End If
```

```
End Function
```

```
Private Function PosisiScan(TF As Boolean)
```

```
    If TF = True Then
```

```
        Command2.Enabled = False
```

```
        Command5.Enabled = False
```

```
        frScan.Enabled = False
```

```
        frResult.Enabled = False
```

```
    Else
```

```
        Command2.Enabled = True
```

```
        Command5.Enabled = True
```

```
        frScan.Enabled = True
```

```
        frResult.Enabled = True
```

```
    End If
```

```
End Function
```

```
Private Function RemoveFromRight(sTemp As String, iRight As Integer) As String
```

```
RemoveFromRight = Left(sTemp, Len(sTemp) - iRight)
```

```
End Function
```

```
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA"  
(ByVal hWnd As Long, ByVal wParam As Long, ByVal lParam As Any) As Long
```

```
Private Declare Sub ReleaseCapture Lib "user32" ()
```

```
Private Const WM_NCLBUTTONDOWN = &HA1
```

```
Private Const HTCAPTION = 2
```

```
Private Sub cmdCancel_Click()
```

```
ck1.value = 0
```

```
ck2.value = 0
```

```
ck3.value = 0
```

```
Me.Hide
```

```
End Sub
```

```
Private Sub cmdOK_Click()
```

```
Me.Hide
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
End Sub
```

```
Private Sub Image1_MouseMove(Button As Integer, Shift As Integer, x As Single, Y As Single)
```

```
Dim lngReturnValue As Long
```

```
If Button = 1 Then
```

Call ReleaseCapture

```
lngReturnValue = SendMessage(Me.hWnd, WM_NCLBUTTONDOWN,  
HTCAPTION, 0&)
```

End If

End Sub

Public sMD5() As String

Public sNamaVirus() As String

Public JumlahVirus As Integer

Public Function BacaDatabase(sPath As String)

Static sTemp As String

Static sTmp() As String

Static sTmp2() As String

Static pisah As String

Static iCount As Integer

Static iTemp As Integer

pisah = Chr(13)

sTemp = ReadAnsiFile(sPath)

sTmp() = Split(sTemp, pisah)

iTemp = UBound(sTmp()) - 1

ReDim sMD5(iTemp) As String

ReDim sNamaVirus(iTemp) As String

```

For iCount = 1 To iTemp
    sTmp2() = Split(sTmp(iCount), ":")
    sMD5(iCount) = Mid(sTmp2(0), 2)
    sNamaVirus(iCount) = sTmp2(1)
Next

JumlahVirus = iTemp

End Function

Public Function isFileVirus(sPath As String, lstVirus As UniList) As Boolean

Static iCount As Integer
Static MD5file As String

MD5file = GET_MD5(sPath)

For iCount = 1 To JumlahVirus
    If sMD5(iCount) = MD5file Then
        lstVirus.AddItem "Virus Found ! [" & sNamaVirus(iCount) & "] - " & sPath
        isFileVirus = True
        Exit Function
    End If
Next

isFileVirus = False

End Function

```

```
Private Declare Function DeleteFile Lib "kernel32" Alias "DeleteFileW" (ByVal lpFileName As Long) As Long
```

```
Private Declare Function SetFileAttributes Lib "kernel32" Alias "SetFileAttributesW" (ByVal lpFileName As Long, ByVal dwFileAttributes As Long) As Long
```

```
Private Declare Function GetFileAttributes Lib "kernel32" Alias "GetFileAttributesW" (ByVal lpFileName As Long) As Long
```

```
Dim RDF As New clsFile
```

```
Public Function ReadUnicodeFile(sFilePath As String, Msg As Boolean, nStart As Long, nLenght As Long) As String
```

```
On Error Resume Next
```

```
Dim zFileName As String
```

```
Dim hFile As Long
```

```
Dim nFileLen As Long
```

```
Dim nOperation As Long
```

```
zFileName = sFilePath
```

```
hFile = RDF.VbOpenFile(zFileName, FOR_BINARY_ACCESS_READ_WRITE_EXISTING, LOCK_NONE)
```

```
If hFile > 0 Then
```

```
nFileLen = RDF.VbFileLen(hFile)
```

```
Dim bufData() As Byte
```



```
nOperation = RDF.VbReadFileB(hFile, nStart, nLenght, bufData)
```

```
ReadUnicodeFile = StrConv(bufData, vbUnicode)
```

```
RDF.VbCloseFile hFile
```

```
Else
```

```
If Msg = True Then MsgBox "#gagal membuka file ! :("
```

```
GoTo TERAKHIR
```

```
End If
```

```
Exit Function
```

```
TERAKHIR:
```

```
End Function
```

```
Public Function ReadAnsiFile(sFile As String) As String
```

```
Dim sTemp As String
```

```
Open sFile For Binary As #1
```

```
    sTemp = Space(LOF(1))
```

```
    Get #1, , sTemp
```

```
Close #1
```

```
ReadAnsiFile = sTemp
```

```
End Function
```

```
Public Function NormalizeAttribute(sPath As String)
```

```
On Error Resume Next
```

```
If GetFileAttributes(StrPtr(sPath)) = 4 Then
```

```
    SetFileAttributes StrPtr(sPath), 0
```

```

ElseIf GetFileAttributes(StrPtr(sPath)) = 6 Then
    SetFileAttributes StrPtr(sPath), 0
ElseIf GetFileAttributes(StrPtr(sPath)) = 2 Then '
    SetFileAttributes sPath, 0
ElseIf GetFileAttributes(StrPtr(sPath)) = 38 Then '
    SetFileAttributes StrPtr(sPath), 0
ElseIf GetFileAttributes(StrPtr(sPath)) = 39 Then '
    SetFileAttributes StrPtr(sPath), 0
End If
End Function

Public Function HapusFile(sPath As String)
On Error Resume Next

SetFileAttributes StrPtr(sPath), 0
DeleteFile StrPtr(sPath)
End Function

Public Function GET_MD5(FileName As String) As String
On Error GoTo Salah

Dim md5 As New clsMD5
Dim Buff As String

Buff = ReadUnicodeFile(FileName, False, 1, 2000)

```

```
Buff = Left(Buff, 2000)
```

```
md5.MD5Init
```

```
md5.DigestStrToHexStr Buff
```

```
GET_MD5 = md5.GetValues
```

```
Set md5 = Nothing
```

```
Exit Function
```

```
Salah:
```

```
End Function
```

```
Dim FSO As Object
```

```
Public BERHENTI As Boolean
```

```
Private Function GET_Folder(Folder As String, lbFile As UniLabel, lstInfo As  
UniList) As String
```

```
On Error Resume Next
```

```
Dim sFolder As Object
```

```
For Each sFolder In FSO.getFolder(Folder).subFolders
```

```
    GET_Folder (sFolder.path), lbFile, lstInfo
```

```
    If BERHENTI = True Then Exit Function
```

```
    GetFile sFolder.path, lbFile, lstInfo
```

```
Next
```

```
End Function
```

```

Private Function GetFile(path As String, lbFile As UniLabel, lstInfo As UniList)

Dim sFile As Object

For Each sFile In FSO.getFolder(path).Files

    DoEvents

    If BERHENTI = True Then Exit Function

    lbFile.Caption = sFile

    If frmOption.ck1.value = 1 Then

        If isProperFile(CStr(sFile), 3, "EXE DLL VBS VMX DB COM SCR BAT")
        = True Then

            If isFileVirus(CStr(sFile), lstInfo) = True Then

                If frmOption.ck2.value = 1 Then HapusFile CStr(sFile)

            End If

            If frmOption.ck3.value = 1 Then NormalizeAttribute CStr(sFile)

        End If

    Else

        If isFileVirus(CStr(sFile), lstInfo) = True Then

            If frmOption.ck2.value = 1 Then HapusFile CStr(sFile)

        End If

        If frmOption.ck3.value = 1 Then NormalizeAttribute CStr(sFile)

    End If

Next

End Function

Public Function ScanFolder(Folder As String, lbFile As UniLabel, lstInfo As
UniList)

```

```
Dim sFile As Object

Set FSO = CreateObject("Scripting.FileSystemObject")

For Each sFile In FSO.getFolder(Folder).Files

    DoEvents

    lbFile.Caption = sFile

    If frmOption.ck1.value = 1 Then

        If isProperFile(CStr(sFile), 3, "EXE DLL VBS VMX DB COM SCR BAT")
        = True Then

            If isFileVirus(CStr(sFile), lstInfo) = True Then

                If frmOption.ck2.value = 1 Then HapusFile CStr(sFile)

            End If

            If frmOption.ck3.value = 1 Then NormalizeAttribute CStr(sFile)

        End If

    Else

        If isFileVirus(CStr(sFile), lstInfo) = True Then

            If frmOption.ck2.value = 1 Then HapusFile CStr(sFile)

        End If

        If frmOption.ck3.value = 1 Then NormalizeAttribute CStr(sFile)

    End If

Next

GET_Folder Folder, lbFile, lstInfo

End Function
```