

PEMBUATAN APLIKASI STACK DOCKLET PADA WINDOWS DENGAN MENGGUNAKAN VB.NET

ABSTRAK

Teknologi informasi memacu semakin banyak pengembangan aplikasi kreatif. Banyaknya software yang digunakan terkadang membuat tampilan desktop dipenuhi banyak shortcut. Aplikasi stack docklet merupakan sebuah aplikasi yang dapat mengatur semua shortcut tersebut menjadi lebih tertata rapi, sehingga semua shortcut software tersebut dapat dikategorikan sesuai fungsinya. Hal ini memudahkan pengaksesan setiap software tersebut menjadi lebih mudah, efektif dan efisien. Aplikasi stack docklet dibuat dengan menggunakan bahasa pemrograman VB.NET dan dapat berjalan di sistem operasi Windows yang sudah terinstall .NET Framework.

Kata Kunci: shortcut, stack doklet

Deasy Indayanti
Dion Mirza

Fakultas Ilmu Komputer
Universitas Gunadarma
deasy@staff.gunadarma.ac.id

PENDAHULUAN

Perkembangan teknologi informasi dewasa ini sangat mempengaruhi proses pengerjaan suatu proyek dengan menggunakan data digital yang dapat diolah menjadi informasi berupa media suara, gambar dan rekaman video. Hasil *output*-nya sangat beragam. Begitu juga program yang digunakan dalam proses pengerjaannya. Banyak program yang disediakan untuk menyelesaikan hal-hal tersebut sehingga sering terlihat tampilan layar desktop yang penuh dengan berbagai macam shortcut dari beberapa program yang digunakan.

Pembuatan aplikasi yang dapat mengatur agar semua shortcut menjadi lebih tertata rapi dan dikategorikan sesuai fungsinya merupakan solusi yang diharapkan dapat berjalan dengan baik, benar dan terjamin keamanannya. Untuk menjamin bahwa aplikasi dapat berjalan sesuai apa yang diharapkan, maka digunakan framework yang mendukung aplikasi itu agar dapat berjalan. Pembuatannya tidak mempersulit pengembangnya dalam menyelesaikan suatu proses yang diinginkan.

Pada sistem operasi Windows terdapat framework yang secara default berjalan di atas sistem operasi ini yaitu .NET yang mendukung beberapa bahasa pemrograman seperti Visual Basic, C++, C# dan J#.

Bahasa Pemrograman Visual Basic .NET adalah bahasa pemrograman yang paling mudah dipelajari karena bentuk sintaks dan kalimat yang digunakan untuk menuliskan algoritma pemrograman mendekati *human language* yang sebenarnya, sehingga untuk memahami sebuah logika dari algoritma program yang ada sangat mudah sekali.

Penggunaan Visual Basic dalam membuat sebuah program sangat banyak dipakai sekarang ini karena penulisan algoritma dan sintaks programnya yang mudah dan juga framework pondasi pendukungnya yaitu .NET membuat bahasa ini banyak digunakan terutama pada sistem operasi Windows.

METODE PENELITIAN

Metode penelitian yang digunakan berupa studi kepustakaan. Untuk itu penulis mengumpulkan sejumlah buku referensi

yang berisi materi dasar tentang pemrograman Visual Basic .NET maupun berbagai macam tutorial tentang pemrograman di internet. Langkah selanjutnya adalah membuat rancangan tampilan maupun alur program yang akan digunakan untuk pembuatan aplikasi. Langkah terakhir adalah mengimplementasikannya ke dalam bahasa pemrograman yang digunakan, yaitu Visual Basic .NET.

Stack Docklet

Stack Docklet diciptakan oleh id user bernama Matonga sebagai sebuah *add-ons* yang berada dalam program *rocketdock*. Program itu digunakan sebagai bentuk aplikasi shortcut yang dapat menampung beberapa shortcut lain, yang kemudian ditampilkan dalam sebuah wadah form panel beserta penggunaan animasi yang menarik, sehingga pengaturan shortcut lebih tertata rapi dan menarik para pengguna. Pada penulisan ini penulis akan membuat sebuah aplikasi komputer yang akan digunakan untuk menampung semua shortcut yang ada pada desktop dalam bentuk *stack docklet*.

Microsoft .NET Framework

Microsoft .NET Framework adalah komponen yang dapat ditambahkan ke sistem operasi atau telah terintegrasi ke dalam Windows (mulai dari dan versi-versi Windows terbaru). Kerangka kerja ini menyediakan sejumlah besar solusi program untuk memenuhi kebutuhan umum suatu program baru, dan mengatur eksekusi program-program yang ditulis secara khusus untuk framework ini.

.NET Framework adalah kunci penawaran utama dari Microsoft, dan dimaksudkan untuk digunakan oleh sebagian besar aplikasi-aplikasi baru yang dibuat untuk platform Windows. Pada dasarnya, .NET Framework memiliki 2 komponen utama: CLR dan .NET Framework Class Library.

Program-program yang ditulis untuk .NET Framework dijalankan pada suatu lingkungan software yang mengatur persyaratan-persyaratan program. ini, yang juga merupakan suatu bagian dari .NET Framework, dikenal sebagai (CLR).

CLR menyediakan penampilan dari ,

sehingga para programmer tidak perlu mengetahui kemampuan tertentu yang akan menjalankan program. CLR juga menyediakan layanan-layanan penting lainnya seperti jaminan keamanan, pengaturan memori, dan /penanganan kesalahan pada saat *runtime*.

Class library dan CLR merupakan komponen inti dari .NET Framework. Kerangka kerja itu pun dibuat sedemikian rupa agar para programmer dapat mengembangkan program komputer dengan jauh lebih mudah, dan juga untuk mengurangi kerawanan aplikasi dan komputer dari beberapa ancaman keamanan.

CLR adalah turunan dari CLI (*Common Language Infrastructure*) yang saat ini merupakan standar . Untuk keterangan lebih lanjut, silakan mengunjungi situs atau kunjungi sumber pranala di bawah artikel ini.

Solusi-solusi program pembentuk dari .NET Framework meng-cover area yang luas dari kebutuhan program pada bidang *user interface*, pengaksesan data, koneksi , , pembuatan , numerik, dan komunikasi jaringan. Fungsi-fungsi yang ada dalam *class library* dapat digabungkan oleh dengan kodenya sendiri untuk membuat suatu program aplikasi baru.

Pada berbagai literatur dan referensi di internet, .NET Framework seringkali disingkat menjadi .NET saja. Menggunakan .NET Framework itu bermanfaat karena mudah, efisien, konsisten, dan mendorong produktivitas. Yang dimaksud dengan mudah di sini adalah kemudahan developer untuk membuat aplikasi yang dijalankan di .NET Framework. Mendukung lebih dari 20 bahasa pemrograman : VB.NET, C#, J#, C++, Pascal, Phyton (IronPhyton), PHP (PhLager).

Implikasi dari kemudahan pada saat proses pembuatan aplikasi adalah efisiensi dari proses produktivitas, baik efisien dalam hal waktu pembuatan aplikasi atau efisien dalam hal lain, seperti biaya. Implikasi lain dari kemudahan-kemudahan waktu proses pembuatan aplikasi adalah konsistensi pada aplikasi yang diciptakan.

Misalnya, dengan adanya Base Class Library, maka bisa digunakan objek atau *class* yang dibuat untuk aplikasi berbasis windows pada aplikasi berbasis web.

Dengan adanya kode yang bisa diintegrasikan ke dalam berbagai macam aplikasi ini, maka konsistensi kode-kode aplikasi kita dapat terjaga.

Semua kemudahan di atas pada akhirnya akan membuat produktivitas menjadi lebih baik. Produktivitas naik, terutama produktivitas para pengembang, akan berdampak pada meningkatnya produktivitas suatu perusahaan atau proyek.

Sistem Operasi Microsoft Windows

Sistem Operasi Microsoft Windows atau yang lebih dikenal dengan sebutan Windows adalah keluarga yang dikembangkan oleh , dengan menggunakan antarmuka berbasis grafik ().

Sistem operasi Windows telah berevolusi dari , sebuah sistem operasi yang berbasis modus teks dan *command-line*. Windows versi pertama, Windows Graphic Environment 1.0 pertama kali diperkenalkan pada , tetapi baru keluar pasar pada bulan November tahun untuk memenuhi kebutuhan komputer dengan tampilan bergambar.

Windows 1.0 merupakan perangkat lunak 16-bit tambahan (bukan merupakan sistem operasi) yang berjalan di atas MS-DOS (dan beberapa varian dari MS-DOS), sehingga ia tidak akan dapat berjalan tanpa adanya sistem operasi DOS. Versi 2.x, versi 3.x juga sama. Beberapa versi terakhir dari Windows (dimulai dari versi 4.0 dan Windows NT 3.1) merupakan sistem operasi mandiri yang tidak lagi bergantung kepada sistem operasi . Microsoft Windows kemudian bisa berkembang dan dapat menguasai penggunaan sistem operasi hingga mencapai 90%.

HASIL DAN PEMBAHASAN

Aplikasi *Stack Docklet* merupakan aplikasi pada PC yang akan mengorganisir semua shortcut yang berada pada desktop. Tujuan aplikasi ini adalah agar para penggunanya dapat mengatur dan menata semua shortcut lebih mudah diakses karena setiap shortcut dimasukkan ke dalam kategori yang sesuai dengan keinginan pengguna. Semua shortcut tersebut ditampilkan kembali dalam bentuk *grid stack docklet* menu pilihan yang diharapkan lebih memudahkan para pengguna aplikasi untuk mengakses beberapa shortcut yang sudah dikategorikan tersebut.

Ketika pertama kali menjalankan program, pengguna aplikasi diminta untuk menentukan folder yang akan dijadikan *stack docklet* lalu menentukan nama dari kategori untuk *stack docklet* dan terakhir menentukan icon yang akan digunakan sebagai tampilan dari *stack docklet* tersebut. Sesudah itu pengguna melanjutkan dengan mengklik tombol "Buat Link di Desktop" untuk merangkum beberapa shortcut di dalam satu folder menjadi satu link di desktop atau menekan tombol "Batal" untuk keluar dari aplikasi. Ketika link *stack docklet* yang merangkum beberapa shortcut di dalamnya dijalankan, akan ditampilkan

kembali ke depan desktop semua shortcut yang ada di dalam folder tersebut dalam bentuk *grid*.

Pembuatan Modul Utama

Pada aplikasi *Stack Docklet* ini, sub-program yang pertama kali dijalankan adalah modul utama. Di sini program menentukan form mana yang akan ditampilkan, apakah pembuatan *stack* baru atau menampilkan *stack docklet*.

Dengan menggunakan *if args.length = 0* maka dicek apakah program dijalankan tanpa argumen. Jika ya maka ditampilkan form *stack* baru, tapi jika tidak maka terdapat argumen dan yang ditampilkan adalah form panel *stack docklet*. Prosedur *getScreenHeight* di sini digunakan untuk menampilkan posisi form panel *stack docklet*, di mana nantinya posisi form berada di bagian bawah layar tempat *shortcut stack* di klik.

```
Module mainmod
...
Public Sub Main(ByVal args() As String)
...
If args.Length = 0 Then
'Jika tidak terdapat argumen maka tampilkan form NewStack
Dim frmNS As New NewStack
frmNS.ShowDialog()
Else
Dim frm As New Stack(args)
frm.ShowDialog()
End If
End Sub
Public Function getScreenHeight() As Integer
Return Screen.PrimaryScreen.Bounds.Bottom
End Function
End Module
```

Pembuatan Form Stack Baru

Pada aplikasi ini form *stack* baru merupakan window yang ditampilkan jika ketika shortcut aplikasi ini dijalankan tidak terdapat argumen. Untuk pembuatan shortcut digunakan *IWshRuntimeLibrary* dan juga disini terdapat beberapa prosedur yang akan dijalani dalam proses pembuatan shortcut *stack*.

Pertama, prosedur *findFolder_Click*. Prosedur ini dijalankan ketika pengguna mengklik button *btnFindFolder* di mana window folder browser dialog dipanggil dan pengguna diminta untuk menentukan folder yang akan digunakan sebagai *stack*. Ketika tombol OK pada folder browser dialog ditekan maka dicek apakah ada folder yang sudah dipilih oleh pengguna. Jika ada folder yang dipilih pengguna, maka alamat folder yang sudah dipilih dimasukkan ke textbox *txtStackFolder*.

```
Private Sub btnFindFolder_Click(...) Handles btnFindFolder.Click
If (Me.FolderBrowserDialog1.ShowDialog = Windows.Forms.DialogResult.OK) Then
If (Me.FolderBrowserDialog1.SelectedPath.Length > 0) Then
Me.txtStackFolder.Text = Me.FolderBrowserDialog1.SelectedPath
End If
End If
End Sub
```

Prosedur kedua yaitu *txtStackFolder_TextChanged*, yang dijalankan setiap kali terjadi perubahan nilai pada textbox *txtStackFolder*. Prosedur ini digunakan untuk mengambil nama folder yang dipilih oleh pengguna sebelumnya di prosedur *btnFindFolder*. Karena nilai dari textbox *txtStackFolder* adalah alamat folder maka pada prosedur ini dilakukan proses substring untuk mengambil bagian

akhir pada nilai textbox *txtStackFolder*. Proses substring dimulai dari indeks terakhir ditambah 1 dari huruf "\ " sampai kata terakhir, lalu hasilnya dimasukkan ke dalam textbox *txStackName* yang dijadikan sebagai nama *stack* nantinya.

Prosedur ketiga adalah *btnFindIcon*. Prosedur ini dijalankan ketika pengguna mengklik button *btnFindIcon* di mana window open file dialog dipanggil dan pengguna diminta untuk menentukan file yang akan digunakan sebagai icon dari *stack*. Sebelum dipanggil window dialog, perlu ditentukan dahulu file yang dicari ber-ekstensi "ico", kemudian filter pada open file dialog di atur untuk menampilkan hanya file ber-ekstensi "*.ico" dan juga bisa all files "*. *". Setelah itu ketika tombol OK pada open file dialog ditekan maka dicek apakah ada file icon yang sudah dipilih oleh pengguna. Jika ada file icon yang dipilih pengguna, maka

alamat file icon yang sudah dipilih dimasukkan ke textbox *txtStackIcon*.

Prosedur terakhir pada form ini adalah prosedur *btnCreateLink_Click*, yaitu pembuatan link shortcut di desktop. Prosedur ini dijalankan ketika pengguna mengklik tombol *btnCreateLink*. Pertama-pertama program mengambil alamat dari desktop pengguna lalu menyimpannya di variabel *sDesktop*. Lalu variabel yang digunakan sebagai *stack shortcut* adalah link berikutnya. Program mencek apakah pengguna sudah memilih folder untuk dijadikan *stack*. Jika belum maka akan keluar message box berjudul "Error membuat Stack" dengan isi "Silahkan masukan alamat folder untuk *stack* terlebih dahulu" yang disertai

gambar peringatan.

Jika folder *stack* sudah dipilih, maka proses berikutnya ialah menentukan alamat *stack* dengan alamat desktop pengguna yang sudah disimpan di dalam variabel *sDesktop* yaitu memasukan nilainya ke dalam variabel *link*. Lalu jika ada icon yang dipilih pengguna maka kita definisikan lokasi icon shortcut *stack* ke dalam *link*.


```
Private Sub txtStackFolder_TextChanged(...) Handles txtStackFolder.TextChanged
    If txtStackFolder.Text.Length > 0 Then
        'Mengambil sebagian teks dari folder menjadi nama
        Me.txtStackName.Text =
        txtStackFolder.Text.Substring(txtStackFolder.Text.LastIndexOf("\") + 1)
    End If
End Sub
```

```
Private Sub btnFindIcon_Click(...) Handles btnFindIcon.Click
    ...
    OpenFileDialog1.DefaultExt = ".ico"
    OpenFileDialog1.Filter = "Windows Icon (*.ico)|*.ico|All files (*.*)|*.*"
    If (Me.OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK) Then
        If (Me.OpenFileDialog1.FileName.Length > 0) Then
            Me.txtStackIcon.Text = Me.OpenFileDialog1.FileName
        End If
    End If
End Sub
```

Berikutnya penentuan argumen yang berada di dalam link yaitu dengan variabel `sStackType`, `sAnimType` dan `sWinStyle`. Ketiga variabel ini menentukan perilaku stack docklet saat ditampilkan. `sStackType` sebagai tipe tampilan stack, bernilai `"/g"`, yang berarti semua shortcut pada folder ditampilkan secara grid atau bisa dibilang berbentuk kolom baris.

`sAnimType` sebagai tipe animasi stack, bernilai `"/u"` artinya pergerakan animasi ke atas (*up*). Akhirnya, `sWinStyle` sebagai style dari tampilan panel, bernilai `"/ag"` artinya tampilan panel stack docklet berpenampilan glass (*all glass*).

Semua argumen yang sudah ditentukan dimasukkan ke dalam link. Setelah itu tentukan target program yang dijalankan ketika shortcut stack dijalankan, yaitu memberikan alamat target di mana program Stack Docklet ini dijalankan pertama kali. Terakhir baru menyimpan link di alamat desktop pengguna yang kemudian muncul tampilan message box berjudul "Sukses" disertai kata-kata "... Selamat menikmati Stack Docklet!".

Tampilan outputnya adalah sebagai berikut:



Gambar 1. Tampilan Halaman Form Stack Baru

Pembuatan Form Panel Stack Doklet

Form panel stack docklet merupakan window yang ditampilkan jika ketika shortcut aplikasi ini dijalankan terdapat beberapa argumen di dalamnya. Beberapa argumen ini menentukan tingkah laku bagaimana form panel stack docklet ini ditampilkan nantinya.

Prosedur pertama yang dijalankan saat program mengakses form ini adalah

prosedur `New(...)`, di mana program akan mengambil setiap argumen yang ada di dalam link shortcut. Di dalam link shortcut yang dibuat di form stack baru dimasukkan 4 argumen ke dalamnya. Pertama, alamat folder stack dimasukkan ke dalam variabel `mStackFolderPath`. Kedua, jenis tampilan stack dimasukkan ke dalam variabel `mStackStyle`. Ketiga, tipe animasi stack dimasukkan ke dalam variabel `slideType`, dan keempat tampilan window panel stack dimasukkan ke dalam variabel `stackLook`.

Setelah semuanya sudah ditentukan barulah dijalankan prosedur `stack_load` untuk menampilkan panel stack docklet. Prosedur ini dijalankan saat form akan ditampilkan, tepatnya setelah prosedur `new` di eksekusi terlebih dahulu. Di dalam prosedur ini juga terdapat pemanggilan sub-prosedur lain yang akan dijelaskan nanti.

Dalam prosedur ini program menentukan judul dari form panel stack docklet dengan mengambil nama folder yang dijadikan stack, lalu membuat list dari semua shortcut yang ada di dalam folder. Lalu diterapkan efek kaca terhadap tampilan window form panel stack docklet, dan terakhir animasi pergerakan form dari bawah ke atas dijalankan.

Prosedur `BuildListOfLinks` digunakan untuk mengambil setiap shortcut di dalam folder untuk dijadikan custom komponen lalu dimasukkan ke dalam komponen `tableLayoutPanel` sesuai banyaknya shortcut. Mula-mula semua shortcut yang ada dimasukkan ke dalam variabel array `links`, lalu dihitung banyaknya shortcut. Jumlah shortcut di dalam folder menentukan lebar dan tinggi dari form panel stack docklet. Jika lebih dari 12 maka ada 6 kolom yang disediakan tabel, tetapi jika jumlahnya kurang dari itu hanya 4 kolom yang disediakan pada tabel.

Besarnya tinggi form ditentukan dengan cara banyaknya shortcut dibagi 6 atau 4 (banyak kolom sesuai jumlah shortcut) dikalikan 65 pixel (besar icon shortcut) ditambah 60 pixel (untuk ovalshape open folder). Sedangkan lebar form dihitung dari 82 pixel dikalikan banyaknya kolom ditambah 20 pixel. Jika tidak ditambah 20 pixel maka shortcut di sebelah kanan akan terlihat sedikit terpotong.

Selanjutnya, setelah tampilan panel sudah disiapkan, program menentukan waktu animasi berjalan. Waktu animasi ditentukan oleh banyaknya shortcut dan juga menentukan tinggi form panel stack docklet nantinya. Jika jumlah shortcut genap maka perhitungan lama animasi adalah 10 ms dikalikan banyaknya shortcut. Jika jumlahnya ganjil maka banyaknya shortcut dikurangi 1 baru dikali 10 ms.

Setelah itu proses berikutnya adalah memasukkan semua atribut di shortcut ke dalam komponen custom dari program dengan menggunakan modul `WinAPI`. Atribut yang diambil ialah nama, alamat target program dan icon yang digunakan shortcut. Jika di dalam folder stack tidak

```
Private Sub btnCreateLink_Click(...) Handles btnCreateLink.Click
    ...
    Dim sDesktop As String =
    Environment.GetFolderPath(Environment.SpecialFolder.DesktopDirectory)
    If (txtStackFolder.Text.Length > 0) Then
        ...
        link = CType(shell.CreateShortcut(sDesktop & "\" & Me.txtStackName.Text & ".lnk"),
        IWshShortcut)

        If Me.txtStackIcon.Text.Length > 0 Then
            link.IconLocation = Me.txtStackIcon.Text
        End If
        sStackType = "/g"
        sAnimType = "/u"
        sWinStyle = "/ag"
        link.Arguments = "" & txtStackFolder.Text & "" & " " & sStackType & " " &
        sAnimType & " " & sWinStyle
        link.TargetPath = Application.StartupPath & "\\Stack Docklet.exe"
        link.Save()
        MsgBox("Link " & txtStackName.Text & "telah dibuat dan ditempatkan di desktop."
        & vbCrLf & vbCrLf & "Pindahkan dan Pin Link-nya ke Windows 7 taskbar." & vbCrLf &
        "Silahkan menikmati Stack Docklet!", MsgBoxStyle.Information, "Sukses!")
        Me.Close()
    Else
        MsgBox("Silahkan masukan alamat folder untuk stack terlebih dahulu",
        MsgBoxStyle.Exclamation, "Error membuat Stack")
        Me.txtStackFolder.Focus()
    End If
End Sub
```

```

Public Sub New(ByVal args() As String)
...
Me.mStackFolderPath = args(0)
If args.Count > 1 Then
    Select Case args(1)
        ...
        Case "/g"
            Me.mStackStyle = StackStyles.GRID
        ...
    End Select
If args.Count > 2 Then
    Select Case args(2)
        ...
        Case "/u"
            Me.slideType = SlideTypes.BOTTOM_UP
        ...
    End Select
If args.Count > 3 Then
    Select Case args(3)
        ...
        Case "/ag"
            Me.stackLook = "ALL_GLASS"
        ...
    End Select
...
End Sub

```

```

Private Sub stack_Load(...) Handles Me.Load
...
Me.Text = Me.mStackFolderPath.Substring(Me.mStackFolderPath.LastIndexOf("\")
+ 1)
BuildListOfLinks(Me.mStackFolderPath)
...
Select Case Me.slideType
    ...
    Case SlideTypes.BOTTOM_UP
        StartSlideUp(iScreenHeight - 60, pos.X, Me.stackHeight)
    ...
End Sub

```

terdapat shortcut maka komponen yang dimasukkan ke dalam tabel layout panel adalah komponen label dengan tulisan "Tidak terdapat link"

```

Private Sub BuildListOfLinks(ByVal LinkFolderPath As String)
    Dim links() As String = Directory.GetFiles(LinkFolderPath, "*.lnk")
...
If links.Count > 0 Then
    'Mengatur tinggi form
    Select Case Me.mStackStyle
        Case StackStyles.GRID
            If links.Count >= 12 Then
                Me.TableLayoutPanel1.ColumnCount = 6
                Me.TableLayoutPanel1.ColumnStyles.Add(New ColumnStyle(SizeType.AutoSize))

                Me.stackHeight = (CInt(Math.Ceiling(links.Count / 6)) * 65) + 60
                Me.stackWidth = (82 * 6) + 20
            Else
                Me.TableLayoutPanel1.ColumnCount = 4
                Me.TableLayoutPanel1.ColumnStyles.Add(New ColumnStyle(SizeType.AutoSize))
                Me.stackHeight = (CInt(Math.Ceiling(links.Count / 4)) * 65) + 60
                Me.stackWidth = (82 * 4) + 20
            End If
            If (links.Count Mod 2 = 0) Then
                Me.slideTime = links.Count * 10
            Else
                Me.slideTime = (links.Count - 1) * 10
            End If
        End Select
    ...
    For Each linkPathName As String In links

```

Perhitungan animasi pergeseran form panel *stack docklet* dari atas ke bawah dilakukan di dalam prosedur *StartSlideUp*. Perhitungannya, pergerakan panel per detik mengambil tinggi dari form yang sudah dihitung sebelumnya, kemudian dibagi dengan hasil bagi dari waktu animasi dibagi

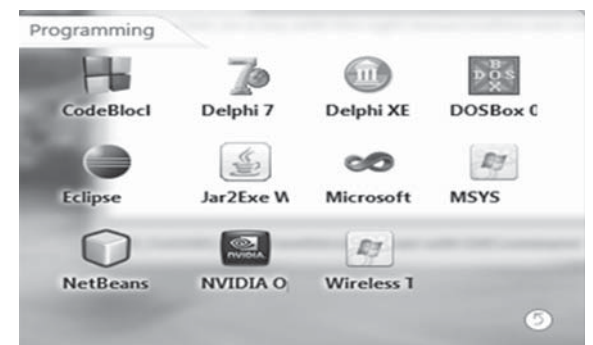
interval pada *timer*.

Timer pun dijalankan dalam prosedur ini. Saat *timer* berjalan dilakukan pengecekan. Jika waktu yang telah berjalan bernilai sama dengan waktu yang dibutuhkan untuk animasi maka *timer* dimatikan. Jika belum, maka akan dilakukan penambahan waktu terhadap

variabel yang menampung waktu yang telah berjalan sebanyak 20 point. Langkah berikutnya, menjalankan prosedur *DoSlide*, yaitu prosedur yang menggerakkan atribut top dari form dan menambah tinggi form panel *stack docklet*

Bagian terakhir yaitu jika pengguna beralih ke window selain form ini. Jika program ini tidak aktif maka otomatis program *stack docklet* dimatikan. Program otomatis keluar jika program dialihkan .

Tampilan outputnya adalah sebagai berikut:



Gambar 2. Tampilan Halaman Form Panel Stack Docklet

Pembuatan Modul Win API

Modul WinAPI digunakan pada pembuatan form panel *stack docklet*. Modul ini menyediakan berbagai prosedur untuk menjalankan animasi dan mendapatkan icon dari sebuah shortcut. Modul ini mengambil juga beberapa fungsi dari System.Runtime.InteropServices, yang berfungsi mengakses file shell32.dll. Dengan shell32.dll program dapat mengambil atribut gambar icon yang digunakan oleh shortcut. Gambar icon yang diambil dikategorikan dalam besar kecilnya. Setelah itu atribut gambar icon dimasukkan ke dalam variabel *myIcon*.

Uji Coba Aplikasi

Uji coba aplikasi ini dilakukan pada komputer yang sudah menggunakan sistem operasi Windows 7. Sistem operasi ini sudah terinstall secara default .NET Framework 3.5, sehingga program dapat dijalankan tanpa kendala. Langkah-langkah yang harus dilakukan dalam menjalankan aplikasi *stack docklet* adalah sebagai berikut:

1. Menjalankan program Visual Studio 2010 yang merupakan IDE yang digunakan untuk membuat program ini, dilanjutkan dengan membuka project *stack docklet* melalui menu Open Project seperti diperlihatkan pada Gambar 3.
2. Setelah project tersebut terbuka, langkah selanjutnya yaitu kompilasi program. Proses kompilasi dapat dilakukan dengan mengakses menu pulldown Build, lalu pilih Build Solution atau bisa menggunakan shortcut F6 untuk mem-*build* project secara langsung. Setelah proses kompilasi berhasil, aplikasi bisa dijalankan melalui menu pulldown debug lalu pilih Start Debugging


```

...
Try
    iIcon = GetIcon(sTargetPath, False)
    bmIcon = iIcon.ToBitmap
Catch ex As Exception
    ...
    bmIcon = My.Resources.alert_lrg
End Try
sDisplayName = Path.GetFileNameWithoutExtension(linkPathName)
Select Case Me.mStackStyle
    Case StackStyles.GRID
        If Me.stackLook = "ALL_GLASS" Then
            AddGridItem(sDisplayName, bmIcon, sTargetPath)
        ...
        End If
    End Select
Next
Else
    ...
    nLabel.Text = "Tidak terdapat link"
    nLabel.TextAlign = ContentAlignment.MiddleRight
    Me.TableLayoutPanel1.Controls.Add(nLabel)
    ...
End If
...
End Sub

```

```

Private Sub StartSlideUp(...)
    Me.mHeightInc = CInt(Me.stackHeight / (Me.slideTime / Me.Timer1.Interval))
    ...
    Me.Timer1.Enabled = True
End Sub

Private Sub Timer1_Tick(...) Handles Timer1.Tick
    If Me.mPassedTicks = Me.slideTime Then
        Me.Height = Me.stackHeight
        Timer1.Stop()
        Exit Sub
    Else
        Me.mPassedTicks += 20
        Call DoSlide()
    End If
End Sub

Private Sub DoSlide()
    ...
    Me.Height += Me.mHeightInc
    Me.Top -= Me.mHeightInc
    ...
End Sub

```

```

Private Sub Stack_Deactivate(...) Handles Me.Deactivate
    Me.Close()
End Sub

```

```

Imports System.Runtime.InteropServices

Module winapi
    Private Declare Ansi Function SHGetFileInfo Lib "shell32.dll" (...) As IntPtr

    Public Function GetIcon(ByVal fileName As String, Optional ByVal smallIcon As Boolean = True) As Icon
        ...
        If smallIcon Then
            hImgSmall = SHGetFileInfo(fileName, 0, shinfo, Marshal.SizeOf(shinfo), SHGFI_ICON Or SHGFI_SMALLICON)
        Else
            hImgLarge = SHGetFileInfo(fileName, 0, shinfo, Marshal.SizeOf(shinfo), SHGFI_ICON Or SHGFI_LARGEICON)
        End If
        "The icon is returned in the hIcon member of the shinfo struct."
        Dim myIcon As System.Drawing.Icon
        myIcon = System.Drawing.Icon.FromHandle(shinfo.hIcon)
        Return myIcon
    End Function

```

seperti terlihat pada Gambar 4 dibawah ini.



Gambar 4. Tampilan Stack Docket Form Stack Baru

Gambar 4 merupakan output tampilan awal pada aplikasi *stack docket*. Di dalam tampilan tersebut terdapat tulisan "Stack Docket v1.0" yang merupakan nama dari aplikasi yang dibuat beserta versinya. Di sebelah textbox Folder Stack terdapat tombol bergambar folder yang digunakan untuk memanggil folder browser dialog untuk menentukan folder stack yang akan digunakan. Gambar 5 memperlihatkan tampilan setelah tombol tersebut di klik.

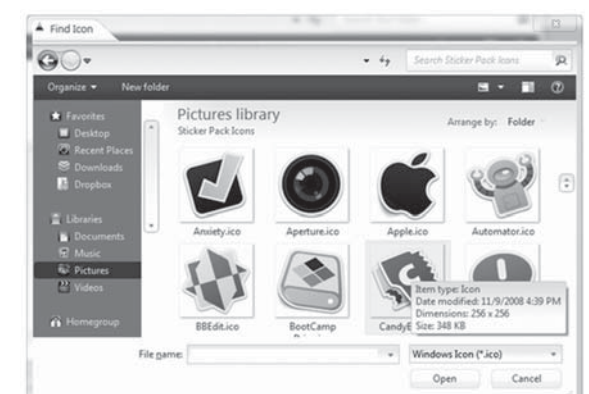


Gambar 5. Tampilan Folder Browser Dialog

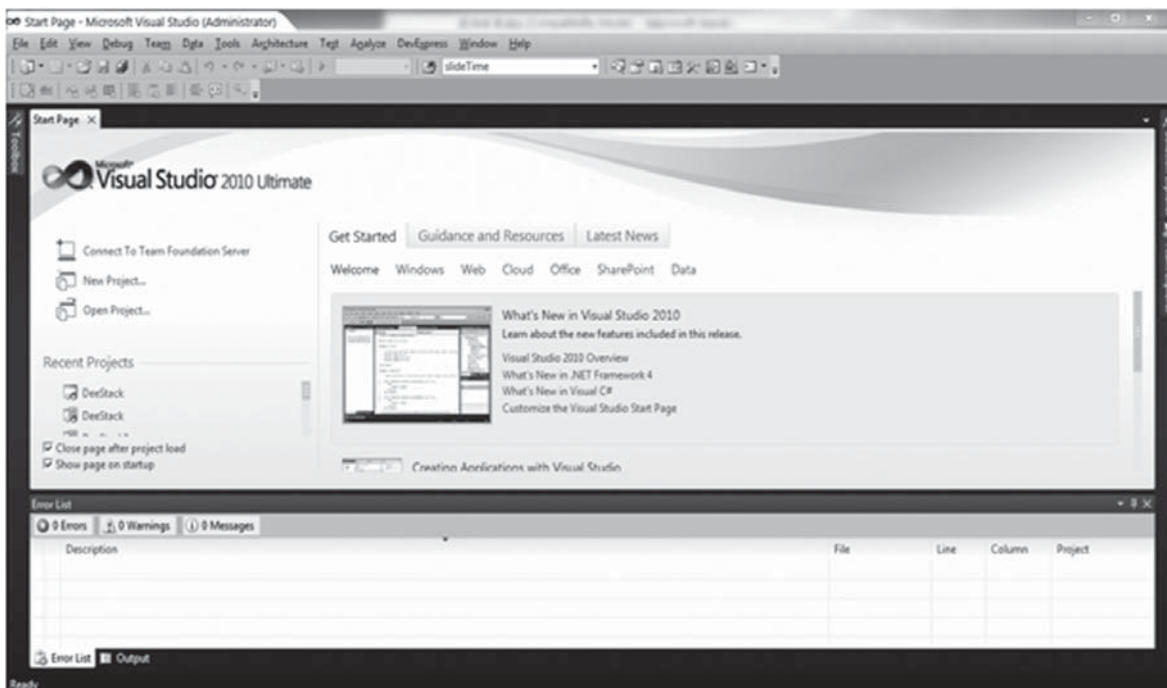
Setelah pengguna menentukan folder yang dijadikan stack maka secara otomatis textbox nama stack pada form stack baru terisi dengan sendirinya seperti ditampilkan pada Gambar 6. Selanjutnya pengguna akan menentukan icon yang akan digunakan untuk tampilan *stack docket*-nya. Dengan menekan tombol bergambar folder yang berada di sebelah textbox stack icon akan muncul open file dialog untuk menentukan icon yang digunakan pada stack seperti diperlihatkan pada Gambar 7.



Gambar 6. Tampilan Form Stack Baru setelah Folder Stack ditentukan

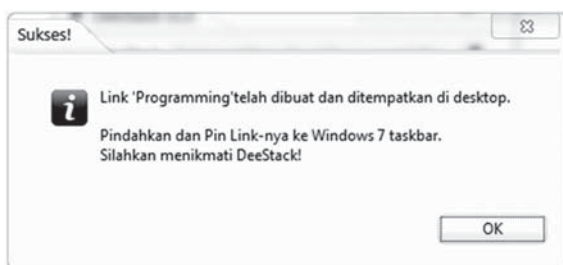


Gambar 7. Tampilan Open File Dialog



Gambar 3. Tampilan Visual Studio 2010

Setelah icon ditentukan barulah dibuat link stack di desktop dengan menekan tombol bernama Buat Stack di Desktop. Jika pembuatan link shortcut berhasil akan muncul tampilan message box yang akan memberitahu pengguna bahwa pembuatan link berhasil. Tampilan message box diperlihatkan pada Gambar 8.



Gambar 8. Tampilan Message Box Sukses

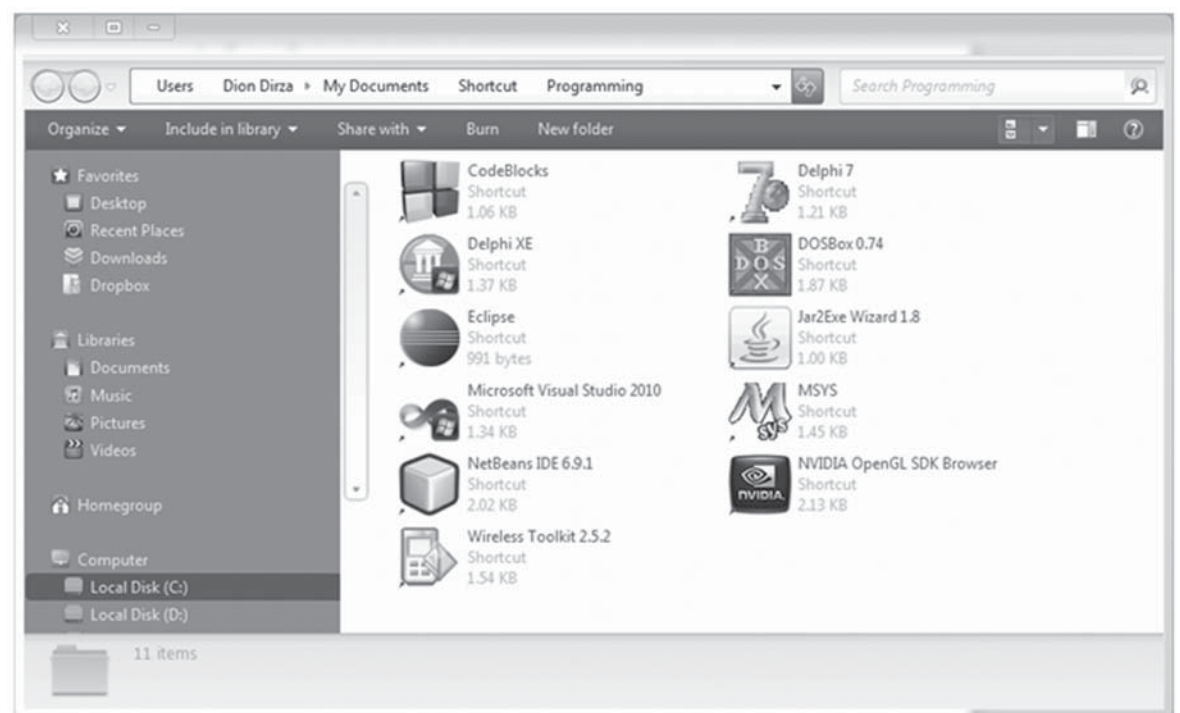
3. Link yang sudah dibuat di desktop kemudian dapat di pin di taskbar yang kemudian dapat dijalankan langsung melalui tempat tersebut. Ketika shortcut pada taskbar diklik, akan terlihat tampilan form panel stack docklet seperti ditampikan pada Gambar 9.



Gambar 9. Tampilan Form Panel Stack Docklet di Taskbar

Jika salah satu icon shortcut pada form ini diklik maka program tersebut akan dijalankan sesuai alamat target program masing-masing. Pada pojok kanan bawah terdapat komponen *ovalshape* yang digunakan untuk menampilkan gambar panah. Jika diklik oleh pengguna maka fungsi komponen tersebut adalah membuka window folder

tempat di mana disimpan alamat semua shortcut seperti ditampikan Gambar 11.



Gambar 10. Tampilan Window Folder Shortcut Stack Docklet disimpan

Kesimpulan

Aplikasi *stack docklet* dibuat agar menjadi aplikasi yang dapat membantu pengguna dalam mengatur dan menata setiap shortcut di desktop sehingga lebih rapi. Dengan teknologi .NET Framework juga penggunaan resource untuk aplikasi ini sangat minimalis sehingga pengguna tidak mengalami *system hang* atau error pada saat runtime. Interface yang sederhana juga membuat pengguna tidak bingung dalam pemakaian program ini. Dalam pembuatan aplikasi *stack*

docklet ini, digunakan bahasa pemrograman VB.NET yang lebih mempermudah dalam pembuatan aplikasi ini. Dengan adanya IDE seperti Visual Studio 2010 pembuatan GUI aplikasi lebih mudah dan juga *reference library* yang mudah untuk menggunakan *file shell* untuk pengaksesan file shortcut di dalam aplikasi ini. Aplikasi ini dapat dijalankan pada komputer dengan karakteristik minimal yaitu, dengan sistem operasi Windows XP atau Windows 7 dan .NET Framework 3.5.

DAFTAR PUSTAKA

- Bill Sheldon, Billy Hollis, Kent Sharkey, Gastón Hillar, Rob Windsor, Jonathan Marbutt. *Professional Visual Basic 2010 and .NET 4*. Wrox. 2010.
- Thearon Willis, Bryan Newsome. *Beginning Microsoft Visual Basic*

2010. Wrox. 2010.

.NET Framework, URL : <http://kambing.ui.ac.id/bebas/vo6/Kuliah/SistemOperasi/BUKU/SistemOperasi-4.X-1/ch08so8.html>

Tanggal akses : 23 January 2012 16:15 Wikipedia Indonesia - .NET Framework, URL : http://id.wikipedia.org/wiki/.NET_Framework Tanggal akses : 23 January 2012 16:00

