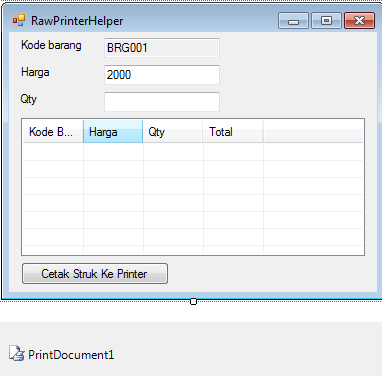
**LATIHAN LISTVIEW-PRINTER**



|  |  |  |
| --- | --- | --- |
| **Object** | **Properties** | |
| **Name** | **Text** |
| Form | FormCetak.vb | Cetak |
| Label1 | Label1 | Kode barang |
| Label2 | Label2 | Harga |
| Label3 | Label3 | Qty |
| TextBox1 | TextBox1 |  |
| TextBox2 | TextBox2 |  |
| TextBox3 | TextBox3 |  |
| Button1 | Button1 | Cetak Struk Ke Printer |
| ListView1 | ListView1 |  |
| PrintDocument | PrintDocument1 |  |
|  | | |
| Class | RawPrinterHelper.vb |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ListView | | | |
| Properties | CheckBoxes | TRUE |  |
| FullRowSelect | TRUE |  |
| GridLines | TRUE |  |
| ListView Taks | View | Detail |  |
| Edit Coloums | Add | Kode |
| Add | Harga |
| Add | Qty |
| Add | Total |

Form: FormCetak

Imports System.Drawing.Printing

Public Class FormCetak

Sub cetak\_struk()

Dim s As String

Const sLN As String = ControlChars.CrLf

Const tabx As String = ControlChars.Tab

Dim pd As New PrintDialog

Dim oPS As New System.Drawing.Printing.PrinterSettings

Dim sps As String

s &= " Toko Brownis Fortuna" & sLN

s &= "Otlate : " & sLN

s &= "Kasir :" & sLN

s &= "No struk :" & sLN

s &= "tanggal : " & TimeOfDay & " " & sLN

s &= "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" & sLN

For Each item As ListViewItem In ListView1.Items

s &= "" & item.Text + vbTab + item.SubItems(1).Text + vbTab + item.SubItems(2).Text + vbTab + item.SubItems(3).Text & sLN

Next

s &= " " & sLN

s &= " " & sLN

s &= " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" & sLN

s &= " Subtotal : 101.000" & sLN

s &= " Cash : 101.000" & sLN

s &= " Kembali : 101.000" & sLN

s &= " " & sLN

s &= " " & sLN

s &= " " & sLN

s &= "\*\*\*\*\*\*Terima Kasih atas kunjungan anda\*\*\*\*\*" & sLN

'MessageBox.Show(oPS.PrinterName)

MessageBox.Show(s)

'pd.PrinterSettings = New PrinterSettings

'Dim prname As String

'RawPrinterHelper.SendStringToPrinter(oPS.PrinterName, s)

End Sub

Private Sub TextBox1\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox1.KeyPress

If e.KeyChar = Chr(13) Then

TextBox2.Focus()

End If

End Sub

Private Sub TextBox2\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox2.KeyPress

If e.KeyChar = Chr(13) Then

TextBox3.Focus()

End If

End Sub

Private Sub TextBox3\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBox3.KeyPress

If e.KeyChar = Chr(13) Then

'add

With ListView1

.Items.Add(TextBox1.Text)

.Items(.Items.Count - 1).SubItems.Add(TextBox2.Text)

.Items(.Items.Count - 1).SubItems.Add(TextBox3.Text)

.Items(.Items.Count - 1).SubItems.Add(CDbl(TextBox2.Text) \* CDbl(TextBox3.Text))

.AutoResizeColumns(ColumnHeaderAutoResizeStyle.HeaderSize)

End With

TextBox1.Text = "BRG" + Mid("000", 1, 3 - (Val(Mid(TextBox1.Text, 4, 3)) + 1).ToString.Length) + (Val(Mid(TextBox1.Text, 4, 3)) + 1).ToString

End If

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

cetak\_struk()

'MessageBox.Show("Mencetak")

End Sub

End Class

CLASS: RawPrinterHelper

Imports System.IO

Imports System.Drawing.Printing

Imports System.Runtime.InteropServices

Public Class RawPrinterHelper

<StructLayout(LayoutKind.Sequential, CharSet:=CharSet.Unicode)> \_

Structure DOCINFOW

<MarshalAs(UnmanagedType.LPWStr)> Public pDocName As String

<MarshalAs(UnmanagedType.LPWStr)> Public pOutputFile As String

<MarshalAs(UnmanagedType.LPWStr)> Public pDataType As String

End Structure

<DllImport("winspool.Drv", EntryPoint:="OpenPrinterW", \_

SetLastError:=True, CharSet:=CharSet.Unicode, \_

ExactSpelling:=True, CallingConvention:=CallingConvention.StdCall)> \_

Public Shared Function OpenPrinter(ByVal src As String, ByRef hPrinter As IntPtr, ByVal pd As Long) As Boolean

End Function

<DllImport("winspool.Drv", EntryPoint:="ClosePrinter", \_

SetLastError:=True, CharSet:=CharSet.Unicode, \_

ExactSpelling:=True, CallingConvention:=CallingConvention.StdCall)> \_

Public Shared Function ClosePrinter(ByVal hPrinter As IntPtr) As Boolean

End Function

<DllImport("winspool.Drv", EntryPoint:="StartDocPrinterW", \_

SetLastError:=True, CharSet:=CharSet.Unicode, \_

ExactSpelling:=True, CallingConvention:=CallingConvention.StdCall)> \_

Public Shared Function StartDocPrinter(ByVal hPrinter As IntPtr, ByVal level As Int32, ByRef pDI As DOCINFOW) As Boolean

End Function

<DllImport("winspool.Drv", EntryPoint:="EndDocPrinter", \_

SetLastError:=True, CharSet:=CharSet.Unicode, \_

ExactSpelling:=True, CallingConvention:=CallingConvention.StdCall)> \_

Public Shared Function EndDocPrinter(ByVal hPrinter As IntPtr) As Boolean

End Function

<DllImport("winspool.Drv", EntryPoint:="StartPagePrinter", \_

SetLastError:=True, CharSet:=CharSet.Unicode, \_

ExactSpelling:=True, CallingConvention:=CallingConvention.StdCall)> \_

Public Shared Function StartPagePrinter(ByVal hPrinter As IntPtr) As Boolean

End Function

<DllImport("winspool.Drv", EntryPoint:="EndPagePrinter", \_

SetLastError:=True, CharSet:=CharSet.Unicode, \_

ExactSpelling:=True, CallingConvention:=CallingConvention.StdCall)> \_

Public Shared Function EndPagePrinter(ByVal hPrinter As IntPtr) As Boolean

End Function

<DllImport("winspool.Drv", EntryPoint:="WritePrinter", \_

SetLastError:=True, CharSet:=CharSet.Unicode, \_

ExactSpelling:=True, CallingConvention:=CallingConvention.StdCall)> \_

Public Shared Function WritePrinter(ByVal hPrinter As IntPtr, ByVal pBytes As IntPtr, ByVal dwCount As Int32, ByRef dwWritten As Int32) As Boolean

End Function

Public Shared Function SendBytesToPrinter(ByVal szPrinterName As String, ByVal pBytes As IntPtr, ByVal dwCount As Int32) As Boolean

Dim hPrinter As IntPtr

Dim dwError As Int32

Dim di As DOCINFOW

Dim dwWritten As Int32

Dim bSuccess As Boolean

With di

.pDocName = "Toko Komputer"

.pDataType = "RAW"

End With

bSuccess = False

If OpenPrinter(szPrinterName, hPrinter, 0) Then

If StartDocPrinter(hPrinter, 1, di) Then

If StartPagePrinter(hPrinter) Then

bSuccess = WritePrinter(hPrinter, pBytes, dwCount, dwWritten)

EndPagePrinter(hPrinter)

End If

EndDocPrinter(hPrinter)

End If

ClosePrinter(hPrinter)

End If

If bSuccess = False Then

dwError = Marshal.GetLastWin32Error()

End If

Return bSuccess

End Function

Public Shared Function SendFileToPrinter(ByVal szPrinterName As String, ByVal szFileName As String) As Boolean

Dim fs As New FileStream(szFileName, FileMode.Open)

Dim br As New BinaryReader(fs)

Dim bytes(fs.Length) As Byte

Dim bSuccess As Boolean

Dim pUnmanagedBytes As IntPtr

bytes = br.ReadBytes(fs.Length)

pUnmanagedBytes = Marshal.AllocCoTaskMem(fs.Length)

Marshal.Copy(bytes, 0, pUnmanagedBytes, fs.Length)

bSuccess = SendBytesToPrinter(szPrinterName, pUnmanagedBytes, fs.Length)

Marshal.FreeCoTaskMem(pUnmanagedBytes)

Return bSuccess

End Function

Public Shared Function SendStringToPrinter(ByVal szPrinterName As String, ByVal szString As String)

Dim pBytes As IntPtr

Dim dwCount As Int32

dwCount = szString.Length()

pBytes = Marshal.StringToCoTaskMemAnsi(szString)

SendBytesToPrinter(szPrinterName, pBytes, dwCount)

Marshal.FreeCoTaskMem(pBytes)

End Function

End Class