

DAFTAR PUSTAKA

- bin Ladjamudin, Al Bahra. *Analisa dan Desain Sistem Informasi*. Yogyakarta: Graha Ilmu, 2005
- Dennis Alan, Barbara Haley Wixom and David Tegarden. *Systems Analysis and Design with UML Version 2.0 an Object Oriented Approach second edition*. US Amerika: John Wiley & Sons, Inc, 2005
- Hasibuan, Malayu. *Dasar – Dasar Perbankan*. Jakarta: Bumi Aksara, 2004
- Jogiyanto HM,MBA,Akt,PhD. *Analisis & Desain Sistem Informasi* edisi ketiga. Yogyakarta: ANDI, 2006
- Kahate, Atul. *Introduction to Database Management System*.
<http://books.google.co.id/books>
- Kasmir. *Manajemen Perbankan*. Jakarta: PT. Raja Grafindo Persada, 2003
- McLeod Jr, Raymond. *Sistem Informasi Manajemen Jilid 1* edisi ketujuh, Alih bahasa: Hendra Teguh, SE.AK. Jakarta: Prenhalindo, 2001
- Munawar. *Pemodelan Visual dengan UML*. Jakarta: Graha Ilmu, 2005
- Muyono, Teguh Pudjo. *Manajemen Perkreditan Bagi Perbankan Komersil*. Yogyakarta: BPFE, 2007
- Rahmat, Firdaus. *Manajemen Prekreditan Bank Umum*. Bandung: Alfabeta, 2004
- Wahana Komputer. *Pemrograman Visual Basic.NET 2005*. Yogyakarta: Penerbit Andi, 2006
- Yuswanto. *Pemrograman Dasar Visual Basic.NET 2005*. Jakarta: Cerdas Pustaka Publisher, 2008
- Sholih. *Pemodelan Sistem Informasi Berorientasi Objek dengan UML*. Jakarta: Graha Ilmu, 2006
- <http://jenzprivate.blogspot.com/2009/02/perkembangan-visual-basicnet.html> (diakses tanggal 16 Oktober 2012)

LAMPIRAN

Form Login

Public Class Login

```
Private Sub  
frmLogin_Load(ByVal sender As  
System.Object, ByVal e As  
System.EventArgs) Handles  
MyBase.Shown  
    tUserName.Text = ""  
    tPassword.Text = ""  
    tUserName.Focus()  
End Sub
```

Sub EnableMenu()

MainMenu.MasterParentMenu.Enabled = True

MainMenu.TransactionParentMenu.Enabled = True

MainMenu.ReportParentMenu.Enabled = True

MainMenu.ToolParentMenu.Enabled = True

MainMenu.CustomerChildMenu.Visible = privilege_customer

MainMenu.CompanyChildMenu.Visible = privilege_company

MainMenu.EmployeeChildMenu.Visible = privilege_employee

MainMenu.PositionChildMenu.Visible = privilege_position

MainMenu.PlafondKreditChildMenu.Visible =
privilege_plafondkredit

MainMenu.ApprovedPlafondChildMenu.Visible =
privilege_approvedplafond

MainMenu.BreakPlafondChildMenu.

Visible =
privilege_breakplafond

MainMenu.PaymentPlafondChildMenu.Visible =
privilege_paymentplafond

MainMenu.ReportSisaPlafondChildMenu.Visible =
privilege_reportsisaplafond

MainMenu.ReportTransaksiKreditChildMenu.Visible =
privilege_reporttransaksikredit

MainMenu.ReportBreakPlafondChildMenu.Visible =
privilege_reportbreakplafond

MainMenu.AdminChildMenu.Visible = privilege_menuadmin
End Sub

```
Private Sub  
tPassword_KeyDown(ByVal sender  
As Object, ByVal e As  
System.Windows.Forms.KeyEventArgs) Handles tPassword.KeyDown  
    If e.KeyCode =  
Keys.Enter Then  
LoginButton_Click(sender, e)  
End Sub
```

```
Private Sub  
tUserName_KeyDown(ByVal sender  
As Object, ByVal e As  
System.Windows.Forms.KeyEventArgs) Handles tUserName.KeyDown  
    If e.KeyCode =  
Keys.Enter Then  
If tPassword.Text  
<> "" Then
```

LoginButton_Click(sender, e)
Else

```
tPassword.Focus()  
End If  
End If  
End Sub
```

```
Private Sub  
tUserName_TextChanged(ByVal  
sender As System.Object, ByVal
```

```

e As System.EventArgs) Handles
tUserName.TextChanged

    End Sub

    Private Sub
CancelButton_Click(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles
CancelButton.Click
        If MsgBox("Exit
Application?",
MsgBoxStyle.YesNo + vbQuestion,
saysText) = MsgBoxResult.Yes
Then
            If
db.connectDatabase Then
                db.disconnect()
            End If
            Close()
        End
        End If
    End Sub

    Private Sub
LoginButton_Click(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles
LoginButton.Click
        Dim dataReader As
SqlClient.SqlDataReader
        Try
            With db
                If
.connectDatabase Then
                    Dim
userName As String =
tUserName.Text.Trim()
                    Dim
password As String =
tPassword.Text.Trim()
                    dataReader
= db.dataReader(" SELECT * FROM
m_user mu WHERE mu.username =
'" & userName & "' AND
mu.[password] = '" & password &
"'")
                Try
                    If
dataReader.Read() Then
                        privilege_company =
dataReader("company")

                        privilege_customer =
dataReader("customer")

                        privilege_employee =
dataReader("employee")

                        privilege_position =
dataReader("position")

                        privilege_plafondkredit =
dataReader("plafond_kredit")

                        privilege_approvedplafond =
dataReader("approved_plafond")

                        privilege_breakplafond =
dataReader("break_plafond")

                        privilege_paymentplafond =
dataReader("payment_plafond")

                        privilege_reportsisaplafond =
dataReader("report_sisaplafond"
)

                        privilege_reporttransaksikredit
=
dataReader("report_transaksikre
dit")

                        privilege_reportbreakplafond =
dataReader("report_breakplafond
")

                        privilege_plafondmonitoring =
dataReader("monitoring")

                        privilege_menuadmin =
dataReader("menu_admin")

                        user_name =
dataReader("username")

                        user_id = dataReader("user_id")

                        employee_id =
dataReader("employee_id")

                        MainMenu.StatusUserLabel.Text =
user_name

                        EnableMenu()

```

```

logged = True
Close()
Else
logged = False
MsgBox("Invalid user name or
password.", 16, saysText)
tUserName.Text = ""
tPassword.Text = ""
tUserName.Focus()

If
tUserName.Text = "" Then
tUserName.Focus()
Else
tPassword.Focus()
tPassword.SelectionStart = 0
tPassword.SelectionLength =
tPassword.Text.Length
End If
Catch ex As
Exception
logged
= False
MsgBox("Invalid user name or
password.", 16, saysText)
tPassword.Focus()
tPassword.SelectionStart = 0
tPassword.SelectionLength =
tPassword.Text.Length
Finally
dataReader.Close()
End Try
Else
MsgBox("Failed connecting to
database server.", 16,
saysText)
Close()
End If
End With
Catch ex As Exception
MsgBox(ex.Message)
End Try
End Sub
End Class

Form Menu Utama
Public Class MainMenu
Private Sub
MainMenu_Load(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
MyBase.Load
DisableMenu()
Login.ShowDialog()
'If Not logged Then
,
LoginChildMenu.Enabled = True
,
LogoutChildMenu.Enabled = False
'Else
LogoutChildMenu.Enabled = True
,
LoginChildMenu.Enabled = False
'End If
End Sub
Sub DisableMenu()
Me.MasterParentMenu.Enabled =
False
Me.TransactionParentMenu.Enable
d = False
Me.ReportParentMenu.Enabled =
False
Me.ToolParentMenu.Enabled =
False
End Sub

```

```

Private Sub
ExitChildMenu_Click(ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
ExitChildMenu.Click
    If MsgBox("Exit
Application?",
MsgBoxStyle.YesNo + vbQuestion,
saysText) = MsgBoxResult.Yes
Then
    If
db.connectDatabase Then
        db.disconnect()
    End If
    Close()
End
End If
End Sub

```

```

Private Sub
AdminChildMenu_Click(ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
AdminChildMenu.Click
    If
(privilege_menuadmin) Then
        Admin.MdiParent =
Me
        Admin.Dock =
DockStyle.Fill
        Admin.Show()
        Admin.Focus()
    Else
        MsgBox("No
authorized to access this
module.", 16, saysText)
    End If
End Sub

```

```

Private Sub
ChangePasswordChildMenu_Click(B
yVal sender As System.Object,
ByVal e As System.EventArgs)
Handles
ChangePasswordChildMenu.Click
    ChangePassword.MdiParent = Me
    ChangePassword.Dock =
DockStyle.Fill
    ChangePassword.Focus()
    ChangePassword.Show()
End Sub

```

```

Private Sub
CustomerChildMenu_Click(ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
CustomerChildMenu.Click
    If (privilege_customer)
Then
        Customer.MdiParent
= Me
        Customer.Dock =
DockStyle.Fill
        Customer.Show()
        Customer.Focus()
    Else
        MsgBox("No
authorized to access this
module.", 16, saysText)
    End If
End Sub

```

```

Private Sub
CompanyChildMenu_Click(ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
CompanyChildMenu.Click
    If (privilege_company)
Then
        Company.MdiParent =
Me
        Company.Dock =
DockStyle.Fill
        Company.Show()
        Company.Focus()
    Else
        MsgBox("No
authorized to access this
module.", 16, saysText)
    End If
End Sub

```

```

Private Sub
EmployeeChildMenu_Click(ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
EmployeeChildMenu.Click
    If (privilege_employee)
Then
        Employee.MdiParent
= Me
        Employee.Dock =
DockStyle.Fill
        Employee.Show()
        Employee.Focus()
    Else

```

```

        MsgBox("No
authorized to access this
module.", 16, saysText)
    End If
End Sub

Private Sub
PositionChildMenu_Click(ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
PositionChildMenu.Click
    If (privilege_position)
Then
        Position.MdiParent
= Me
        Position.Dock =
DockStyle.Fill
        Position.Show()
        Position.Focus()
    Else
        MsgBox("No
authorized to access this
module.", 16, saysText)
    End If
End Sub

Private Sub
PlafondKreditChildMenu_Click(By
Val sender As System.Object,
ByVal e As System.EventArgs)
    If
(privilege_plafondkredit) Then
PlafondKredit.MdiParent = Me
        PlafondKredit.Dock
= DockStyle.Fill
PlafondKredit.Show()
PlafondKredit.Focus()
    Else
        MsgBox("No
authorized to access this
module.", 16, saysText)
    End If
End Sub

Private Sub
ApprovedPlafondChildMenu_Click(
ByVal sender As System.Object,
ByVal e As System.EventArgs)
Handles
ApprovedPlafondChildMenu.Click

```

```

    If
(privilege_approvedplafond)
Then
        ApprovedPlafond.MdiParent = Me
        ApprovedPlafond.Dock =
DockStyle.Fill
        ApprovedPlafond.Show()
        ApprovedPlafond.Focus()
    Else
        MsgBox("No
authorized to access this
module.", 16, saysText)
    End If
End Sub

Private Sub
PaymentPlafondChildMenu_Click(B
yVal sender As System.Object,
ByVal e As System.EventArgs)
Handles
PaymentPlafondChildMenu.Click
    If
(privilege_paymentplafond) Then
PaymentPlafond.MdiParent = Me
        PaymentPlafond.Dock
= DockStyle.Fill
        PaymentPlafond.Show()
        PaymentPlafond.Focus()
    Else
        MsgBox("No
authorized to access this
module.", 16, saysText)
    End If
End Sub

Private Sub
BreakPlafondChildMenu_Click(ByV
al sender As System.Object,
ByVal e As System.EventArgs)
Handles
BreakPlafondChildMenu.Click
    If
(privilege_breakplafond) Then
PlafondBreak.MdiParent = Me
        PlafondBreak.Dock =
DockStyle.Fill

```

```

        PlafondBreak.Show()
PlafondBreak.Focus()
    Else
        MsgBox("No
authorized to access this
module.", 16, saysText)
    End If
End Sub

Private Sub
PlafondKreditChildMenu_Click_1(
ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles
PlafondKreditChildMenu.Click
    If
(privilege_plafondkredit) Then
PlafondKredit.MdiParent = Me
        PlafondKredit.Dock
= DockStyle.Fill

PlafondKredit.Show()

PlafondKredit.Focus()
    Else
        MsgBox("No
authorized to access this
module.", 16, saysText)
    End If
End Sub

Private Sub
ReportBreakPlafondChildMenu_Cli
ck(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
ReportBreakPlafondChildMenu.Cli
ck
    If
(privilege_reportbreakplafond)
Then
ReportBreakPlafond.MdiParent =
Me

ReportBreakPlafond.Dock =
DockStyle.Fill

ReportBreakPlafond.Show()

ReportBreakPlafond.Focus()
    Else

```

```

        MsgBox("No
authorized to access this
module.", 16, saysText)
    End If
End Sub

Private Sub
ReportTransaksiKreditChildMenu_
Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
ReportTransaksiKreditChildMenu.
Click
    If
(privilege_reporttransaksikredi
t) Then
ReportPaymentPlafond.MdiParent
= Me

ReportPaymentPlafond.Dock =
DockStyle.Fill

ReportPaymentPlafond.Show()

ReportPaymentPlafond.Focus()
    Else
        MsgBox("No
authorized to access this
module.", 16, saysText)
    End If
End Sub

Private Sub
ReportSisaPlafondChildMenu_Clic
k(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
ReportSisaPlafondChildMenu.Clic
k
    If
(privilege_plafondmonitoring)
Then
PlafondMonitoring.MdiParent =
Me

PlafondMonitoring.Dock =
DockStyle.Fill

PlafondMonitoring.Show()

PlafondMonitoring.Focus()
    Else

```

```

        MsgBox("No
authorized to access this
module.", 16, saysText)
    End If
End Sub
End Class

```

Form Company

```
Imports System.Text
```

```
Public Class Company
```

```

    Private isEdited As Boolean
    = False
    Private ds2 As DataSet

```

```

    Private Sub
Company_Load(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
MyBase.Load
        initData()
    End Sub

```

```

    Sub initData()
        Text = "Insert New
Company"
        isEdited = False
        txtCompanyID.Text =
"[auto_generate]"
        txtCompanyName.Text =
""
        txtCompanyAddress.Text
= ""
        txtCompanyPhone.Text =
""
        txtCompanyFax.Text = ""
        txtCompanyEmail.Text =
""
        txtCompanyWebsite.Text
= ""
        txtCompanyCity.Text =
""
        txtCompanyCountry.Text
= ""
        txtCompanyName.Focus()
    End Sub

```

```

    WriteOnly Property
ds2Call() As DataSet
        Set(ByVal value As
DataSet)
            ds2 = value

```

```

        End Set
    End Property

```

```

    Private Sub
cNew_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnNew.Click
        initData()
    End Sub

```

```

    Private Sub
Delete_Company(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnDelete.Click
        If isEdited Then
            Dim companyID As
String = txtCompanyID.Text

```

```

                If MsgBox("Delete
This Company?",
MsgBoxStyle.YesNo + vbQuestion,
saysText) = MsgBoxResult.Yes
Then

```

```

                    If
db.connectDatabase Then
                        Dim
sqlString As String = "SELECT *
FROM m_company mc JOIN
m_customer mc2 ON
mc2.company_id = mc.company_id
WHERE mc.company_id = ' " &
companyID & "' "

```

```

                        Dim
dataReader As
SqlClient.SqlDataReader =
db.dataReader(sqlString)
                        If
dataReader.HasRows() Then
                            MsgBox("Cannot Delete Company
Which Still Has Customers!",
MsgBoxStyle.Critical, saysText)

```

```

                            dataReader.Close()
                        Else

```

```

                            dataReader.Close()
                            Dim
result As Integer =
db.executeNonQuery("DELETE FROM
m_company WHERE company_id = ' "
& companyID & "'")

```



```

                                If
result > 0 Then
    MsgBox(companyID + " Has Been
    Sucessfully Deleted.",
    MsgBoxStyle.Information,
    saysText)
    initData()
                                End If
                                End If
                                Else
                                MsgBox("Failed connecting to
                                database server.", 16,
                                saysText)
                                End If
                                End If
                                Else
                                MsgBox("Cannot
                                Delete Empty Customer!",
                                MsgBoxStyle.Critical, saysText)
                                End If
                                End Sub

                                Private Sub
                                Load_Companies(ByVal sender As
                                System.Object, ByVal e As
                                System.EventArgs) Handles
                                cDocument.Click
                                    Search_CompanyList()
                                End Sub

                                Sub Search_CompanyList()
                                    Dim qs As New
                                    QuerySelection
                                    With qs
                                        qs.sqlMain = "
                                        mc.company_id, mc.company_name,
                                        mc.[address], mc.city,
                                        mc.country, mc.phone, mc.fax,
                                        mc.email, mc.website FROM
                                        m_company mc "
                                        qs.sqlOrderBy = "
                                        mc.company_id DESC "
                                    End With
                                    Dim h As New
                                    Browser(qs, Me, "ds2Call",
                                    "Company List")
                                    Text = "Edit Existing
                                    Company"
                                    If h.ShowDialog() =
                                    Windows.Forms.DialogResult.Yes
                                    Then
                                        txtCompanyID.Text =
                                        ds2.Tables(0).Rows(0).Item(0)
                                        txtCompanyName.Text
                                        = ds2.Tables(0).Rows(0).Item(1)
                                        txtCompanyAddress.Text =
                                        ds2.Tables(0).Rows(0).Item(2)
                                        txtCompanyCity.Text
                                        = ds2.Tables(0).Rows(0).Item(3)
                                        txtCompanyCountry.Text =
                                        ds2.Tables(0).Rows(0).Item(4)
                                        txtCompanyPhone.Text =
                                        ds2.Tables(0).Rows(0).Item(5)
                                        txtCompanyFax.Text
                                        = ds2.Tables(0).Rows(0).Item(6)
                                        txtCompanyEmail.Text =
                                        ds2.Tables(0).Rows(0).Item(7)
                                        txtCompanyWebsite.Text =
                                        ds2.Tables(0).Rows(0).Item(8)
                                        isEdited = True
                                    End If
                                End Sub

                                Private Sub
                                InsertUpdate_Company(ByVal
                                sender As System.Object, ByVal
                                e As System.EventArgs) Handles
                                btnSave.Click
                                    If txtCompanyName.Text
                                    = "" Then
                                        MsgBox(fillText &
                                        "Name", MsgBoxStyle.Critical,
                                        saysText)
                                        txtCompanyName.Focus()
                                    ElseIf
                                        txtCompanyAddress.Text = ""
                                    Then
                                        MsgBox(fillText &
                                        "Address",
                                        MsgBoxStyle.Critical, saysText)
                                        txtCompanyAddress.Focus()
                                    ElseIf
                                        txtCompanyPhone.Text = "" Then
                                        MsgBox(fillText &
                                        "Phone", MsgBoxStyle.Critical,
                                        saysText)

```

```

txtCompanyPhone.Focus()
    ElseIf
txtCompanyFax.Text = "" Then
    MsgBox(fillText &
"Fax", MsgBoxStyle.Critical,
saysText)

txtCompanyFax.Focus()
    ElseIf
txtCompanyEmail.Text = "" Then
    MsgBox(fillText &
"Email", MsgBoxStyle.Critical,
saysText)

txtCompanyEmail.Focus()
    ElseIf
txtCompanyCity.Text = "" Then
    MsgBox(fillText &
"City", MsgBoxStyle.Critical,
saysText)

txtCompanyCity.Focus()
    ElseIf
txtCompanyCountry.Text = ""
Then
    MsgBox(fillText &
"Country",
MsgBoxStyle.Critical, saysText)

txtCompanyCountry.Focus()
    Else
Integer
    Dim result As
    If Not isEdited
Then
txtCompanyID.Text =
db.getAutoNo("COM", True)
    Dim strInsert
As StringBuilder = New
StringBuilder

strInsert.Append(" INSERT INTO
m_company ")

strInsert.AppendFormat(" VALUES
('{0}', '{1}', '{2}', '{3}',
'{4}', '{5}', '{6}', '{7}',
'{8}')" , txtCompanyID.Text,
txtCompanyName.Text,
txtCompanyAddress.Text,
txtCompanyCity.Text,
txtCompanyCountry.Text,
txtCompanyPhone.Text,
txtCompanyFax.Text,
txtCompanyEmail.Text,
txtCompanyWebsite.Text)

strUpdate.AppendFormat(" SET
company_name = '{0}', [address]
= '{1}', city = '{2}', country
= '{3}', phone = '{4}', fax =
'{5}', email = '{6}', website =
'{7}'" , txtCompanyName.Text,
txtCompanyAddress.Text,
txtCompanyCity.Text,
txtCompanyCountry.Text,
txtCompanyPhone.Text,
txtCompanyFax.Text,
txtCompanyEmail.Text,
txtCompanyWebsite.Text)

strUpdate.AppendFormat(" WHERE
company_id = '{0}'" ,
txtCompanyID.Text)
    result =
db.executeNonQuery(strUpdate.To
String())
    End If

    If result > 0 Then
    If Not isEdited
Then
MsgBox("Data Has Been
Sucessfully Inserted.",
MsgBoxStyle.Information,
saysText)
    Else
MsgBox("Data Has Been
Sucessfully Updated.",
MsgBoxStyle.Information,
saysText)
    End If
    initData()

```

```

        End If
    End If
End Sub

Private Sub
cDocument_KeyUp(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles cDocument.KeyUp
    If e.KeyCode =
Keys.Enter Then

Search_CompanyList()
    End If
End Sub

Private Sub
txtCompanyName_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
txtCompanyName.KeyUp
    If e.KeyCode =
Keys.Enter Then

txtCompanyAddress.Focus()
    End If
End Sub

Private Sub
txtCompanyAddress_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
txtCompanyAddress.KeyUp
    If e.KeyCode =
Keys.Enter Then

txtCompanyPhone.Focus()
    End If
End Sub

Private Sub
txtCompanyPhone_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
txtCompanyPhone.KeyUp
    If e.KeyCode =
Keys.Enter Then

txtCompanyFax.Focus()
    End If
End Sub

Private Sub
txtCompanyFax_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles txtCompanyFax.KeyUp
    If e.KeyCode =
Keys.Enter Then

txtCompanyEmail.Focus()
    End If
End Sub

Private Sub
txtCompanyEmail_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
txtCompanyEmail.KeyUp
    If e.KeyCode =
Keys.Enter Then

txtCompanyWebsite.Focus()
    End If
End Sub

Private Sub
txtCompanyWebsite_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
txtCompanyWebsite.KeyUp
    If e.KeyCode =
Keys.Enter Then

txtCompanyCity.Focus()
    End If
End Sub

Private Sub
txtCompanyCity_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
txtCompanyCity.KeyUp
    If e.KeyCode =
Keys.Enter Then

txtCompanyCountry.Focus()
    End If
End Sub
End Class

```

Form Customer

```
Imports System.Text
Imports System.Windows.Forms
Imports System.Data.SqlClient

Public Class Customer
    Private isEdited As Boolean
    = False
    Private companyID As String
    = Nothing
    Private ds2 As DataSet

    Private Sub
Customer_Load(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
MyBase.Shown
        initData()
    End Sub

    Sub initData()
        Text = "Insert New
Customer"
        isEdited = False
        companyID = Nothing
        txt_customerid.Text =
"[auto_generate]"
        txt_customername.Text =
""
        txtCompanyName.Text =
""
        cmbType.Text = ""
        txt_address.Text = ""
        txt_phone.Text = ""
        txt_email.Text = ""
        txt_city.Text = ""
        txt_country.Text = ""
        btnLoadCompany.Enabled
= True

        txt_customername.Focus()
    End Sub

    WriteOnly Property
ds2Call() As DataSet
        Set(ByVal value As
DataSet)
            ds2 = value
        End Set
    End Property

    Private Sub
Load_Customers(ByVal sender As
```

```
System.Object, ByVal e As
System.EventArgs) Handles
CustomerFindButton.Click
        Search_CustomerList()
    End Sub

    Sub Search_CustomerList()
        Dim qs As New
QuerySelection
        With qs
            qs.sqlMain = "
mc.customer_id, mc.name,
mc.[address], mc.[type],
mc.phone, mc.email, mc.city,
mc.country, mc.company_id,
mc2.company_name " & _
" FROM
m_customer mc JOIN m_company
mc2 ON mc2.company_id =
mc.company_id "
            qs.sqlOrderBy = "
mc.customer_id DESC "
        End With
        Dim h As New
Browser(qs, Me, "ds2Call",
"Customer List")
        Text = "Edit Existing
Customer"
        If h.ShowDialog() =
Windows.Forms.DialogResult.Yes
Then
            txt_customerid.Text
= ds2.Tables(0).Rows(0).Item(0)
            txt_customername.Text =
ds2.Tables(0).Rows(0).Item(1)
            txt_address.Text =
ds2.Tables(0).Rows(0).Item(2)
            cmbType.Text =
ds2.Tables(0).Rows(0).Item(3)
            txt_phone.Text =
ds2.Tables(0).Rows(0).Item(4)
            txt_email.Text =
ds2.Tables(0).Rows(0).Item(5)
            txt_city.Text =
ds2.Tables(0).Rows(0).Item(6)
            txt_country.Text =
ds2.Tables(0).Rows(0).Item(7)
            txtCompanyName.Text
= ds2.Tables(0).Rows(0).Item(8)
            companyID =
ds2.Tables(0).Rows(0).Item(8)
            btnLoadCompany.Enabled = False
```

```

        isEdited = True
    End If
End Sub

Private Sub
InsertUpdate_Customer(ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
btnSave.Click
    If
txt_customername.Text = "" Then
        MsgBox(fillText &
"Name", MsgBoxStyle.Critical,
saysText)
txt_customername.Focus()
    ElseIf
txtCompanyName.Text = "" Then
        MsgBox(fillText &
"Company",
MsgBoxStyle.Critical, saysText)
txtCompanyName.Focus()
    ElseIf cmbType.Text =
"" Then
        MsgBox(chooseText &
"Type", MsgBoxStyle.Critical,
saysText)
        txt_address.Focus()
    ElseIf txt_address.Text
= "" Then
        MsgBox(fillText &
"Address",
MsgBoxStyle.Critical, saysText)
        txt_address.Focus()
    ElseIf txt_phone.Text =
"" Then
        MsgBox(fillText &
"Phone", MsgBoxStyle.Critical,
saysText)
        txt_city.Focus()
    ElseIf txt_email.Text =
"" Then
        MsgBox(fillText &
"Email", MsgBoxStyle.Critical,
saysText)
        txt_email.Focus()
    ElseIf txt_city.Text =
"" Then
        MsgBox(fillText &
"City", MsgBoxStyle.Critical,
saysText)
        txt_city.Focus()
    ElseIf txt_country.Text
= "" Then
        MsgBox(fillText &
"Country",
MsgBoxStyle.Critical, saysText)
        txt_country.Focus()
    Else
        Dim result As
Integer
        Dim strBuilder As
StringBuilder
        Dim sqlString As
String = Nothing, customerID As
String =
txt_customerid.Text.Trim(),
country As String =
txt_country.Text.Trim()
        Dim customerName As
String =
txt_customername.Text.Trim(),
type As String =
cmbType.Text.Trim(), address As
String =
txt_address.Text.Trim()
        Dim phone As String
= txt_phone.Text.Trim(), email
As String =
txt_email.Text.Trim(), city As
String = txt_city.Text.Trim()

        If Not isEdited
Then
            customerID =
db.getAutoNo("CUS", True)
            txt_customerid.Text =
customerID
            strBuilder =
New StringBuilder
            strBuilder.Append(" INSERT INTO
m_customer VALUES (")
            strBuilder.AppendFormat("
'{0}', ", customerID)
            strBuilder.AppendFormat("
'{0}', ", companyID)
            strBuilder.AppendFormat("
'{0}', ", customerName)
            strBuilder.AppendFormat("
'{0}', ", type)

```

```

strBuilder.AppendFormat("
'{0}', ", address)

strBuilder.AppendFormat("
'{0}', ", phone)

strBuilder.AppendFormat("
'{0}', ", email)

strBuilder.AppendFormat("
'{0}', ", city)

strBuilder.AppendFormat("
'{0}') ", country)
sqlString =
strBuilder.ToString()
result =
db.executeNonQuery(sqlString)
Else
strBuilder =
New StringBuilder

strBuilder.Append(" UPDATE
m_customer SET ")

strBuilder.AppendFormat("
company_id = '{0}', ",
companyID)

strBuilder.AppendFormat(" name
= '{0}', ", customerName)

strBuilder.AppendFormat("
[type] = '{0}', ", type)

strBuilder.AppendFormat("
[address] = '{0}', ", address)

strBuilder.AppendFormat(" phone
= '{0}', ", phone)

strBuilder.AppendFormat(" email
= '{0}', ", email)

strBuilder.AppendFormat(" city
= '{0}', ", city)

strBuilder.AppendFormat("
country = '{0}' ", country)

strBuilder.AppendFormat(" WHERE
customer_id = '{0}'",
customerID)

sqlString =
strBuilder.ToString()
result =
db.executeNonQuery(sqlString)
End If

If result > 0 Then
If Not isEdited
Then
MsgBox("Data Has Been
Sucessfully Inserted.",
MsgBoxStyle.Information,
saysText)
Else
MsgBox("Data Has Been
Sucessfully Updated.",
MsgBoxStyle.Information,
saysText)
End If
initData()
End If
End If
End Sub

Private Sub
Delete_Customer(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnDelete.Click
If isEdited Then
Dim customerID As
String = txt_customerid.Text

If MsgBox("Delete
This Customers?",
MsgBoxStyle.YesNo + vbQuestion,
saysText) = MsgBoxResult.Yes
Then
If
db.connectDatabase Then
Dim
dataReader As SqlDataReader =
db.dataReader("SELECT * FROM
t_payment_plafond_detail tppd
WHERE tppd.customer_id = '" &
customerID & "'")
If
dataReader.HasRows() Then
MsgBox("Cannot Delete This
Customer, Because It Has

```

```

Reference To PaymentDetail!",
MsgBoxStyle.Critical, saysText)

dataReader.Close()
Else
dataReader.Close()
Dim
result As Integer =
db.executeNonQuery("DELETE FROM
m_customer WHERE customer_id =
'" & customerID & "'")
If
result > 0 Then
MsgBox(customerID + " Has Been
Sucessfully Deleted.",
MsgBoxStyle.Information,
saysText)
initData()
End If
Else
MsgBox("Failed connecting to
database server.", 16,
saysText)
End If
Else
MsgBox("Cannot
Delete Empty Customer!",
MsgBoxStyle.Critical, saysText)
End If
End Sub

Private Sub
cmdNew_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnNew.Click
initData()
End Sub

Private Sub
btnLoadCompany_Click(ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
btnLoadCompany.Click
Search_CustomerList()
End Sub

Sub Search_CompanyList()
companyID = Nothing
Dim qs As New
QuerySelection
With qs
qs.sqlMain = "
mc.company_id, mc.company_name,
mc.[address], mc.city,
mc.country, mc.phone, mc.fax,
mc.email, mc.website FROM
m_company mc "
qs.sqlOrderBy = "
mc.company_id DESC "
End With
Dim h As New
Browser(qs, Me, "ds2Call",
"Company List")
If h.ShowDialog() =
Windows.Forms.DialogResult.Yes
Then
companyID =
ds2.Tables(0).Rows(0).Item(0)
txtCompanyName.Text
= ds2.Tables(0).Rows(0).Item(1)
End If
End Sub

Private Sub
CustomerFindButton_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
CustomerFindButton.KeyUp
If e.KeyCode =
Keys.Enter Then
Search_CustomerList()
End If
End Sub

Private Sub
btnLoadCompany_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
btnLoadCompany.KeyUp
If e.KeyCode =
Keys.Enter Then
Search_CompanyList()
End If
End Sub

Private Sub
txt_customername_KeyUp(ByVal

```



```

sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
txt_customername.KeyUp
    If e.KeyCode =
Keys.Enter Then
        txt_address.Focus()
    End If
End Sub

Private Sub
txt_address_KeyUp(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles txt_address.KeyUp
    If e.KeyCode =
Keys.Enter Then
        txt_phone.Focus()
    End If
End Sub

Private Sub
txt_phone_KeyUp(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles txt_phone.KeyUp
    If e.KeyCode =
Keys.Enter Then
        txt_email.Focus()
    End If
End Sub

Private Sub
txt_email_KeyUp(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles txt_email.KeyUp
    If e.KeyCode =
Keys.Enter Then
        txt_city.Focus()
    End If
End Sub

Private Sub
txt_city_KeyUp(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles txt_city.KeyUp
    If e.KeyCode =
Keys.Enter Then
        txt_country.Focus()
    End If
End Sub
End Class

```

Form Employee

```

Imports System.Text
Imports System.Data.SqlClient

Public Class Employee
    Private isEdited As Boolean
    = False
    Private ds2 As DataSet
    Private userName As String
    = Nothing

    Private Sub
Employee_Load(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
MyBase.Shown
        initData()
    End Sub

    Sub initData()
        Text = "Insert New
Employee"
        isEdited = False
        txtEmployeeID.Text =
"[auto_generate]"
        cmbPosition.DataSource
        = GetPositions()

        cmbPosition.DisplayMember =
"Name"
        cmbPosition.ValueMember
        = "ID"
        rbMale.Checked = True
        txtEmployeeName.Text =
""
        dtDateOfBirth.Text = ""
        txtEmployeeEmail.Text =
""
        txtEmployeeAddress.Text
        = ""
        txtEmployeePhone.Text =
""
        txtEmployeeCity.Text =
""
        txtEmployeeCountry.Text
        = ""
        txtEmployeeName.Focus()
    End Sub

    WriteOnly Property
ds2Call() As DataSet

```



```

        Set(ByVal value As
DataSet)
            ds2 = value
        End Set
    End Property

    Private Sub
Load_Employees(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
cDocument.Click
        Search_EmployeeList()
    End Sub

    Sub Search_EmployeeList()
        Dim qs As New
QuerySelection
        With qs
            qs.sqlMain = "
me.employee_id AS ID, me.name
AS Name, me.sex AS Gender,
me.[address] AS Address,
me.dateofbirth AS DateOfBirth,
" & _
            "
me.city AS City, me.country AS
Country, me.phone AS Phone,
me.email AS Email, mp.name AS
Position, mp.position_id AS
PositionID, mu.username AS
UserName " & _
            " FROM
m_employee me JOIN m_position
mp ON mp.position_id =
me.position_id " & _
            " JOIN
m_user mu ON mu.employee_id =
me.employee_id "
            qs.sqlOrderBy = "
me.employee_id DESC "
        End With
        Dim h As New
Browser(qs, Me, "ds2Call",
"Employee List")
        Text = "Edit Existing
Employee"
        If h.ShowDialog() =
Windows.Forms.DialogResult.Yes
Then
            txtEmployeeID.Text
=
ds2.Tables(0).Rows(0).Item(0).T
oString().Trim()
            txtEmployeeName.Text =
ds2.Tables(0).Rows(0).Item(1).T
oString().Trim()
            Dim gender As
String =
ds2.Tables(0).Rows(0).Item(2).T
oString().Trim()
            If gender.ToUpper()
= "MALE" Then
                rbMale.Checked
= True
            Else
                rbFemale.Checked = True
            End If
            txtEmployeeAddress.Text =
ds2.Tables(0).Rows(0).Item(3).T
oString().Trim()
            dtDateOfBirth.Text
= ds2.Tables(0).Rows(0).Item(4)
            txtEmployeeCity.Text =
ds2.Tables(0).Rows(0).Item(5).T
oString().Trim()
            txtEmployeeCountry.Text =
ds2.Tables(0).Rows(0).Item(6).T
oString().Trim()
            txtEmployeePhone.Text =
ds2.Tables(0).Rows(0).Item(7).T
oString().Trim()
            txtEmployeeEmail.Text =
ds2.Tables(0).Rows(0).Item(8).T
oString().Trim()
            cmbPosition.Text =
ds2.Tables(0).Rows(0).Item(9).T
oString().Trim()
            userName =
ds2.Tables(0).Rows(0).Item(11).
ToString().Trim()
            isEdited = True
        End If
    End Sub

    Function GetPositions() As
List(Of ComboBoxItem)

        Dim positions = New
List(Of ComboBoxItem)

```

```

        Dim dataReader As
        SqlDataReader =
        db.dataReader("SELECT
        mp.position_id, mp.name FROM
        m_position mp")

        If dataReader.HasRows =
        True Then
            While
            dataReader.Read()

            positions.Add(New
            ComboBoxItem(dataReader("positi
            on_id"), dataReader("name")))
            End While
        End If
        dataReader.Close()
        Return positions

    End Function

    Private Sub
    Delete_Employee(ByVal sender As
    System.Object, ByVal e As
    System.EventArgs) Handles
    btnDelete.Click
        If isEdited Then
            If Not
            userName.Trim.ToUpper().Equals(
            user_name.Trim.ToUpper())
            Then
                Dim employeeID
                As String = txtEmployeeID.Text

                If
                MsgBox("Delete This Employee
                Will Also Delete Its User
                Reference. Are You Sure?",
                MsgBoxStyle.YesNo + vbQuestion,
                saysText) = MsgBoxResult.Yes
                Then
                    If
                    db.connectDatabase Then
                        Dim
                        sqlString As String = " DELETE
                        FROM m_user WHERE username = '"
                        & userName & "'"

                        sqlString += " DELETE FROM
                        m_employee WHERE employee_id =
                        '" & employeeID & "'"

                        Dim
                        result As Integer =
                        db.executeNonQuery(sqlString)

                        If
                        result > 0 Then
                            MsgBox(employeeID + " Has Been
                            Sucessfully Deleted.",
                            MsgBoxStyle.Information,
                            saysText)

                            initData()
                        End If
                    Else
                        MsgBox("Failed connecting to
                        database server.",
                        MsgBoxStyle.Critical, saysText)
                    End If
                Else
                    MsgBox("Cannot
                    Delete Your Own Data!",
                    MsgBoxStyle.Critical, saysText)
                End If
            Else
                MsgBox("Cannot
                Delete Empty Customer!",
                MsgBoxStyle.Critical, saysText)
            End If
        End Sub

        Private Sub
        Save_Employee(ByVal sender As
        System.Object, ByVal e As
        System.EventArgs) Handles
        btnSave.Click
            If cmbPosition.Text =
            "" Then
                MsgBox(chooseText +
                "Jabatan",
                MsgBoxStyle.Critical, saysText)
                cmbPosition.Focus()
            ElseIf
            txtEmployeeName.Text = "" Then
                MsgBox(fillText +
                "Nama", MsgBoxStyle.Critical,
                saysText)

                txtEmployeeName.Focus()
            ElseIf
            dtDateOfBirth.Text = "" Then
                MsgBox(fillText +
                "Tanggal Lahir",
                MsgBoxStyle.Critical, saysText)

                dtDateOfBirth.Focus()
            End If
        End Sub
    End Sub

```

```

ElseIf
txtEmployeeAddress.Text = ""
Then
    MsgBox(fillText +
"Alamat", MsgBoxStyle.Critical,
saysText)

txtEmployeeAddress.Focus()
Else
    Dim strBuilder As
StringBuilder
    Dim sqlString As
String = ""
    Dim result As
Integer
    Dim
positionSelected As
ComboBoxItem =
cmbPosition.SelectedItem
    Dim userName1 As
String =
txtEmployeeName.Text.Trim().ToL
ower()
    Dim userName2 As
String =
txtEmployeeName.Text.Trim().ToL
ower()
    Dim birthDate As
String =
dtDateOfBirth.Text.Trim()
    Dim userSplit() As
String = userName1.Split("
".ToCharArray())
    If userSplit.Length
= 1 Then
        Dim birth() As
String =
birthDate.Split("/".ToCharArray
())
        birthDate =
birth(1) &
birth(2).Substring(2, 2)
        userName2 =
userName1 & birthDate
    ElseIf
userSplit.Length = 2 Then
        userName2 =
userName1.Replace(" ", ".")
    Else
        userName2 =
userSplit(0) & "." &
userSplit(1)
    End If

    Dim gender As
String = rbFemale.Text
    If rbMale.Checked
Then
        gender =
rbMale.Text
    End If

    If Not isEdited
Then
        txtEmployeeID.Text =
db.getAutoNo("EMP", True)
        strBuilder =
New StringBuilder
        strBuilder.Append(" INSERT INTO
m_employee VALUES (")
        strBuilder.AppendFormat("{0}', ", txtEmployeeID.Text)
        strBuilder.AppendFormat("{0}', ", txtEmployeeName.Text)
        strBuilder.AppendFormat("{0}', ", gender)
        strBuilder.AppendFormat("{0}', ",
txtEmployeeAddress.Text)
        strBuilder.AppendFormat("{0}', ",
dtDateOfBirth.Text)
        strBuilder.AppendFormat("{0}', ",
txtEmployeeCity.Text)
        strBuilder.AppendFormat("{0}', ",
txtEmployeeCountry.Text)
        strBuilder.AppendFormat("{0}', ",
txtEmployeePhone.Text)
        strBuilder.AppendFormat("{0}', ",
txtEmployeeEmail.Text)
        strBuilder.AppendFormat("{0}') ", positionSelected.ID)
        sqlString =
strBuilder.ToString()

```

```

        strBuilder =
New StringBuilder

strBuilder.Append(" INSERT INTO
m_user(username, [password],
employee_id) VALUES (")

strBuilder.AppendFormat("' {0}',
", userName2)

strBuilder.AppendFormat("' {0}',
", userName2)

strBuilder.AppendFormat("' {0}')"
", txtEmployeeID.Text)
sqlString +=
strBuilder.ToString()
result =
db.executeNonQueryTransaction(s
qlString)

Else
        strBuilder =
New StringBuilder

strBuilder.Append(" UPDATE
m_employee SET ")

strBuilder.AppendFormat(" name
= '{0}', ",
txtEmployeeName.Text)

strBuilder.AppendFormat(" sex =
'{0}', ", gender)

strBuilder.AppendFormat("
[address] = '{0}', ",
txtEmployeeAddress.Text)

strBuilder.AppendFormat("
dateofbirth = '{0}', ",
dtDateOfBirth.Text)

strBuilder.AppendFormat(" city
= '{0}', ",
txtEmployeeCity.Text)

strBuilder.AppendFormat("
country = '{0}', ",
txtEmployeeCountry.Text)

strBuilder.AppendFormat(" phone
= '{0}', ",
txtEmployeePhone.Text)

strBuilder.AppendFormat(" email
= '{0}', ",
txtEmployeeEmail.Text)

strBuilder.AppendFormat("
position_id = '{0}'",
positionSelected.ID)

strBuilder.AppendFormat(" WHERE
employee_id = '{0}'",
txtEmployeeID.Text)
sqlString =
strBuilder.ToString()

        strBuilder =
New StringBuilder

strBuilder.Append(" UPDATE
m_user SET ")

strBuilder.AppendFormat("
username = '{0}' ", userName2)

strBuilder.AppendFormat(" WHERE
username = '{0}' ", userName)
sqlString +=
strBuilder.ToString()
result =
db.executeNonQueryTransaction(s
qlString)
End If

If result > 0 Then
    If Not isEdited
Then
        MsgBox("Data Has Been
Sucessfully Inserted.",
MsgBoxStyle.Information,
saysText)

        Dim message
As String = "Your UserName And
Password Is " & """" &
userName2 & """"

        MsgBox(message,
MsgBoxStyle.Information,
saysText)

    Else

```

```

MsgBox("Data Has Been
Sucessfully Updated.",
MsgBoxStyle.Information,
saysText)

        Dim message
As String = "Your New UserName
Is " & "'''''' & userName2 & "''''''

MsgBox(message,
MsgBoxStyle.Information,
saysText)

        End If
        initData()
    End If
End Sub

Private Sub
cmdNew_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnNew.Click
    initData()
End Sub

Private Sub
cDocument_KeyUp(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles cDocument.KeyUp
    If e.KeyCode =
Keys.Enter Then
Search_EmployeeList()
    End If
End Sub

Private Sub
txtEmployeeName_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
txtEmployeeName.KeyUp
    If e.KeyCode =
Keys.Enter Then
        rbMale.Focus()
    End If
End Sub

Private Sub
txtEmployeeEmail_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr

```

```

gs) Handles
txtEmployeeEmail.KeyUp
    If e.KeyCode =
Keys.Enter Then
txtEmployeeAddress.Focus()
    End If
End Sub

Private Sub
txtEmployeePhone_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
txtEmployeePhone.KeyUp
    If e.KeyCode =
Keys.Enter Then
txtEmployeeCity.Focus()
    End If
End Sub

Private Sub
txtEmployeeCity_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
txtEmployeeCity.KeyUp
    If e.KeyCode =
Keys.Enter Then
txtEmployeeCountry.Focus()
    End If
End Sub

Private Sub
txtEmployeeName_TextChanged(ByV
al sender As System.Object,
ByVal e As System.EventArgs)
Handles
txtEmployeeName.TextChanged

    End Sub
End Class

```

Form Position

```

Imports System.Text

Public Class Position

```

```

        Private isEdited As Boolean
        = False
        Private ds2 As DataSet

        Private Sub
        frmType_Load(ByVal sender As
        System.Object, ByVal e As
        System.EventArgs) Handles
        MyBase.Load
            initData()
        End Sub

        Sub initData()
            Text = "Insert New
            Position"
            isEdited = False
            txtPositionID.Text =
            "[auto_generate]"
            txtPositionName.Text =
            ""
            txtNotes.Text = ""
            txtPositionName.Focus()
        End Sub
        WriteOnly Property
        ds2Call() As DataSet
            Set(ByVal value As
            DataSet)
                ds2 = value
            End Set
        End Property

        Private Sub
        Delete_Position(ByVal sender As
        System.Object, ByVal e As
        System.EventArgs) Handles
        btnDelete.Click
            If isEdited Then
                Dim positionID As
                String = txtPositionID.Text

                If MsgBox("Delete
                This Position?",
                MsgBoxStyle.YesNo + vbQuestion,
                saysText) = MsgBoxResult.Yes
                Then
                    If
                    db.connectDatabase Then
                        Dim result
                        As Integer =
                        db.executeNonQuery("DELETE FROM
                        m_position WHERE position_id =
                        '" & positionID & "'")
                        If result >
                        0 Then
                            MsgBox(positionID + " Has Been
                            Sucessfully Deleted.",
                            MsgBoxStyle.Critical, saysText)
                        End If
                    Else
                        MsgBox("Failed connecting to
                        database server.", 16,
                        saysText)
                    End If
                Else
                    MsgBox("Cannot
                    Delete Empty Customer!",
                    MsgBoxStyle.Critical, saysText)
                End If
            End Sub

        Private Sub
        Load_Positions(ByVal sender As
        System.Object, ByVal e As
        System.EventArgs) Handles
        cDocument.Click
            Search_PositionList()
        End Sub

        Sub Search_PositionList()
            Dim qs As New
            QuerySelection
            With qs
                qs.sqlMain = "
                mp.position_id AS ID, mp.name
                AS Name, mp.note AS Note FROM
                m_position mp "
                qs.sqlOrderBy = "
                mp.position_id DESC "
            End With
            Dim h As New
            Browser(qs, Me, "ds2Call",
            "Position List")
            Text = "Edit Existing
            Position"
            If h.ShowDialog() =
            Windows.Forms.DialogResult.Yes
            Then
                txtPositionID.Text
                = ds2.Tables(0).Rows(0).Item(0)
                txtPositionName.Text =
                ds2.Tables(0).Rows(0).Item(1)
            End If
        End Sub
    
```

```

        txtNotes.Text =
ds2.Tables(0).Rows(0).Item(2)
        isEdited = True
    End If
End Sub

Private Sub
btnNew_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnNew.Click
    initData()
End Sub

Private Sub
btnSave_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnSave.Click
    Dim positionName As
String =
txtPositionName.Text.Trim()
    Dim notes As String =
txtNotes.Text.Trim(), sqlString
As String = Nothing
    If positionName = ""
Then
        MsgBox(fillText +
"Name", MsgBoxStyle.Critical,
saysText)
    txtPositionName.Focus()
    Else
        Dim result As
Integer
        If Not isEdited
Then
            txtPositionID.Text =
db.getAutoNo("POS",
True).Trim()
            Dim strInsert
As StringBuilder = New
StringBuilder
            strInsert.Append(" INSERT INTO
m_position ")
            strInsert.AppendFormat(" VALUES
('{0}', '{1}', '{2}')"
, txtPositionID.Text,
positionName, notes)
            sqlString =
strInsert.ToString()
            result =
db.executeNonQuery(sqlString)
        Else
            Dim strUpdate
As StringBuilder = New
StringBuilder
            strUpdate.Append(" UPDATE
m_position ")
            strUpdate.AppendFormat(" SET
name = '{0}', note = '{1}'",
positionName, notes)
            strUpdate.AppendFormat(" WHERE
position_id = '{0}'",
txtPositionID.Text.Trim())
            sqlString =
strUpdate.ToString()
            result =
db.executeNonQuery(sqlString)
        End If
        If result > 0 Then
            If Not isEdited
Then
                MsgBox("Data Has Been
Sucessfully Inserted.",
MsgBoxStyle.Information,
saysText)
            Else
                MsgBox("Data Has Been
Sucessfully Updated.",
MsgBoxStyle.Information,
saysText)
            End If
            initData()
        End If
    End Sub

Private Sub
cDocument_KeyUp(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles cDocument.KeyUp
    If e.KeyCode =
Keys.Enter Then
        Search_PositionList()
    End If
End Sub

```



```

        Private Sub
txtPositionID_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles txtPositionID.KeyUp
        If e.KeyCode =
Keys.Enter Then

Search_PositionList()
        End If
    End Sub

    Private Sub
txtPositionName_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
txtPositionName.KeyUp
        If e.KeyCode =
Keys.Enter Then
            txtNotes.Focus()
        End If
    End Sub

    Private Sub
ToolStrip1_ItemClicked(ByVal
sender As System.Object, ByVal
e As
System.Windows.Forms.ToolStripI
temClickedEventArgs) Handles
ToolStrip1.ItemClicked

    End Sub
End Class

```

Form PlafondKredit

```

Imports System.Text
Imports System.Data.SqlClient

Public Class PlafondKredit
    Private isEdited As Boolean
= False
    Private companyID As String
= Nothing
    Private plafondID As String
= Nothing
    Private nominalPlan As
String = Nothing
    Private company_Name As
String = Nothing

```

```

    Private statusName As
String = Nothing
    Private notes As String =
Nothing
    Private ds2 As DataSet

    Sub initData()
        Text = "Insert New
Plafond"
        isEdited = False
        companyID = Nothing
        txtPlafondID.Text =
"[auto_generate]"
        txtNominalPlan.Text =
""
        txtNominalPlan.Focus()
        txtNominalActual.Text =
""
        txtCompanyID.Text = ""
        txtStatus.Text =
"UNAPPROVED"
        txtNotes.Text = ""
        btnLoadCompany.Enabled
= True
    End Sub

    WriteOnly Property
ds2Call() As DataSet
        Set(ByVal value As
DataSet)
            ds2 = value
        End Set
    End Property

    Private Sub
Load_Plafond(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnLoadPlafond.Click

```

```

Search_PlafondKreditList()
    End Sub

    Sub
Search_PlafondKreditList()
        plafondID = Nothing
        Dim qs As New
QuerySelection
        With qs
            qs.sqlMain = "
tp.Plafond_id AS ID,
tp.Plafond_nominal_plan AS
NominalPlan,
tp.Plafond_nominal_actual AS

```

```

NominalActual, tp.company_id as
CompanyID, " & _
"
mc.company_name AS CompanyName,
tp.[status] AS Status, tp.notes
AS Notes, tp.date AS
CreateDate, tp.input_by AS
CreatedBy " & _
" FROM
t_plafondkredit tp JOIN
m_company mc ON
mc.company_id = tp.company_id "
qs.sqlOrderBy = "
tp.Plafond_id DESC "
End With
Dim h As New
Browser(qs, Me, "ds2Call",
"Plafond List")
Text = "Edit Existing
Plafond"
If h.ShowDialog() =
Windows.Forms.DialogResult.Yes
Then
txtPlafondID.Text =
ds2.Tables(0).Rows(0).Item(0).T
oString().Trim()
plafondID =
txtPlafondID.Text
txtNominalPlan.Text
=
ds2.Tables(0).Rows(0).Item(1).T
oString().Trim()
txtNominalActual.Text =
ds2.Tables(0).Rows(0).Item(2).T
oString().Trim()
companyID =
ds2.Tables(0).Rows(0).Item(3).T
oString().Trim()
txtCompanyID.Text =
ds2.Tables(0).Rows(0).Item(4).T
oString().Trim()
txtStatus.Text =
ds2.Tables(0).Rows(0).Item(5).T
oString().Trim()
txtNotes.Text =
ds2.Tables(0).Rows(0).Item(6).T
oString().Trim()
btnLoadCompany.Enabled = False
isEdited = True
End If
End Sub

Private Sub
Delete_Data(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnDelete.Click
If isEdited Then
plafondID =
txtPlafondID.Text

If
MsgBox(deleteText,
MsgBoxStyle.YesNo + vbQuestion,
saysText) = MsgBoxResult.Yes
Then
If
db.connectDatabase Then
Dim result
As Integer =
db.executeNonQuery("DELETE FROM
t_plafondkredit WHERE
Plafond_id = '" & plafondID &
'")
If result >
0 Then
MsgBox(plafondID + " Has Been
Sucessfully Deleted.",
MsgBoxStyle.Information,
saysText)
initData()
End If
Else
MsgBox(failedConnectionText,
MsgBoxStyle.Critical, saysText)
End If
End If
Else
MsgBox(deleteEmptyText,
MsgBoxStyle.Critical, saysText)
End If
End Sub

Private Sub Save_Data(ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
btnSave.Click
nominalPlan =
txtNominalPlan.Text.Trim()
company_Name =
txtCompanyID.Text.Trim()

```

```

        statusName =
txtStatus.Text.Trim()
        notes =
txtNotes.Text.Trim()

        If
String.IsNullOrEmpty(nominalPla
n) Then
            MsgBox(fillText +
"Nominal Plan",
MsgBoxStyle.Critical, saysText)

txtNominalPlan.Focus()
            ElseIf Not
IsNumeric(nominalPlan) Then
                MsgBox("Data Type
Must Be In Numeric!",
MsgBoxStyle.Exclamation,
saysText)

txtNominalPlan.Focus()

txtNominalPlan.SelectAll()
            ElseIf
String.IsNullOrEmpty(company_Na
me) Then
                MsgBox(fillText +
"Company",
MsgBoxStyle.Critical, saysText)

txtCompanyID.Focus()
            Else
                Dim result As
Integer = 0
                Dim strBuilder As
StringBuilder
                Dim sqlString As
String = ""
                Dim planNominal As
Long = Long.Parse(nominalPlan)

                If Not isEdited
Then
                    txtPlafondID.Text =
db.getAutoNo("PLF", True)
                    plafondID =
txtPlafondID.Text.Trim()
                    strBuilder =
New StringBuilder

                    strBuilder.Append(" INSERT INTO
t_plafondkredit(Plafond_id,
Plafond_nominal_plan,
company_id, [status], notes,
input_by) ")

                    strBuilder.AppendFormat("
VALUES ('{0}', ", plafondID)

                    strBuilder.AppendFormat(" {0},
", nominalPlan)

                    strBuilder.AppendFormat("
'{0}', ", companyID)

                    strBuilder.AppendFormat("
'{0}', ", statusName)

                    strBuilder.AppendFormat("
'{0}', ", notes)

                    strBuilder.AppendFormat("
'{0}') ", user_name)

                    sqlString =
strBuilder.ToString()
                    result =
db.executeNonQuery(sqlString)
                Else
                    strBuilder =
New StringBuilder

                    strBuilder.Append(" UPDATE
t_plafondkredit SET ")

                    strBuilder.AppendFormat("
Plafond_nominal_plan = {0}, ",
nominalPlan)

                    strBuilder.AppendFormat("
company_id = '{0}', ",
companyID)

                    strBuilder.AppendFormat(" notes
= '{0}', ", notes)

                    strBuilder.AppendFormat("
input_by = '{0}' ", user_name)

                    strBuilder.AppendFormat(" WHERE
Plafond_id = '{0}'", plafondID)

                    sqlString =
strBuilder.ToString()
                    result =
db.executeNonQuery(sqlString)
                End If

                If result > 0 Then

```

```

                                If Not isEdited
Then
MsgBox("Data Has Been
Sucessfully Inserted.",
MsgBoxStyle.Information,
saysText)
                                Else
MsgBox("Data Has Been
Sucessfully Updated.",
MsgBoxStyle.Information,
saysText)
                                End If
                                initData()
                                End If
                                End If
                                End Sub

                                Private Sub
Insert_Data(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnNew.Click
                                initData()
                                End Sub

                                Private Sub
PlafondKredit_Load(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles
MyBase.Shown
                                initData()
                                End Sub

                                Private Sub
btnLoadCompany_Click(ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
btnLoadCompany.Click
                                Search_CompanyList()
                                End Sub

                                Sub Search_CompanyList()
                                Dim qs As New
QuerySelection
                                With qs
                                qs.sqlMain = "
mc.company_id, mc.company_name,
mc.[address], mc.phone, " & _
"
mc.fax, mc.email, mc.website,
mc.city, mc.country " & _
" FROM
m_company mc " & _
" LEFT
JOIN t_plafondkredit tp ON
tp.company_id = mc.company_id "
& _
"
WHERE tp.Plafond_nominal_actual
= 0 " & _
" OR
tp.Plafond_id IS NULL "
qs.sqlOrderBy = "
mc.company_id DESC "
                                End With
                                Dim h As New
Browser(qs, Me, "ds2Call",
"Company List")
                                Text = "Edit Existing
Company"
                                If h.ShowDialog() =
Windows.Forms.DialogResult.Yes
Then
                                companyID =
ds2.Tables(0).Rows(0).Item(0).T
oString().Trim()
                                txtCompanyID.Text =
ds2.Tables(0).Rows(0).Item(1).T
oString().Trim()
                                End If
                                End Sub

                                Private Sub
btnLoadPlafond_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
btnLoadPlafond.KeyUp
                                If e.KeyCode =
Keys.Enter Then
                                Search_PlafondKreditList()
                                End If
                                End Sub

                                Private Sub
txtNominalPlan_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
txtNominalPlan.KeyUp
                                If e.KeyCode =
Keys.Enter Then
                                txtNominalActual.Focus()

```

```

        End If
    End Sub

    Private Sub
txtNominalActual_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
txtNominalActual.KeyUp
    If e.KeyCode =
Keys.Enter Then

txtCompanyID.Focus()
    End If
End Sub

    Private Sub
btnLoadCompany_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
btnLoadCompany.KeyUp
    If e.KeyCode =
Keys.Enter Then

Search_CompanyList()
    End If
End Sub

    Private Sub
txtCompanyID_KeyUp(ByVal sender
As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles txtCompanyID.KeyUp
    If e.KeyCode =
Keys.Enter Then

Search_CompanyList()
    End If
End Sub

    Private Sub
txtStatus_KeyUp(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles txtStatus.KeyUp
    If e.KeyCode =
Keys.Enter Then
        txtNotes.Focus()
    End If
End Sub

    Private Sub
txtNominalPlan_TextChanged(ByVa

```

```

l sender As System.Object,
ByVal e As System.EventArgs)
Handles
txtNominalPlan.TextChanged
    If Not
String.IsNullOrEmpty(txtNominal
Plan.Text.Trim) Then
        If
IsNumeric(txtNominalPlan.Text.T
rim) Then

txtNominalActual.Text =
txtNominalPlan.Text
        Else
            MsgBox("Nominal
Plan Must Be In Numeric!",
MsgBoxStyle.Exclamation,
saysText)

txtNominalPlan.SelectAll()

txtNominalPlan.Focus()
        End If
    End If
End Sub
End Class

```

Form ApprovedPlafond

```

Imports System.Text
Imports System.Data.SqlClient

Public Class ApprovedPlafond
    Private approvedID As
String = Nothing
    Private plafondID As String
= Nothing
    Private notes As String =
Nothing
    Private isEdited As Boolean
= False
    Private ds2 As DataSet

    Sub initData()
        txtApprovedID.Text =
"[auto generate]"
        approvedID = Nothing
        txtPlafondID.Text = ""
        plafondID = Nothing
        txtNotes.Text = ""
        notes = Nothing
        isEdited = False
    End Sub
End Class

```

```

        btnLoadPlafond.Enabled
= True
    End Sub

    WriteOnly Property
ds2Call() As DataSet
    Set(ByVal value As
DataSet)
        ds2 = value
    End Set
    End Property

    Private Sub
Load_Plafond(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnLoadPlafond.Click

Search_PlafondKreditList()
    End Sub

    Sub
Search_PlafondKreditList()
        plafondID = Nothing
        Dim qs As New
QuerySelection
        With qs
            qs.sqlMain = "
tp.Plafond_id AS ID,
tp.Plafond_nominal_plan AS
NominalPlan,
tp.Plafond_nominal_actual AS
NominalActual, tp.company_id as
CompanyID, " & _
mc.company_name AS CompanyName,
tp.[status] AS Status, tp.notes
AS Notes, tp.date AS
CreateDate, tp.input_by AS
CreatedBy " & _
            " FROM
            t_plafondkredit tp " & _
            " JOIN
            m_company mc ON
mc.company_id = tp.company_id "
            & _
            " LEFT
JOIN t_approvedplafond ta ON
ta.plafond_id = tp.Plafond_id "
            & _
            " LEFT
JOIN t_plafondbreak_header tph
ON tph.Plafond_id =
tp.Plafond_id " & _
            WHERE ta.approved_id IS NULL
            AND tph.t_plafondbreak_id IS
            NULL " & _
            " AND
            tp.[status] = 'UNAPPROVED' "
            qs.sqlOrderBy = "
            tp.Plafond_id DESC "
        End With
        Dim h As New
Browser(qs, Me, "ds2Call",
"Plafond List")
        If h.ShowDialog() =
Windows.Forms.DialogResult.Yes
Then
            txtPlafondID.Text =
ds2.Tables(0).Rows(0).Item(0).T
oString().Trim()
            plafondID =
txtPlafondID.Text
        End If
    End Sub

    Private Sub
Delete_Data(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnDelete.Click
        approvedID =
txtApprovedID.Text
        If IsEdited Then
            If
MsgBox(deleteText,
MsgBoxStyle.YesNo + vbQuestion,
saysText) = MsgBoxResult.Yes
Then
                If
db.connectDatabase Then
                    Dim
sqlString As String = " DELETE
FROM t_approvedplafond WHERE
approved_id = '" & approvedID &
"'"
                    sqlString
+= " UPDATE t_plafondkredit SET
[status] = 'UNAPPROVED' WHERE
Plafond_id = '" & plafondID &
"'"
                    Dim result
As Integer =
db.executeNonQueryTransaction(s
qlString)
                    If result >
0 Then

```

```

MsgBox(approvedID + " Has Been
Sucessfully Deleted.",
MsgBoxStyle.Information,
saysText)

initData()
End If
Else
MsgBox(failedConnectionText,
MsgBoxStyle.Critical, saysText)
End If
initData()
End If
Else
MsgBox(deleteEmptyText,
MsgBoxStyle.Critical, saysText)
End If
End Sub

Private Sub Save_Data(ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
btnSave.Click
If Not isEdited Then
notes =
txtNotes.Text.Trim()
If
(String.IsNullOrEmpty(txtPlafon
dID.Text) Or
String.IsNullOrEmpty(plafondID)
) Then
MsgBox(chooseText + "Plafond",
MsgBoxStyle.Critical, saysText)

txtPlafondID.Focus()
Else
Dim result As
Integer = 0
Dim strBuilder
As StringBuilder
Dim sqlString
As String = ""
Dim statusName
As String = "APPROVED"

txtApprovedID.Text =
db.getAutoNo("APR", True)
approvedID =
txtApprovedID.Text.Trim()

strBuilder =
New StringBuilder

strBuilder.Append(" INSERT INTO
t_approvedplafond (approved_id,
plafond_id, [status], notes,
employee_id) ")

strBuilder.AppendFormat("
VALUES('{0}', ", approvedID)

strBuilder.AppendFormat("
'{0}', ", plafondID)

strBuilder.AppendFormat("
'{0}', ", statusName)

strBuilder.AppendFormat("
'{0}', ", notes)

strBuilder.AppendFormat("
'{0}') ", employee_id)
sqlString =
strBuilder.ToString()

strBuilder =
New StringBuilder

strBuilder.AppendFormat("
UPDATE t_plafondkredit SET
[status] = '{0}',
Plafond_nominal_actual =
Plafond_nominal_plan WHERE
Plafond_id= '{1}' ",
statusName, plafondID)
sqlString +=
strBuilder.ToString()
result =
db.executeNonQueryTransaction(s
qlString)

If result > 0
Then
MsgBox("Data Has Been
Sucessfully Inserted.",
MsgBoxStyle.Information,
saysText)

btnLoadPlafond.Enabled = False
initData()
End If

End If

```



```

Else
    MsgBox("Cannot Save
Approved Plafond!",
MsgBoxStyle.Exclamation,
saysText)
End If
End Sub

Private Sub
Insert_Data(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnNew.Click
    initData()
End Sub

Private Sub
PlafondKredit_Load(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles
MyBase.Shown
    initData()
End Sub

Private Sub
Load_Approved(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnLoadApproved.Click

Search_ApprovedPlafondList()
End Sub

Sub
Search_ApprovedPlafondList()
    approvedID = Nothing
    Dim qs As New
QuerySelection
    With qs
        qs.sqlMain = "
ta.approved_id AS ApprovedID,
ta.plafond_id AS PlafondID,
ta.[status] AS StatusName, " &
_
        "
ta.notes AS Notes, ta.date AS
CreateDate, me.name AS
CreatedBy " & _
        " FROM
t_approvedplafond ta " &
_
        " JOIN
m_employee me ON me.employee_id
= ta.employee_id " & _
        " LEFT
JOIN t_plafondbreak_header tph
ON tph.plafond_id =
ta.plafond_id " & _
        "
WHERE tph.t_plafondbreak_id IS
NULL "
        qs.sqlOrderBy = "
ta.approved_id DESC "
    End With
    Dim h As New
Browser(qs, Me, "ds2Call",
"Approved List")
    If h.ShowDialog() =
Windows.Forms.DialogResult.Yes
Then
        btnLoadPlafond.Enabled = False
        txtApprovedID.Text
=
ds2.Tables(0).Rows(0).Item(0).T
oString().Trim()
        approvedID =
txtPlafondID.Text
        txtPlafondID.Text =
ds2.Tables(0).Rows(0).Item(1).T
oString().Trim()
        plafondID =
txtPlafondID.Text
        txtNotes.Text =
ds2.Tables(0).Rows(0).Item(3).T
oString().Trim()

        btnLoadPlafond.Enabled = False
        isEdited = True
    End If
End Sub

Private Sub
btnLoadApproved_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
btnLoadApproved.KeyUp
    If e.KeyCode =
Keys.Enter Then

Search_ApprovedPlafondList()
End If
End Sub

Private Sub
btnLoadPlafond_KeyUp(ByVal
sender As Object, ByVal e As

```

```

System.Windows.Forms.KeyEventArgs) Handles
btnLoadPlafond.KeyUp

Search_PlafondKreditList()
End Sub
End Class

```

Form PlafondBreak

```

Imports System
Imports System.Text
Imports System.String
Imports System.Globalization
Imports System.Data.SqlClient

Public Class PlafondBreak
    Private ds1 As DataSet
    Private ds2 As DataSet
    Private ds3 As DataSet
    Private ds4 As DataSet
    Private isEdited As Boolean
    = False
    Private plafondID As String
    = Nothing
    Private breakPlafondID As
String = Nothing

    Sub initData()
        Text = "Insert New
Break Plafond"
        isEdited = False
        txtBreakPlafondID.Text
= "[auto generate]"
        txtPlafondID.Text = ""
        txtNominalActual.Text =
""
        plafondID = Nothing

        DataSet1.Tables(0).Rows.Clear()
        txtNotes.Text = ""
        txtNotes.Focus()
        DataGridView1.Visible =
False
    End Sub

    Function isExist(ByVal
customerID As String) As
Boolean
        Dim row As DataRow
        Dim exist As Boolean =
False

```

```

        For Each row In
DataSet1.Tables(0).Rows
            If
row("CustomerID").ToString().To
Upper().Equals(customerID) Then
                exist = True
            Exit For
        End If
    Next
    Return exist
End Function

```

```

WriteOnly Property
ds4Call() As DataSet
    Set(ByVal value As
DataSet)
        ds4 = value
    End Set
End Property

```

```

WriteOnly Property
ds1Call() As DataSet
    Set(ByVal value As
DataSet)
        ds1 = value
    End Set
End Property

```

```

WriteOnly Property
ds2Call() As DataSet
    Set(ByVal value As
DataSet)
        ds2 = value
    End Set
End Property

```

```

WriteOnly Property
ds3Call() As DataSet
    Set(ByVal value As
DataSet)
        ds3 = value
    End Set
End Property

```

```

Private Sub
Load_Plafonds(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnLoadPlafond.Click

    Search_PlafondKreditList()
End Sub

```

```

Sub
Search_PlafondKreditList()
    plafondID = Nothing
    Dim qs As New
    QuerySelection
        With qs
            qs.sqlMain = "
tp.Plafond_id AS ID,
tp.Plafond_nominal_plan AS
NominalPlan,
tp.Plafond_nominal_actual AS
NominalActual, " & _
"
tp.company_id as CompanyID,
mc.company_name AS CompanyName,
tp.[status] AS Status, tp.notes
AS Notes, " & _
"
tp.date AS CreatedDate,
tp.input_by AS CreatedBy " & _
" FROM
t_plafondkredit tp " & _
" JOIN
m_company mc ON
mc.company_id = tp.company_id "
& _
" JOIN
t_approvedplafond ta ON
ta.plafond_id = tp.Plafond_id "
& _
"
WHERE
tp.Plafond_nominal_actual > 0 "
& _
" AND
tp.[status] = 'APPROVED'
AND ta.[status] = 'APPROVED' "
qs.sqlOrderBy = "
tp.Plafond_id DESC "
End With
Dim h As New
Browser(qs, Me, "ds2Call",
"Plafond List")
If h.ShowDialog() =
Windows.Forms.DialogResult.Yes
Then
    txtPlafondID.Text =
ds2.Tables(0).Rows(0).Item(0).T
oString().Trim()
    plafondID =
txtPlafondID.Text

txtNominalActual.Text =
ds2.Tables(0).Rows(0).Item(2)

DataGridView1.Visible = True
txtNotes.Focus()
End If
End Sub

Private Sub
cNew_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnClear.Click
    initData()
End Sub

Private Sub
cClose_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs)
    Close()
End Sub

Private Sub
DataGridView1_CellBeginEdit(ByV
al sender As Object, ByVal e As
System.Windows.Forms.DataGridVi
ewCellCancelEventArgs) Handles
DataGridView1.CellBeginEdit
    If e.ColumnIndex < 2
Then
        e.Cancel = True
    Else
        Dim strCustomer As
String = DataGridView1.Item(0,
e.RowIndex).Value.ToString()
        If
(String.IsNullOrEmpty(strCustom
er)) Then
            MsgBox("Please
Add Customer, First!",
MsgBoxStyle.Exclamation,
saysText)
            e.Cancel = True
        End If
    End If
End Sub

Private Sub
DataGridView1_CellClick(ByVal
sender As Object, ByVal e As
System.Windows.Forms.DataGridVi

```

```

ewCellEventArgs) Handles
DataGridView1.CellClick
    If (e.RowIndex) >= 0
Then
    If e.ColumnIndex =
1 Or e.ColumnIndex = 0 Then
Search_CustomerList(e.RowIndex)
    Else
DataGridView1.Item(2,
e.RowIndex).Selected = True
    End If
    End If
End Sub

Sub
Search_CustomerList(ByVal
rowIndex As Integer)
    Dim qs As New
QuerySelection
    With qs
        qs.sqlMain = "
mc2.customer_id, mc2.name,
mc2.[address], mc2.[type],
mc2.phone, mc2.email, " & _
mc2.city, mc2.country,
tp.company_id, mc.company_name
" & _
        " FROM
t_plafondkredit tp " & _
        " JOIN
m_company mc ON mc.company_id =
tp.company_id " & _
        " JOIN
m_customer mc2 ON
mc2.company_id = mc.company_id
" & _
WHERE tp.Plafond_id = ' " &
plafondID & "' "
        qs.sqlOrderBy = "
mc2.customer_id DESC "
    End With
    Dim h As New
Browser(qs, Me, "ds1Call",
"Customer List")

    If h.ShowDialog() =
Windows.Forms.DialogResult.Yes
Then
        Dim customerID As
String =
ds1.Tables(0).Rows(0).Item(0).T
oString().Trim().ToUpper()
        If
isExist(customerID) Then
            MsgBox("Data
Already Exist!", 48, saysText)
        Else
DataGridView1.Rows(rowIndex).Ce
lls(0).Value =
ds1.Tables(0).Rows(0).Item(0)

DataGridView1.Rows(rowIndex).Ce
lls(1).Value =
ds1.Tables(0).Rows(0).Item(1)

DataGridView1.Item(2,
rowIndex).Selected = True

DataGridView1.BeginEdit(True)
        End If
    End If
End Sub

Private Sub
DataGridView1_CellDoubleClick(B
yVal sender As Object, ByVal e
As
System.Windows.Forms.DataGridVi
ewCellEventArgs) Handles
DataGridView1.CellDoubleClick
    If (e.RowIndex) >= 0
Then
        If e.ColumnIndex =
1 Or e.ColumnIndex = 0 Then
Search_CustomerList(e.RowIndex)
        Else
DataGridView1.Item(2,
e.RowIndex).Selected = True
        End If
    End If
End Sub

Private Sub
Load_BreakPlafond(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles
btnLoadBreakPlafond.Click
Search_BreakPlafondList()
End Sub

```

```

Sub
Search_BreakPlafondList()
    Dim rc As
    SqlClient.SqlDataReader
    Dim qs As New
    QuerySelection
    Dim dt As DataRow
    With qs
        .sqlMain = "
tph.t_plafondbreak_id AS
PlafondBreakID, tph.date AS
CreateDate, tph.Plafond_id AS
PlafondID, " & _
"
tp.Plafond_nominal_actual AS
NominalActual,
tp.Plafond_nominal_plan AS
NominalPlan, ta.approved_id AS
ApprovedID, tp.company_id AS
CompanyID, " & _
"
mc.company_name AS CompanyName,
mc.[address] AS CompanyAddress,
tph.note AS Notes " & _
" FROM
t_plafondbreak_header tph " & _
" JOIN
t_plafondkredit tp ON
tp.Plafond_id = tph.Plafond_id
" & _
" JOIN
t_approvedplafond ta ON
ta.plafond_id = tp.Plafond_id "
& _
" JOIN
m_company mc ON mc.company_id =
tp.company_id " & _
" WHERE
tp.[status] = 'APPROVED' AND
ta.[status] = 'APPROVED' "
.sqlOrderBy = "
tph.t_plafondbreak_id DESC "
    End With
    Dim h As New
    Browser(qs, Me, "ds3Call",
    "Break Plafond List")

    If h.ShowDialog() =
    Windows.Forms.DialogResult.Yes
    Then

        txtBreakPlafondID.Text =
        ds3.Tables(0).Rows(0).Item(0).T
        oString().Trim().ToUpper()

        breakPlafondID =
        txtBreakPlafondID.Text
        txtPlafondID.Text =
        ds3.Tables(0).Rows(0).Item(2).T
        oString().Trim().ToUpper()
        plafondID =
        txtPlafondID.Text

        txtNominalActual.Text =
        ds3.Tables(0).Rows(0).Item(3).T
        oString()

        txtNotes.Text =
        ds3.Tables(0).Rows(0).Item(9).T
        oString().Trim()

        DataSet1.Tables(0).Rows.Clear()
        Dim sqlBuilder As
        StringBuilder = New
        StringBuilder
        sqlBuilder.Append("
SELECT tpd.t_plafondbreak_id,
tpd.customer_id, mc.name,
tpd.notes, tpd.nominal AS
Nominal ")
        sqlBuilder.Append("
FROM t_plafondbreak_detail tpd
")
        sqlBuilder.Append("
JOIN m_customer mc ON
mc.customer_id =
tpd.customer_id ")
        sqlBuilder.AppendFormat(" WHERE
tpd.t_plafondbreak_id = '{0}'
", breakPlafondID)
        sqlBuilder.Append("
ORDER BY tpd.t_plafondbreak_id
DESC ")
        Dim sqlString As
        String = sqlBuilder.ToString()
        rc =
        db.dataReader(sqlString)

        If rc.HasRows()
        Then
            isEdited = True
            Text = "Edit
Break Plafond"

            DataGridView1.Visible = True
            While rc.Read()
                dt =
                DataSet1.Tables(0).NewRow()

```

```

dt("CustomerID") =
rc("customer_id").ToString()

dt("CustomerName") =
rc("name").ToString()

dt("Nominal") =
CType(rc("nominal"), Double)
dt("Notes")
= rc("notes")

DataSet1.Tables(0).Rows.Add(dt)
End While
End If
rc.Close()
End If
End Sub

Private Sub
cSave_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnSave.Click

DataGridView1.EndEdit(True)
Dim detailCount As
Integer =
Me.DataGridView1.Rows.Count
Dim counter As Integer
= detailCount - 2
Dim notes As String =
txtNotes.Text.Trim()
Dim result As Integer =
0
Dim nominalActual As
Double =
Double.Parse(txtNominalActual.T
ext)
Dim sqlString As String
= ""

Try
If
(String.IsNullOrEmpty(plafondID
)) Then

MsgBox(chooseText & "Plafond
ID", MsgBoxStyle.Critical)

txtPlafondID.Focus()
ElseIf detailCount
= 0 Then

MsgBox(fillText
& "Detail",
MsgBoxStyle.Critical)
Else
If counter >= 0
Then
Dim
sqlStringDetail As String = ""
Dim
totalNominalDetail As Integer =
0

If Not
isEdited Then breakPlafondID =
db.getAutoNo("BPL", True)

For i As
Integer = 0 To counter
Dim
customerIDDetail As String =
DataGridView1.Rows(i).Cells(0).
Value.ToString().ToUpper()
If Not
String.IsNullOrEmpty(customerID
Detail) Then
Dim
nominalDetail As Double =
Convert.ToDouble(DataGridView1.
Rows(i).Cells(2).Value.ToString
())
Dim
notesDetail As String =
DataGridView1.Rows(i).Cells(3).
Value.ToString().Trim()
If
Not IsNumeric(nominalDetail)
Then
Throw New Exception("Nominal
Must Be In Numeric! Please
Check On Customer '" &
customerIDDetail & "'")

ElseIf Not nominalDetail > 0
Then
Throw New Exception("Nominal
Must Be Greater Than (0) Zero!
Please Check On Customer '" &
customerIDDetail & "'")

Else

```

```

Dim sqlBuilder As StringBuilder
= New StringBuilder

sqlBuilder.Append(" INSERT INTO
t_plafondbreak_detail VALUES("

sqlBuilder.AppendFormat("
'{0}', ", breakPlafondID)

sqlBuilder.AppendFormat("
'{0}', ", customerIDDetail)

sqlBuilder.AppendFormat(" {0},
", nominalDetail)

sqlBuilder.AppendFormat("
'{0}') ", notesDetail)

sqlStringDetail +=
sqlBuilder.ToString()

totalNominalDetail +=
nominalDetail

If
End
End If
Next
If isEdited
Then
Dim
strSql As String = "SELECT
tpd.nominal FROM
t_plafondbreak_detail tpd WHERE
tpd.t_plafondbreak_id = '" &
breakPlafondID & "'"

Dim
dataReader As
SqlClient.SqlDataReader =
db.dataReader(strSql)
If
dataReader.HasRows() Then
While dataReader.Read()

nominalActual +=
dataReader("nominal")

End
While
End If
dataReader.Close()
End If

If
totalNominalDetail > 0 Then
If Not
totalNominalDetail <=
nominalActual Then
Throw New Exception("Nominal
Detail Must Be Less Than Or
Equal To Nominal Actual!")
Else
If
Not isEdited Then
Dim sqlBuilder As StringBuilder
= New StringBuilder

sqlBuilder.Append(" INSERT INTO
t_plafondbreak_header
(t_plafondbreak_id, note,
Plafond_id) ")

sqlBuilder.Append(" VALUES("

sqlBuilder.AppendFormat("
'{0}', ", breakPlafondID)

sqlBuilder.AppendFormat("
'{0}', ", notes)

sqlBuilder.AppendFormat("
'{0}') ", plafondID)

sqlString +=
sqlBuilder.ToString()

Else
If
(String.IsNullOrEmpty(breakPlafondID)) Then
MsgBox(fillText & "Break
Plafond ID",
MsgBoxStyle.Critical, saysText)

txtBreakPlafondID.Focus()

Else
sqlString += " UPDATE
t_plafondbreak_header SET note
= '" & notes & "' WHERE

```



```

t_plafondbreak_id = '' &
breakPlafondID & '' ''

sqlString += " DELETE FROM
t_plafondbreak_detail WHERE
t_plafondbreak_id = '' &
breakPlafondID & '' ''

End If

If

percentage As Double =
Utility.GetFormatDouble((totalN
ominalDetail / nominalActual) *
100)

question As String = "Amount
Has Reached " &
percentage.ToString() & "%. Are
You Sure Want To Save?"

sqlString += sqlStringDetail

nominalActual -=
totalNominalDetail

sqlString += " UPDATE
t_plafondkredit SET
Plafond_nominal_actual = " &
nominalActual & " WHERE
Plafond_id = '' & plafondID &
'' ''

MsgBox(question,
MsgBoxStyle.YesNo + vbQuestion,
saysText) = MsgBoxResult.Yes
Then

result =
db.executeNonQueryTransaction(s
qlString)

If result > 0 Then

If isEdited Then

MsgBox("Data Has Been
Successfully Updated.",
MsgBoxStyle.Information,
saysText)

Else

MsgBox("Data Has Been
Successfully Inserted.",
MsgBoxStyle.Information,
saysText)

End If

initData()

End If

If

End If

Else

Throw New
Exception("Please Insert At
Least One Detail With Correctly
Nominal!")

End If

End If

Catch ex As Exception
MsgBox(ex.Message,
MsgBoxStyle.Critical, saysText)
End Try

End Sub

Private Sub
frmBrngMsk_Load(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
 MyBase.Load
initData()

DataGridView1.Columns(0).Defaul
tCellStyle.BackColor =
Color.AliceBlue

DataGridView1.Columns(0).DataGr
idView.AlternatingRowsDefaultCe
llStyle.BackColor =
Color.AliceBlue

End Sub

Private Sub
cDelete_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
 btnDelete.Click

```

```

        If isEdited Then
            Dim detailCount As Integer =
                Me.DataGridView1.Rows.Count - 2
            Dim result As Integer = 0
            Dim nominalActual As Double =
                Double.Parse(txtNominalActual.Text)
            breakPlafondID =
                txtBreakPlafondID.Text
            plafondID =
                txtPlafondID.Text
            Try
                For i As Integer = 0 To detailCount
                    nominalActual +=
                        Convert.ToDouble(DataGridView1.
                            Rows(i).Cells(2).Value)
                Next
                Dim sqlString As String = " DELETE FROM
                t_plafondbreak_detail WHERE
                t_plafondbreak_id = '" &
                breakPlafondID & "' "
                sqlString += "
                DELETE FROM
                t_plafondbreak_header WHERE
                t_plafondbreak_id = '" &
                breakPlafondID & "' "
                sqlString += "
                UPDATE t_plafondkredit SET
                Plafond_nominal_actual =
                Plafond_nominal_plan WHERE
                Plafond_id = '" & plafondID &
                "' "
                result =
                db.executeNonQueryTransaction(s
                qlString)
                If result > 0
                Then
                    MsgBox(breakPlafondID & " Has
                    Been Successfully Deleted.",
                    MsgBoxStyle.Information,
                    saysText)
                    initData()
                End If
            Catch ex As Exception
                MsgBox(ex.Message,
                MsgBoxStyle.Critical, saysText)
            End Try
        Else
            MsgBox("Cannot
            Delete Empty Data!",
            MsgBoxStyle.Critical, saysText)
        End If
    End Sub

    Private Sub
        btnLoadBreakPlafond_KeyUp(ByVal
        sender As Object, ByVal e As
        System.Windows.Forms.KeyEventAr
        gs) Handles
        btnLoadBreakPlafond.KeyUp
            If e.KeyCode =
                Keys.Enter Then
                Search_BreakPlafondList()
            End If
        End Sub

    Private Sub
        txtBreakPlafondID_KeyUp(ByVal
        sender As Object, ByVal e As
        System.Windows.Forms.KeyEventAr
        gs) Handles
        txtBreakPlafondID.KeyUp
            If e.KeyCode =
                Keys.Enter Then
                Search_BreakPlafondList()
            End If
        End Sub

    Private Sub
        txtPlafondID_KeyUp(ByVal sender
        As Object, ByVal e As
        System.Windows.Forms.KeyEventAr
        gs) Handles txtPlafondID.KeyUp
            If e.KeyCode =
                Keys.Enter Then
                Search_PlafondKreditList()
            End If
        End Sub

    Private Sub
        txtNominalActual_KeyUp(ByVal
        sender As Object, ByVal e As
        System.Windows.Forms.KeyEventAr

```

```

gs) Handles
txtNominalActual.KeyUp
    If e.KeyCode =
Keys.Enter Then
        txtNotes.Focus()
    End If
End Sub
End Class

```

Form PaymentPlafond

```

Imports System.Text
Imports System.Data
Imports System.Data.OleDb

Public Class PaymentPlafond
    Private ds2 As DataSet
    Private isEdited As Boolean
    Private paymentID As String
    Private Sub
frmType_Load(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
MyBase.Load
        initData()
    End Sub
    Sub initData()
        txtPaymentID.Text =
"[auto_generate]"
        paymentID = Nothing
        txtFilePath.Text = ""
        txtNotes.Text = ""
        DataSet1.Tables(0).Rows.Clear()
        btnOpenFile.Enabled =
True
        isEdited = False
        HideMenu()
        txtFilePath.Focus()
    End Sub
    Sub HideMenu()
        DataGridView1.Visible =
False
        lblCompanyID.Visible =
False

```

```

txtCompanyID.Visible =
False
        lblCompanyName.Visible
= False
        txtCompanyName.Visible
= False
        lblAmount.Visible =
False
        txtAmount.Visible =
False
        lblStatus.Visible =
False
        txtStatus.Visible =
False
        lblPaymentDate.Visible
= False
        txtPaymentDate.Visible
= False
        lblNotes.Visible =
False
        txtNotes.Visible =
False
    End Sub
    Sub ShowMenu()
        DataGridView1.Visible =
True
        lblCompanyID.Visible =
True
        txtCompanyID.Visible =
True
        lblCompanyName.Visible
= True
        txtCompanyName.Visible
= True
        lblAmount.Visible =
True
        txtAmount.Visible =
True
        lblStatus.Visible =
True
        txtStatus.Visible =
True
        lblPaymentDate.Visible
= True
        txtPaymentDate.Visible
= True
        lblNotes.Visible = True
        txtNotes.Visible = True
    End Sub
    WriteOnly Property
ds2Call() As DataSet

```

```

        Set(ByVal value As
DataSet)
            ds2 = value
        End Set
    End Property

    Public Function
LoadExcel(ByVal fileName As
String) As DataTable
        Dim sExcelFile As
String = Nothing
        Dim sheetName As String
= "Sheet1"
        sExcelFile = fileName
        Dim strConn As String =
"Provider=Microsoft.ACE.OLEDB.1
2.0;" & "Data Source=" &
fileName & ";" & "Extended
Properties="""Excel
12.0;IMEX=1""""

        Dim myData As
OleDbDataAdapter = New
OleDbDataAdapter
        Dim myDataset As
DataSet = New DataSet
        myData = New
OleDbDataAdapter("SELECT * FROM
[" & sheetName & "$]", strConn)

        myData.TableMappings.Add("Table
", "Book1")
        myData.Fill(myDataset)
        Return
myDataset.Tables(0)
    End Function

    Private Sub
Load_Payment(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnLoadPaymentID.Click

        Search_PaymentPlafondList()
    End Sub

    Sub
Search_PaymentPlafondList()
        Dim dataReader As
SqlClient.SqlDataReader
        Dim dRow As DataRow
        Dim qs As New
QuerySelection
        With qs
            qs.sqlMain = "
            tpph.payment_id AS PaymentID,
            tpph.company_id AS CompanyID,
            mc.company_name AS CompanyName,
            " & _
            "
            tpph.amount AS Amount,
            tpph.paymentDate AS
PaymentDate, tpph.kredit_status
AS StatusKredit, tpph.notes AS
Notes " & _
            " FROM
            t_payment_plafond_header
            tpph " & _
            " JOIN
            m_company mc ON
            mc.company_id = tpph.company_id
            "
            qs.sqlOrderBy = "
            tpph.payment_id DESC "
        End With
        Dim h As New
Browser(qs, Me, "ds2Call",
"Payment List")
        If h.ShowDialog() =
Windows.Forms.DialogResult.Yes
Then
            ShowMenu()
            btnOpenFile.Enabled
= False
            paymentID =
ds2.Tables(0).Rows(0).Item(0).T
oString().Trim()
            txtPaymentID.Text =
paymentID
            txtCompanyID.Text =
ds2.Tables(0).Rows(0).Item(1).T
oString().Trim()
            txtCompanyName.Text
=
ds2.Tables(0).Rows(0).Item(2).T
oString().Trim()
            txtAmount.Text =
ds2.Tables(0).Rows(0).Item(3).T
oString().Trim()
            txtPaymentDate.Text
= ds2.Tables(0).Rows(0).Item(4)
            txtStatus.Text =
ds2.Tables(0).Rows(0).Item(5).T
oString().Trim()
            txtNotes.Text =
ds2.Tables(0).Rows(0).Item(6).T
oString().Trim()
            isEdited = True
        End If
    End Sub

```

```

DataSet1.Tables(0).Rows.Clear()
    Dim sqlBuilder As
StringBuilder = New
StringBuilder
    sqlBuilder.Append("
SELECT tppd.customer_id,
mc.name, tppd.nominal,
tppd.notes ")
    sqlBuilder.Append("
FROM t_payment_plafond_detail
tppd ")
    sqlBuilder.Append("
JOIN m_customer mc ON
mc.customer_id =
tppd.customer_id ")

sqlBuilder.AppendFormat(" WHERE
tppd.payment_id = '{0}' ",
paymentID)
    sqlBuilder.Append("
ORDER BY tppd.customer_id ASC
")
    Dim sqlString As
String = sqlBuilder.ToString()
    dataReader =
db.dataReader(sqlString)

    If
dataReader.HasRows() Then
        isEdited = True

DataGridView1.Visible = True
        While
dataReader.Read()
            dRow =
DataSet1.Tables(0).NewRow()

dRow("CustomerID") =
dataReader("customer_id").ToString()

dRow("CustomerName") =
dataReader("name").ToString()

dRow("Nominal") =
CType(dataReader("nominal"),
Double)

dRow("Notes") =
dataReader("notes")

DataSet1.Tables(0).Rows.Add(dRow)
        End While
    End If
    dataReader.Close()
End If
End Sub

Private Sub
btnNew_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnClear.Click
    initData()
End Sub

Private Sub
btnOpenFile_Click(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles
btnOpenFile.Click
    Select_File()
End Sub

Sub Select_File()
    With FileBrowser
        .Filter = "Excel
Workbook|*.xls;*.xlsx"
        .FilterIndex = 1
        .Title = "Select
File To Import"
        If .ShowDialog =
Windows.Forms.DialogResult.OK
Then
            Try
                txtFilePath.Text = .FileName
            Catch ex As
Exception
                Throw ex
            End Try
        End If
    End With
End Sub

Private Sub
btnSave_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnSave.Click
    Dim path As String =
txtFilePath.Text

```

```

        Dim dt As DataTable =
New DataTable
        If Not isEdited Then
            Try
                dt =
LoadExcel(path)
                paymentID =
db.getAutoNo("PPL", True)

txtPaymentID.Text = paymentID
                Dim index As
Integer = 0, totalAmount As
Double = 0
                Dim rowCount As
Integer = dt.Rows.Count
                Dim dateNow As
DateTime = DateTime.Now,
dateCreated As DateTime
                Dim dateDiff As
System.TimeSpan, indexDetail As
Integer = 0
                Dim statusName
As String = "OK"
                Dim strBuilder
As StringBuilder
                Dim companyID
As String =
dt.Rows(index)(0).ToString().Trim()
                Dim companyName
As String =
dt.Rows(index)(1).ToString().Trim()
                Dim amount As
Double =
Double.Parse(dt.Rows(index)(2).ToString().Trim())
                Dim notesHeader
As String =
dt.Rows(index)(3).ToString().Trim()
                Dim sqlString
As String = " SELECT * FROM
t_plafondkredit tp WHERE
tp.company_id = '" & companyID
& "' "
                Dim dataReader
As SqlClient.SqlDataReader =
db.dataReader(sqlString)
                If
dataReader.HasRows() Then
                    While
dataReader.Read()

                        dateCreated =
dataReader("date")
                        Exit
                    While
                        End While
                    End If

dataReader.Close()
                        dateDiff =
dateNow.Subtract(dateCreated)
                        If
dateDiff.Days > 30 Then
                            statusName = "SLOW"

                                strBuilder =
New StringBuilder

                                strBuilder.Append(" INSERT INTO
t_payment_plafond_header(paymen
t_id, company_id, amount,
notes, kredit_status) ")
                                strBuilder.AppendFormat("
VALUES('{0}', ", paymentID)
                                strBuilder.AppendFormat("
'{0}', ", companyID)
                                strBuilder.AppendFormat(" {0},
", amount)
                                strBuilder.AppendFormat("
'{0}', ", notesHeader)
                                strBuilder.AppendFormat("
'{0}') ", statusName)
                                sqlString =
strBuilder.ToString()
                                index += 3
                                indexDetail =
index
                                While
                                    (String.IsNullOrEmpty(dt.Rows(i
ndex)(0).ToString().Trim()) =
False)
                                        Dim
customerID As String =
dt.Rows(index)(0).ToString().Trim().ToUpper()
                                        Dim nominal
As Double =
Double.Parse(dt.Rows(index)(2).ToString().Trim())

```

```

Dim
notesDetail As String =
dt.Rows(index)(3).ToString().Trim()

strBuilder
= New StringBuilder

strBuilder.Append(" INSERT INTO
t_payment_plafond_detail
VALUES(")

strBuilder.AppendFormat("
'{0}', ", paymentID)

strBuilder.AppendFormat("
'{0}', ", customerID)

strBuilder.AppendFormat(" {0},
", nominal)

strBuilder.AppendFormat("
'{0}')" ", notesDetail)

sqlString
+= strBuilder.ToString()
totalAmount
+= nominal

index += 1
If index >=
rowCount Then Exit While
End While

Dim result As
Integer = 0
If amount =
totalAmount Then
result =
db.executeNonQueryTransaction(s
qlString)

If result >
0 Then

MsgBox("Data Has Been
Successfully Inserted.",
MsgBoxStyle.Information,
saysText)

initData()

btnOpenFile.Enabled = False

ShowMenu()

DataSet1.Tables(0).Rows.Clear()

txtCompanyID.Text = companyID

txtCompanyName.Text =
companyName

txtAmount.Text = amount

txtStatus.Text = statusName

txtPaymentDate.Text = dateNow

txtNotes.Text = notesHeader

While
(String.IsNullOrEmpty(dt.Rows(i
ndexDetail)(0).ToString().Trim(
)) = False)

Dim
customerID As String =
dt.Rows(indexDetail)(0).ToStrin
g().Trim().ToUpper()

Dim
customerName As String =
dt.Rows(indexDetail)(1).ToStrin
g().Trim()

Dim
nominal As Double =
Double.Parse(dt.Rows(indexDetai
l)(2).ToString().Trim())

Dim
notesDetail As String =
dt.Rows(indexDetail)(3).ToStrin
g().Trim()

Dim
dRow As DataRow =
DataSet1.Tables(0).NewRow()

dRow("CustomerID") = customerID

dRow("CustomerName") =
customerName

dRow("Nominal") = nominal

dRow("Notes") = notesDetail

DataSet1.Tables(0).Rows.Add(dRo
w)

indexDetail += 1

If
indexDetail >= rowCount Then
Exit While

```



```

End
While
    isEdited = True
    End If
Else
    MsgBox("Total Nominal Detail Is Not Equal To Amount Header!",
    MsgBoxStyle.Critical, saysText)
    initData()
    End If
    Catch ex As
Exception
    Dim message As
String = ex.Message
    If
ex.Message.Contains("Sheet1$")
Then
        message =
        "Import Failed! Please Rename Your Sheet To ""Sheet1""."
        End If
        MsgBox(message,
        MsgBoxStyle.Critical, saysText)
        initData()
    End Try
    End If
End Sub

Private Sub
btnDelete_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnDelete.Click
    If isEdited Then
        Dim result As
Integer = 0
        paymentID =
txtPaymentID.Text
        Try
            Dim sqlString
As String = " DELETE FROM
t_payment_plafond_detail WHERE
payment_id = '" & paymentID &
"' "
            sqlString += "
DELETE FROM
t_payment_plafond_header WHERE
payment_id = '" & paymentID &
"' "
            result =
db.executeNonQueryTransaction(s
qlString)
        End Try
    End If
End Sub

Then
    MsgBox(paymentID & " Has Been
Successfully Deleted.",
    MsgBoxStyle.Information,
    saysText)
    initData()
    End If
    Catch ex As
Exception
    MsgBox(ex.Message,
    MsgBoxStyle.Critical, saysText)
    End Try
    Else
        MsgBox("Cannot
Delete Empty Data!",
        MsgBoxStyle.Critical, saysText)
        End If
    End Sub

Private Sub
btnLoadPaymentID_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
btnLoadPaymentID.KeyUp
    If e.KeyCode =
Keys.Enter Then
        Search_PaymentPlafondList()
    End If
End Sub

Private Sub
btnOpenFile_KeyUp(ByVal sender
As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles btnOpenFile.KeyUp
    Select_File()
End Sub

Private Sub
txtFilePath_KeyUp(ByVal sender
As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles txtFilePath.KeyUp
    Select_File()
End Sub

Private Sub
txtCompanyID_KeyUp(ByVal sender
As Object, ByVal e As

```

```

System.Windows.Forms.KeyEventArgs) Handles txtCompanyID.KeyUp
    If e.KeyCode =
Keys.Enter Then

txtCompanyName.Focus()
    End If
End Sub

Private Sub
txtCompanyName_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles
txtCompanyName.KeyUp
    If e.KeyCode =
Keys.Enter Then
        txtAmount.Focus()
    End If
End Sub

Private Sub
txtAmount_KeyUp(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles txtAmount.KeyUp
    If e.KeyCode =
Keys.Enter Then
        txtStatus.Focus()
    End If
End Sub

Private Sub
txtStatus_KeyUp(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles txtStatus.KeyUp
    If e.KeyCode =
Keys.Enter Then

txtPaymentDate.Focus()
    End If
End Sub
End Class

```

Form PlafondMonitoring

```

Imports System
Imports System.Text
Imports System.String

```

```

Imports System.Globalization
Imports System.Data.SqlClient
Imports
CrystalDecisions.CrystalReports
.Engine
Imports
CrystalDecisions.ReportSource
Imports CrystalDecisions.Shared
Imports
CrystalDecisions.Windows.Forms

Public Class PlafondMonitoring
    Private ds2 As DataSet
    Private dataSetReport As
DataSet

    'Notes Status
    'Jika Plafond Baru dibuat
    maka status adalah "New"/Belum
    ada nilai breakdown
    'Jika Sisa Plafond sudah
    mulai dibuat break tetapi nilai
    masih diatas 20% dari plafond
    status "FREE"
    'Jika sisa Plafond sudah
    >=20% maka status "WARNING"

    WriteOnly Property
ds2Call() As DataSet
        Set(ByVal value As
DataSet)
            ds2 = value
        End Set
    End Property

    Private Sub
PlafondMonitoring_Load(ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
MyBase.Load
        InitData()
    End Sub

    Private Sub
btnDelete_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs)
        Close()
    End Sub

    Private Sub InitData()
        txtPlafondIDFrom.Text =
""
    End Sub

```

```

        txtPlafondIDTo.Text =
"""
        dtFrom.Text = ""
        dtTo.Text = ""
        rbPlafondID.Checked =
True
        Radio_Change()

DataSet1.Tables(0).Rows.Clear()

txtPlafondIDFrom.Focus()
    End Sub

    Private Sub Radio_Change()
        If rbPlafondID.Checked
Then
txtPlafondIDFrom.Enabled = True

txtPlafondIDTo.Enabled = True
        btnPlafond.Enabled
= True

btnLoadPlafondFrom.Enabled =
True

btnLoadPlafondTo.Enabled = True
        dtFrom.Enabled =
False

False
        dtTo.Enabled =
False

        btnDate.Enabled =
False
        Else
txtPlafondIDFrom.Enabled =
False

txtPlafondIDTo.Enabled = False
        btnPlafond.Enabled
= False

btnLoadPlafondFrom.Enabled =
False

btnLoadPlafondTo.Enabled =
False

        dtFrom.Enabled =
True

        dtTo.Enabled = True
        btnDate.Enabled =
True
        End If
    End Sub

    Private Sub
rbDate_CheckedChanged(ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
rbDate.CheckedChanged
        Radio_Change()
    End Sub

    Private Sub
rbPlafondID_CheckedChanged(ByVal
l sender As System.Object,
ByVal e As System.EventArgs)
Handles
rbPlafondID.CheckedChanged
        Radio_Change()
    End Sub

    Private Sub
btnLoadPlafondFrom_Click(ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
btnLoadPlafondFrom.Click
        Dim plafond_from As
String = txtPlafondIDFrom.Text
        Dim plafond_to As
String = txtPlafondIDTo.Text
        Search_Plafond(True)
    End Sub

    Private Sub
btnLoadPlafondFrom_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
btnLoadPlafondFrom.KeyUp
        If e.KeyCode =
Keys.Enter Then
            Dim plafond_from As
String = txtPlafondIDFrom.Text
            Dim plafond_to As
String = txtPlafondIDTo.Text
            Search_Plafond(True)
        End If
    End Sub

    Private Sub
Search_Plafond(ByVal
isPlafondFrom As Boolean)
        Dim qs As New
QuerySelection
        Dim sqlString As String
= " tp.Plafond_id AS ID,

```

```

tp.Plafond_nominal_plan AS
NominalPlan, " & _

" tp.Plafond_nominal_actual AS
NominalActual, tp.company_id as
CompanyID, " & _

" mc.company_name AS
CompanyName, tp.[status] AS
Status, tp.notes AS Notes, " &
_

" tp.date AS CreatedDate,
tp.input_by AS CreatedBy " & _

" FROM      t_plafondkredit tp
" & _

" JOIN      m_company mc ON
mc.company_id = tp.company_id "
& _

" JOIN t_approvedplafond ta ON
ta.plafond_id = tp.Plafond_id "
    With qs
    .sqlMain =
sqlString
    .sqlOrderBy = "
tp.Plafond_id ASC "
    End With

    Dim h As New
Browser(qs, Me, "ds2Call",
"Plafond List")
    If h.ShowDialog() =
Windows.Forms.DialogResult.Yes
Then
        If isPlafondFrom
Then
            txtPlafondIDFrom.Text =
ds2.Tables(0).Rows(0).Item(0).T
oString().Trim().ToUpper()
            Else
                txtPlafondIDTo.Text =
ds2.Tables(0).Rows(0).Item(0).T
oString().Trim().ToUpper()
            End If
        End If
    End Sub

    Private Sub
btnLoadPlafondTo_Click(ByVal

```

```

sender As System.Object, ByVal
e As System.EventArgs) Handles
btnLoadPlafondTo.Click
    Dim plafond_from As
String = txtPlafondIDFrom.Text
    Dim plafond_to As
String = txtPlafondIDTo.Text
    Search_Plafond(False)
    End Sub

    Private Sub
btnLoadPlafondTo_KeyUp(ByVal
sender As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles
btnLoadPlafondTo.KeyUp
    If e.KeyCode =
Keys.Enter Then
        Dim plafond_from As
String = txtPlafondIDFrom.Text
        Dim plafond_to As
String = txtPlafondIDTo.Text
        Search_Plafond(False)
    End If
    End Sub

    Private Sub
btnPlafond_Click(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles
btnPlafond.Click
    If
String.IsNullOrEmpty(txtPlafond
IDFrom.Text) And Not
String.IsNullOrEmpty(txtPlafond
IDTo.Text) Then
        txtPlafondIDFrom.Text =
txtPlafondIDTo.Text
    ElseIf
String.IsNullOrEmpty(txtPlafond
IDTo.Text) And Not
String.IsNullOrEmpty(txtPlafond
IDFrom.Text) Then
        txtPlafondIDTo.Text
= txtPlafondIDFrom.Text
    End If
    End Sub

    Private Sub
btnPlafond_KeyUp(ByVal sender
As Object, ByVal e As

```

```

System.Windows.Forms.KeyEventArgs) Handles btnPlafond.KeyUp
    If e.KeyCode =
Keys.Enter Then
    If
String.IsNullOrEmpty(txtPlafond
IDFrom.Text) And Not
String.IsNullOrEmpty(txtPlafond
IDTo.Text) Then

txtPlafondIDFrom.Text =
txtPlafondIDTo.Text
    ElseIf
String.IsNullOrEmpty(txtPlafond
IDTo.Text) And Not
String.IsNullOrEmpty(txtPlafond
IDFrom.Text) Then

txtPlafondIDTo.Text =
txtPlafondIDFrom.Text
    End If
    End If
End Sub

Private Sub
btnDate_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnDate.Click
    If
String.IsNullOrEmpty(dtFrom.Text)
And Not
String.IsNullOrEmpty(dtTo.Text)
Then
    dtFrom.Text =
dtTo.Text
    ElseIf
String.IsNullOrEmpty(dtTo.Text)
And Not
String.IsNullOrEmpty(dtFrom.Text)
Then
    dtTo.Text =
dtFrom.Text
    End If
End Sub

Private Sub
btnDate_KeyUp(ByVal sender As
Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles btnDate.KeyUp
    If e.KeyCode =
Keys.Enter Then
        If
String.IsNullOrEmpty(dtFrom.Text)
And Not
String.IsNullOrEmpty(dtTo.Text)
Then
            dtFrom.Text =
dtTo.Text
        ElseIf
String.IsNullOrEmpty(dtTo.Text)
And Not
String.IsNullOrEmpty(dtFrom.Text)
Then
            dtTo.Text =
dtFrom.Text
        End If
    End If
End Sub

Private Sub
Button1_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
Button1.Click
    DataSet1.Tables(0).Rows.Clear()
    Dim plafond_from As
String = txtPlafondIDFrom.Text,
    plafond_to As String =
txtPlafondIDTo.Text
    Dim date_from As
DateTime = dtFrom.Text, date_to
As DateTime = dtTo.Text
    date_to =
date_to.AddDays(1)
    Dim datefrom As String
= date_from.ToString(), dateto
As String = date_to.ToString()
    Dim isAllowed As
Boolean = False

    If rbPlafondID.Checked
Then
        If
String.IsNullOrEmpty(plafond_fr
om) And
String.IsNullOrEmpty(plafond_to
) Then
            MsgBox("To
Begin Searching, Please Choose
Plafond ID First!",
MsgBoxStyle.Exclamation,
saysText)
        ElseIf Not
String.IsNullOrEmpty(plafond_fr
om) And
String.IsNullOrEmpty(plafond_to
) Then
            Dim datefrom As String
= date_from.ToString(), dateto
As String = date_to.ToString()
            Dim isAllowed As
Boolean = False

            If rbPlafondID.Checked
Then
                If
String.IsNullOrEmpty(plafond_fr
om) And
String.IsNullOrEmpty(plafond_to
) Then
                    MsgBox("To
Begin Searching, Please Choose
Plafond ID First!",
MsgBoxStyle.Exclamation,
saysText)
                ElseIf Not
String.IsNullOrEmpty(plafond_fr
om) And
String.IsNullOrEmpty(plafond_to
) Then
                    Dim datefrom As String
= date_from.ToString(), dateto
As String = date_to.ToString()
                    Dim isAllowed As
Boolean = False
                End If
            End If
        End If
    End If
End Sub

```

```

om) And Not
String.IsNullOrEmpty(plafond_to
) Then
    If plafond_from
> plafond_to Then

MsgBox("PlafondTo Must Be
Grater Than PlafondFrom!",
MsgBoxStyle.Exclamation,
saysText)
    Else
        isAllowed =
True
    End If
    Else
        isAllowed =
True
    End If
    ElseIf date_from >
date_to Then
        MsgBox("DateTo Must
Be Greater Than DateFrom!",
MsgBoxStyle.Exclamation,
saysText)
    Else
        isAllowed = True
    End If

    If isAllowed Then
        Dim sqlBuilder As
StringBuilder = New
StringBuilder()
sqlBuilder.Append("SELECT
    x.PlafondID,
    x.CreatedDate, x.CompanyID,
    x.CompanyName, ")
        sqlBuilder.Append("
    MAX(x.NominalPlan)
AS NominalPlan,
    MAX(x.NominalActual) AS
    NominalActual, ")
        sqlBuilder.Append("
    MAX(x.Percentage)
AS Percentage, x.StatusName ")
        sqlBuilder.Append("FROM ")
        sqlBuilder.Append("(" ")
        SELECT
            tp.Plafond_id AS
PlafondID, tp.date AS
CreatedDate, ")

sqlBuilder.Append("
        tp.company_id AS
CompanyID, mc.company_name AS
CompanyName, ")
        sqlBuilder.Append("
        tp.Plafond_nominal_plan
AS NominalPlan,
        tp.Plafond_nominal_actual AS
NominalActual, ")
        sqlBuilder.Append("
        CAST(((tp.Plafond_nominal
_actual /
        tp.Plafond_nominal_plan) * 100)
AS FLOAT) AS Percentage, ")
        sqlBuilder.Append("
        CASE
        WHEN tph.t_plafondbreak_id IS
        NULL THEN 'NEW' ELSE ")
        sqlBuilder.Append("
        CASE WHEN
        ((tp.Plafond_nominal_actual /
        tp.Plafond_nominal_plan) * 100)
        > 20 THEN 'FREE' ")
        sqlBuilder.Append("
        ELSE 'WARNING' END ")
        sqlBuilder.Append("
        StatusName ")
        sqlBuilder.Append("
        FROM
        t_plafondkredit tp ")
        sqlBuilder.Append("
        JOIN
        m_company mc
ON mc.company_id =
        tp.company_id ")
        sqlBuilder.Append("
        JOIN
        t_approvedplafond ta ON
        ta.plafond_id = tp.Plafond_id
        ")
        sqlBuilder.Append("
        LEFT JOIN
        t_plafondbreak_header tph
ON tph.Plafond_id =
        tp.Plafond_id ")
        If
        rbPlafondID.Checked Then
            If Not
String.IsNullOrEmpty(plafond_fr

```

```

om) And Not
String.IsNullOrEmpty(plafond_to
) Then

sqlBuilder.AppendFormat(" WHERE
tp.Plafond_id BETWEEN '{0}' AND
'{1}' ", plafond_from,
plafond_to)
        ElseIf Not
String.IsNullOrEmpty(plafond_fr
om) Then

sqlBuilder.AppendFormat(" WHERE
tp.Plafond_id >= '{0}' ",
plafond_from)
        Else

sqlBuilder.AppendFormat(" WHERE
tp.Plafond_id <= '{0}' ",
plafond_to)
        End If
    Else

sqlBuilder.AppendFormat(" WHERE
tp.date BETWEEN '{0}' AND '{1}'
", datefrom, dateto)
        End If

sqlBuilder.Append(") x ")

sqlBuilder.Append("GROUP BY
x.PlafondID, x.CreatedDate,
x.CompanyID, x.CompanyName,
x.StatusName ")
        Dim sqlString As
String = sqlBuilder.ToString()
        Dim dataReader As
SqlDataReader =
db.dataReader(sqlString)
        If
dataReader.HasRows() Then
            While
dataReader.Read()
                Dim dRow As
DataRow =
DataSet1.Tables(0).NewRow()

dRow("dsPlafondID") =
dataReader.Item("PlafondID")
                Dim
createdDate As DateTime =
dataReader.Item("CreatedDate")

dRow("dsCreatedDate") =
createdDate.ToString("dd-MM-
yyyy")

dRow("dsCompanyID") =
dataReader.Item("CompanyID")

dRow("dsCompanyName") =
dataReader.Item("CompanyName")

dRow("dsNominalPlan") =
Utility.getFormatDouble(dataRea
der.Item("NominalPlan").ToStrin
g())

dRow("dsNominalActual") =
Utility.getFormatDouble(dataRea
der.Item("NominalActual").ToStr
ing())

dRow("dsPercentage") =
dataReader.Item("Percentage")

dRow("dsStatus") =
dataReader.Item("StatusName")

DataSet1.Tables(0).Rows.Add(dRo
w)
            End While
            dataSetReport =
DataSet1
        Else
            MsgBox("There's
No Data!",
MsgBoxStyle.Exclamation,
saysText)
        End If
        dataReader.Close()
    End If
End Sub

Private Sub
btnExport_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnExport.Click
    Dim rowCount As Integer
= Me.DataGridView1.Rows.Count
    rowCount -= 2
    If rowCount >= 0 Then
        With SaveFileDialog
            .Reset()
            .Filter =
"Excel 97-2003

```



```

Workbook|*.xls|Excel
Workbook|*.xlsx"
        .FilterIndex =
1
        .Title = "Save
Excel As:"

        If .ShowDialog
= Windows.Forms.DialogResult.OK
Then
            Try
Save_File(.FileName)
            Catch ex As
Exception
MsgBox(ex.InnerException,
MsgBoxStyle.Critical, saysText)
            End Try
        End If
    End With
Else
    MsgBox("Cannot
Export Empty Data! Please
Search Data First To Export.",
MsgBoxStyle.Exclamation,
saysText)
    End If
End Sub

Private Sub Save_File(ByVal
fullPath As String)
    Dim xlApp As
Microsoft.Office.Interop.Excel.
Application
    Dim xlWorkBook As
Microsoft.Office.Interop.Excel.
Workbook
    Dim xlWorkSheet As
Microsoft.Office.Interop.Excel.
Worksheet
    Dim misValue As Object
=
System.Reflection.Missing.Value
    Dim i As Integer, j As
Integer

    xlApp = New
Microsoft.Office.Interop.Excel.
ApplicationClass
    xlWorkBook =
xlApp.Workbooks.Add(misValue)
    xlWorkSheet =
xlWorkBook.Sheets("sheet1")

        For i = 0 To
DataGridView1.RowCount - 2
            For j = 0 To
DataGridView1.ColumnCount - 1
                For k As
Integer = 1 To
DataGridView1.Columns.Count
xlWorkSheet.Cells(1, k) =
DataGridView1.Columns(k -
1).HeaderText
xlWorkSheet.Cells(i + 2, j + 1)
= DataGridView1(j,
i).Value.ToString()
                Next
            Next
        Next

xlApp.DisplayAlerts =
False
xlWorkSheet.SaveAs(fullPath)
xlWorkBook.Close()
xlApp.Quit()

releaseObject(xlApp)

releaseObject(xlWorkBook)

releaseObject(xlWorkSheet)
    End Sub

Private Sub
releaseObject(ByVal obj As
Object)
    Try
        System.Runtime.InteropServices.
Marshal.ReleaseComObject(obj)
        obj = Nothing
    Catch ex As Exception
        obj = Nothing
    Finally
        GC.Collect()
    End Try
End Sub

Private Sub
btnClear_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnClear.Click

```

```

        InitData()
    End Sub

    Private Sub
    btnPrint_Click(ByVal sender As
    System.Object, ByVal e As
    System.EventArgs) Handles
    btnPrint.Click
        Dim rowCount As Integer
    = Me.DataGridView1.Rows.Count
        rowCount -= 2
        If rowCount >= 0 Then
            Dim report_viewer
    As ReportViewer = New
    ReportViewer
            Dim report As New
    ReportDocument
            Dim pField As New
    ParameterField()
            Dim disVal As New
    ParameterDiscreteValue()
            Dim pFields As New
    ParameterFields()

            'pField.ParameterFieldName =
    "testing"
            'disVal.Value =
    txtPlafondIDFrom.Text

            'pField.CurrentValues.Add(disVal)

            'pFields.Add(pField)
            report = New
    crPlafond

            report.SetDataSource(DataSet1.Tables(0))
            report.Refresh()
            Me.Cursor =
    Cursors.WaitCursor

            report_viewer.CrystalReportViewer1.ParameterFieldInfo =
    pFields

            report_viewer.CrystalReportViewer1.ReportSource = report

            report_viewer.CrystalReportViewer1.Show()
            Me.Cursor =
    Cursors.Default

            report_viewer.ShowDialog()
        Else
            MsgBox("Cannot
    Print Empty Data! Please Search
    Data First To Export.",
    MsgBoxStyle.Exclamation,
    saysText)
        End If
    End Sub
End Class

Form ReportViewer

Public Class ReportViewer

    Private Sub
    ReportViewer_KeyDown(ByVal
    sender As Object, ByVal e As
    System.Windows.Forms.KeyEventArgs) Handles Me.KeyDown
        If e.KeyCode =
    Keys.Escape Then Close()
    End Sub
End Class

Form ReportBreakPlafond

Imports System
Imports System.Text
Imports System.String
Imports System.Globalization
Imports System.Data.SqlClient
Imports
    CrystalDecisions.CrystalReports
    .Engine
Imports
    CrystalDecisions.ReportSource
Imports CrystalDecisions.Shared
Imports
    CrystalDecisions.Windows.Forms

Public Class ReportBreakPlafond
    Private nominal_plan As
    Double
    Private created_date As
    DateTime

```

```

        Private company_id As
String = Nothing
        Private company_name As
String = Nothing
        Private notes As String =
Nothing
        Private ds2 As DataSet

        WriteOnly Property
ds2Call() As DataSet
        Set(ByVal value As
DataSet)
            ds2 = value
        End Set
    End Property

    Private Sub
ReportBreakPlafond_Load(ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
MyBase.Load
        txtPlafondID.Focus()
    End Sub

    Private Sub
btnSearch_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnSearch.Click
        Try
            Dim qs As New
QuerySelection
            With qs
                .sqlMain = "
tp.Plafond_id AS PlafondID,
MAX(tp.Plafond_nominal_plan) AS
NominalPlan, " & _
                "
MAX(tp.Plafond_nominal_actual)
AS NominalActual, tp.company_id
AS CompanyID, " & _
                "
mc.company_name AS CompanyName,
tp.[status] AS Status, " & _
                "
tp.notes AS Notes, tp.date AS
CreateDate, tp.input_by AS
CreatedBy " & _
                "
FROM t_plafondkredit tp " & _
                "
JOIN m_company mc ON
mc.company_id = tp.company_id "
                & _
                "
                JOIN t_plafondbreak_header tph
                ON tph.Plafond_id =
                tp.Plafond_id " & _
                "
                GROUP BY tp.Plafond_id,
                tp.company_id, mc.company_name,
                tp.[status], tp.notes, tp.date,
                tp.input_by "
                .sqlOrderBy = "
                tp.Plafond_id DESC "
            End With
            Dim h As New
Browser(qs, Me, "ds2Call",
"Plafond List")
            If h.ShowDialog() =
Windows.Forms.DialogResult.Yes
Then
                txtPlafondID.Text =
ds2.Tables(0).Rows(0).Item(0).T
oString().Trim()
                nominal_plan =
ds2.Tables(0).Rows(0).Item(1)
                company_id =
ds2.Tables(0).Rows(0).Item(3).T
oString().Trim().ToUpper()
                company_name =
ds2.Tables(0).Rows(0).Item(4).T
oString().Trim()
                notes =
ds2.Tables(0).Rows(0).Item(6).T
oString().Trim()
                created_date =
ds2.Tables(0).Rows(0).Item(7)

                Dim report As
New ReportDocument
                Dim pField As
New ParameterField()
                Dim disVal As
New ParameterDiscreteValue()
                Dim pFields As
New ParameterFields()
                Dim dsData As
DataSet = New DataSet

                pField.ParameterFieldName =
"PlafondID"
                disVal.Value =
txtPlafondID.Text
            End If
        Catch
        End Try
    End Sub

```

```

pField.CurrentValues.Add(disVal
)
pFields.Add(pField)

        pField = New
ParameterField()
        disVal = New
ParameterDiscreteValue()

pField.ParameterFieldName =
"NominalPlan"
        disVal.Value =
Utility.getFormatDouble(nominal
_plan)
pField.CurrentValues.Add(disVal
)
pFields.Add(pField)

        pField = New
ParameterField()
        disVal = New
ParameterDiscreteValue()

pField.ParameterFieldName =
"CreatedDate"
        disVal.Value =
created_date.ToString("dd-MM-
yyyy")
pField.CurrentValues.Add(disVal
)
pFields.Add(pField)

        pField = New
ParameterField()
        disVal = New
ParameterDiscreteValue()

pField.ParameterFieldName =
"CompanyID"
        disVal.Value =
company_id
pField.CurrentValues.Add(disVal
)
pFields.Add(pField)

        pField = New
ParameterField()
        disVal = New
ParameterDiscreteValue()

pField.ParameterFieldName =
"CompanyName"
        disVal.Value =
company_name
pField.CurrentValues.Add(disVal
)
pFields.Add(pField)

        pField = New
ParameterField()
        disVal = New
ParameterDiscreteValue()

pField.ParameterFieldName =
"Notes"
        disVal.Value =
notes
pField.CurrentValues.Add(disVal
)
pFields.Add(pField)

        report = New
crBreakPlafond
        Dim strBuilder
As New StringBuilder()
        strBuilder.Append("SELECT
        CAST(ROW_NUMBER()
OVER(ORDER BY
tph.t_plafondbreak_id ASC) AS
VARCHAR) AS Row, ")

        strBuilder.Append("
        tph.t_plafondbreak_id AS
PlafondBreakID,
CONVERT(VARCHAR, tph.date, 105)
AS CreatedDate, ")

        strBuilder.Append("
        tpd.customer_id AS
CustomerID, mc.name AS
CustomerName, ")

        strBuilder.Append("

```

```

        tpd.nominal AS Nominal,
        tph.note AS Notes ")

strBuilder.Append("FROM
        t_plafondbreak_header tph
")

strBuilder.Append("JOIN
        t_plafondbreak_detail tpd
ON tpd.t_plafondbreak_id =
tph.t_plafondbreak_id ")

strBuilder.Append("JOIN
        m_customer mc ON
mc.customer_id =
tpd.customer_id ")

strBuilder.AppendFormat("WHERE
        tph.Plafond_id = '{0}' ",
txtPlafondID.Text)
        Dim sqlString
As String =
strBuilder.ToString()
        Dim sqlAdapter
As SqlDataAdapter =
db.dataAdapter(sqlString)

sqlAdapter.Fill(dsData,
"PlafondList")

report.SetDataSource(dsData.Tab
les(0))

report.Refresh()
Me.Cursor =
Cursors.WaitCursor

ReportViewer.ParameterFieldInfo
= pFields

ReportViewer.ReportSource =
report
Me.Cursor =
Cursors.Default

ReportViewer.Show()
        End If
        Catch ex As Exception
            MsgBox(ex.Message,
MsgBoxStyle.Critical, saysText)
        End Try
    End Sub
End Class

```

Form ReportPaymentPlafond

```

Imports System
Imports System.Text
Imports System.String
Imports System.Globalization
Imports System.Data.SqlClient
Imports
CrystalDecisions.CrystalReports
.Engine
Imports
CrystalDecisions.ReportSource
Imports CrystalDecisions.Shared
Imports
CrystalDecisions.Windows.Forms

Public Class
ReportPaymentPlafond
    Private amount As Double
    Private rest As Double
    Private kredit_status As
String = Nothing
    Private created_date As
DateTime
    Private company_id As
String = Nothing
    Private company_name As
String = Nothing
    Private notes As String =
Nothing
    Private ds2 As DataSet
    WriteOnly Property
ds2Call() As DataSet
        Set(ByVal value As
DataSet)
            ds2 = value
        End Set
    End Property

    Private Sub
ReportBreakPlafond_Load(ByVal
sender As System.Object, ByVal
e As System.EventArgs) Handles
MyBase.Load
        txtPlafondID.Focus()
    End Sub

    Private Sub
btnSearch_Click(ByVal sender As
System.Object, ByVal e As

```

```
System.EventArgs) Handles
btnSearch.Click
```

```
    Try
        Dim qs As New
        QuerySelection
        With qs
            qs.sqlMain = "
            tp.Plafond_id AS PlafondID,
            MAX(tp.Plafond_nominal_plan) AS
            NominalPlan, " & _
            "
            MAX(tp.Plafond_nominal_actual)
            AS NominalActual, tp.company_id
            AS CompanyID, " & _
            "
            mc.company_name AS CompanyName,
            tp.[status] AS Status, " & _
            "
            tp.notes AS Notes, tp.date AS
            CreatedDate, " & _
            "
            tp.input_by AS CreatedBy,
            MAX(tpph.amount) AS Amount, " &
            _
            "
            CAST((SUM(tpd.nominal) -
            MAX(tpph.amount)) AS
            NUMERIC(18,2)) AS Rest, " & _
            "
            tpph.kredit_status AS
            KreditStatus " & _
            "
            FROM t_plafondkredit tp " & _
            "
            JOIN m_company mc ON
            mc.company_id = tp.company_id "
            & _
            "
            JOIN t_payment_plafond_header
            tpph ON tpph.company_id =
            mc.company_id " & _
            "
            JOIN t_plafondbreak_header tph
            ON tph.Plafond_id =
            tp.Plafond_id " & _
            "
            JOIN t_plafondbreak_detail tpd
            ON tpd.t_plafondbreak_id =
            tph.t_plafondbreak_id "
            qs.sqlGroupBy =
            " tp.Plafond_id, tp.company_id,
            mc.company_name, tp.[status], "
            & _
```

```
        " tp.notes, tp.date,
        tp.input_by, tpph.kredit_status
        "
```

```
            qs.sqlOrderBy =
            " tp.Plafond_id DESC "
            End With
            Dim h As New
            Browser(qs, Me, "ds2Call",
            "Plafond List")
            If h.ShowDialog() =
            Windows.Forms.DialogResult.Yes
            Then
```

```
                txtPlafondID.Text =
                ds2.Tables(0).Rows(0).Item(0).T
                oString().Trim()
                company_id =
                ds2.Tables(0).Rows(0).Item(3).T
                oString().Trim().ToUpper()
                company_name =
                ds2.Tables(0).Rows(0).Item(4).T
                oString().Trim()
                notes =
                ds2.Tables(0).Rows(0).Item(6).T
                oString().Trim()
                created_date =
                ds2.Tables(0).Rows(0).Item(7)
                amount =
                ds2.Tables(0).Rows(0).Item(9)
                rest =
                ds2.Tables(0).Rows(0).Item(10)
                kredit_status =
                ds2.Tables(0).Rows(0).Item(11)
```

```
                Dim report As
                New ReportDocument
                Dim pField As
                New ParameterField()
                Dim disVal As
                New ParameterDiscreteValue()
                Dim pFields As
                New ParameterFields()
                Dim dsData As
                DataSet = New DataSet
```

```
                pField.ParameterFieldName =
                "CompanyID"
                disVal.Value =
                company_id
                pField.CurrentValues.Add(disVal
                )
```

```

pFields.Add(pField)

        pField = New
ParameterField()
        disVal = New
ParameterDiscreteValue()

pField.ParameterFieldName =
"CompanyName"
        disVal.Value =
company_name

pField.CurrentValues.Add(disVal
)

pFields.Add(pField)

        pField = New
ParameterField()
        disVal = New
ParameterDiscreteValue()

pField.ParameterFieldName =
"CreatedDate"
        disVal.Value =
created_date.ToString("dd-MM-
yyyy")

pField.CurrentValues.Add(disVal
)

pFields.Add(pField)

        pField = New
ParameterField()
        disVal = New
ParameterDiscreteValue()

pField.ParameterFieldName =
"Amount"
        disVal.Value =
Utility.GetFormatDouble(amount)

pField.CurrentValues.Add(disVal
)

pFields.Add(pField)

        pField = New
ParameterField()
        disVal = New
ParameterDiscreteValue()

pField.ParameterFieldName =
"Notes"
        disVal.Value =
notes

pField.CurrentValues.Add(disVal
)

pFields.Add(pField)

        pField = New
ParameterField()
        disVal = New
ParameterDiscreteValue()

pField.ParameterFieldName =
"Rest"
        disVal.Value =
rest

pField.CurrentValues.Add(disVal
)

pFields.Add(pField)

        pField = New
ParameterField()
        disVal = New
ParameterDiscreteValue()

pField.ParameterFieldName =
"KreditStatus"
        disVal.Value =
kredit_status

pField.CurrentValues.Add(disVal
)

pFields.Add(pField)

        report = New
crPayment

        Dim strBuilder
As New StringBuilder()

strBuilder.Append("SELECT
        CAST(ROW_NUMBER()
OVER(ORDER BY tppd.customer_id
ASC) AS VARCHAR) AS Row, ")

strBuilder.Append("
        tppd.customer_id AS

```



```

CustomerID, mc2.name AS
CustomerName, ")

strBuilder.Append("
    CONVERT(VARCHAR,
tpph.paymentDate, 105) AS
CreatedDate, ")

strBuilder.Append("
    tppd.nominal AS Nominal,
tppd.notes AS Notes, ")

strBuilder.Append("
    tpph.kredit_status AS
Status ")

strBuilder.Append("FROM
    t_payment_plafond_header
tpph ")

strBuilder.Append("JOIN
    t_payment_plafond_detail
tppd ON tppd.payment_id =
tpph.payment_id ")

strBuilder.Append("JOIN
    m_company mc ON
mc.company_id = tpph.company_id
")

strBuilder.Append("JOIN
    m_customer mc2 ON
mc2.customer_id =
tppd.customer_id ")

strBuilder.Append("JOIN
    t_plafondkredit tp ON
tp.company_id = mc.company_id
")

strBuilder.AppendFormat("WHERE
tp.Plafond_id = '{0}' ",
txtPlafondID.Text)
Dim sqlString
As String =
strBuilder.ToString()
Dim sqlAdapter
As SqlDataAdapter =
db.dataAdapter(sqlString)

sqlAdapter.Fill(dsData,
"PlafondList")

```

```

report.SetDataSource(dsData.Tab
les(0))

report.Refresh()
Me.Cursor =
Cursors.WaitCursor

ReportViewer.ParameterFieldInfo
= pFields

ReportViewer.ReportSource =
report
Me.Cursor =
Cursors.Default

ReportViewer.Show()
End If
Catch ex As Exception
    MsgBox(ex.Message,
MsgBoxStyle.Critical, saysText)
End Try
End Sub
End Class

```

Form Admin

```

Imports System.Text

Public Class Admin
    Private employeeID As
String = Nothing
    Private ds2 As DataSet

    Sub initData()
        Text = "Menu Admin"
        txtUserName.Text = ""
        txtUserName.Focus()
        txtPassword.Text = ""
        txtConfirm.Text = ""
        cbCustomer.Checked =
False
        cbCompany.Checked =
False
        cbEmployee.Checked =
False
        cbPosition.Checked =
False
        cbPlafondKredit.Checked
= False
    End Sub

```

```

cbApprovedPlafond.Checked =
False
    cbBreakPlafond.Checked
= False

cbPaymentPlafond.Checked =
False

cbReportSisaPlafond.Checked =
False

cbReportTransaksiKredit.Checked
= False

cbReportBreakPlafond.Checked =
False
    cbAdminMenu.Checked =
False
    cbImportData.Checked =
False
End Sub

Sub Search_UserList()
    Dim qs As New
QuerySelection
    With qs
        Dim sqlString As
StringBuilder = New
StringBuilder
        sqlString.Append("
mu.[user_id] AS ID, mu.username
AS UserName, mu.[password] AS
Password, mu.customer AS
CustomerRights, ")
        sqlString.Append("
mu.company AS CompanyRights,
mu.employee AS EmployeeRights,
mu.position AS PositionRights,
")
        sqlString.Append("
mu.plafond_kredit AS
PlafondKreditRights,
mu.approved_plafond AS
ApprovedPlafondRights, ")
        sqlString.Append("
mu.break_plafond AS
BreakPlafondRights,
mu.payment_plafond AS
PaymentPlafondRights, ")
        sqlString.Append("
mu.report_sisaplafond AS
ReportSisaPlafondRights,
mu.report_transaksikredit AS
ReportTransaksiKreditRights, ")
        sqlString.Append("
mu.report_breakplafond AS
ReportBreakPlafondRights,
mu.menu_admin AS
AdminMenuRights, ")
        sqlString.Append("
mu.importdata AS
ImportDataRights,
me.employee_id AS EmployeeID ")
        sqlString.Append("
FROM m_user mu ")
        sqlString.Append("
JOIN m_employee me ON
me.employee_id = mu.employee_id
")
        qs.sqlMain =
sqlString.ToString()
        qs.sqlOrderBy = "
mu.[user_id] DESC "
        End With
        Dim h As New
Browser(qs, Me, "ds2Call",
"User List")
        Text = "Edit Existing
User"
        If h.ShowDialog() =
Windows.Forms.DialogResult.Yes
Then
            employeeID =
Nothing
            txtUserName.Text =
ds2.Tables(0).Rows(0).Item(1).T
oString().Trim()
            txtPassword.Text =
ds2.Tables(0).Rows(0).Item(2).T
oString().Trim()
            txtConfirm.Text =
ds2.Tables(0).Rows(0).Item(2).T
oString().Trim()
            cbCustomer.Checked
= ds2.Tables(0).Rows(0).Item(3)
            cbCompany.Checked =
ds2.Tables(0).Rows(0).Item(4)
            cbEmployee.Checked
= ds2.Tables(0).Rows(0).Item(5)
            cbPosition.Checked
= ds2.Tables(0).Rows(0).Item(6)

            cbPlafondKredit.Checked =
ds2.Tables(0).Rows(0).Item(7)

```

```

cbApprovedPlafond.Checked =
ds2.Tables(0).Rows(0).Item(8)

cbBreakPlafond.Checked =
ds2.Tables(0).Rows(0).Item(9)

cbPaymentPlafond.Checked =
ds2.Tables(0).Rows(0).Item(10)

cbReportSisaPlafond.Checked =
ds2.Tables(0).Rows(0).Item(11)

cbReportTransaksiKredit.Checked =
=
ds2.Tables(0).Rows(0).Item(12)

cbReportBreakPlafond.Checked =
ds2.Tables(0).Rows(0).Item(13)
    cbAdminMenu.Checked =
ds2.Tables(0).Rows(0).Item(14)

cbImportData.Checked =
ds2.Tables(0).Rows(0).Item(15)
    employeeID =
ds2.Tables(0).Rows(0).Item(16).
ToString().Trim().ToUpper()
    End If
End Sub

WriteOnly Property
ds2Call() As DataSet
    Set(ByVal value As
DataSet)
        ds2 = value
    End Set
End Property

Private Sub
cNew_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnClear.Click
    initData()
End Sub

Private Sub
Admin_Load(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
MyBase.Shown
    initData()
End Sub

Private Sub
Update_User(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnSave.Click

    Dim password As String
    = txtPassword.Text.Trim()
    Dim confirmPwd As
String = txtConfirm.Text.Trim()

    If txtUserName.Text =
"" Then
        MsgBox(fillText &
"Name", MsgBoxStyle.Critical,
saysText)
        txtUserName.Focus()
    ElseIf txtPassword.Text
= "" Then
        MsgBox(chooseText &
"Password",
MsgBoxStyle.Critical, saysText)
        txtPassword.Focus()
    ElseIf txtConfirm.Text
= "" Then
        MsgBox(fillText &
"Confirm Password",
MsgBoxStyle.Critical, saysText)
        txtConfirm.Focus()
    ElseIf password =
confirmPwd Then
        Dim result As
Integer
        Dim strUpdate As
StringBuilder = New
StringBuilder
        strUpdate.Append("
UPDATE m_user SET ")

        strUpdate.AppendFormat("
[password] = '{0}', ",
txtPassword.Text)

        strUpdate.AppendFormat("
customer = '{0}', ",
cbCustomer.Checked)

        strUpdate.AppendFormat("
company = '{0}', ",
cbCompany.Checked)

        strUpdate.AppendFormat("

```

```

employee = '{0}', ",
cbEmployee.Checked)

strUpdate.AppendFormat("
position = '{0}', ",
cbPosition.Checked)

strUpdate.AppendFormat("
plafond_kredit = '{0}', ",
cbPlafondKredit.Checked)

strUpdate.AppendFormat("
approved_plafond = '{0}', ",
cbApprovedPlafond.Checked)

strUpdate.AppendFormat("
break_plafond = '{0}', ",
cbBreakPlafond.Checked)

strUpdate.AppendFormat("
payment_plafond = '{0}', ",
cbPaymentPlafond.Checked)

strUpdate.AppendFormat("
report_sisaplafond = '{0}', ",
cbReportSisaPlafond.Checked)

strUpdate.AppendFormat("
report_transaksikredit = '{0}',
",
cbReportTransaksiKredit.Checked
)

strUpdate.AppendFormat("
report_breakplafond = '{0}', ",
cbReportBreakPlafond.Checked)

strUpdate.AppendFormat("
menu_admin = '{0}', ",
cbAdminMenu.Checked)

strUpdate.AppendFormat("
importdata = '{0}', ",
cbImportData.Checked)

strUpdate.AppendFormat(" WHERE
username = '{0}'",
txtUserName.Text)
result =
db.executeNonQuery(strUpdate.ToString())

If result > 0 Then

MsgBox("Data
Has Been Successfully Updated.",
MsgBoxStyle.Information,
saysText)

initData()
End If
Else
MsgBox("Password Do
Not Match!",
MsgBoxStyle.Exclamation,
saysText)

txtPassword.Text =
""
txtConfirm.Text =
""
txtPassword.Focus()
End If
End Sub

Private Sub
Load_Users(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnLoadUser.Click
Search_UserList()
End Sub

Private Sub
Delete_User(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
btnDelete.Click
Dim userName As String
= txtUserName.Text
If
(String.IsNullOrEmpty(userName)
= False) Then
If Not
userName.Trim().ToUpper().Equal
s(user_name.Trim().ToUpper())
Then
If
MsgBox("Delete This User Will
Also Delete Its Employee
Reference. Are You Sure?",
MsgBoxStyle.YesNo + vbQuestion,
saysText) = MsgBoxResult.Yes
Then
If
db.connectDatabase Then
Dim
sqlString As String = " DELETE

```

```

FROM m_user WHERE username = ' "
& userName & "'

sqlString += " DELETE FROM
m_employee WHERE employee_id =
'" & employeeID & "'"

Dim
result As Integer =
db.executeNonQueryTransaction(s
qlString)

If
result > 0 Then

MsgBox(userName + " Has Been
Sucessfully Deleted.",
MsgBoxStyle.Information,
saysText)

initData()

End If
Else

MsgBox("Failed connecting to
database server.",
MsgBoxStyle.Critical, saysText)

End If
End If
Else
MsgBox("Cannot
Delete Your Own Data!",
MsgBoxStyle.Critical, saysText)

End If
Else
MsgBox("Cannot
Delete Empty User!",
MsgBoxStyle.Exclamation,
saysText)

End If
End Sub

Private Sub
btnLoadUser_KeyUp(ByVal sender
As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles btnLoadUser.KeyUp
If e.KeyCode =
Keys.Enter Then
Search_UserList()

End If
End Sub

Private Sub
txtUserName_KeyUp(ByVal sender
As Object, ByVal e As
System.Windows.Forms.KeyEventAr
gs) Handles txtUserName.KeyUp
If e.KeyCode =
Keys.Enter Then
txtPassword.Focus()

End If
End Sub

Private Sub
Admin_Load_1(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
MyBase.Load

End Sub
End Class

```