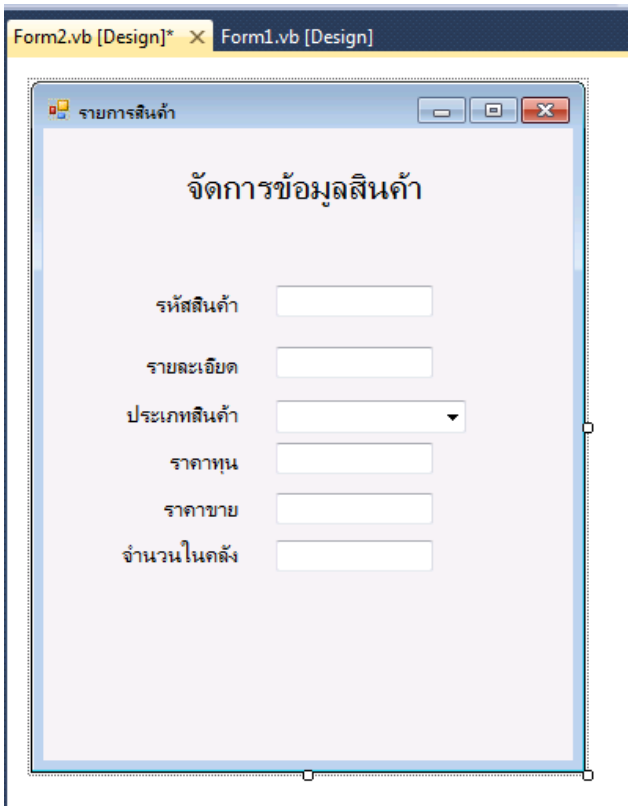
การสร้างฟอร์มข้อมูลสินค้า

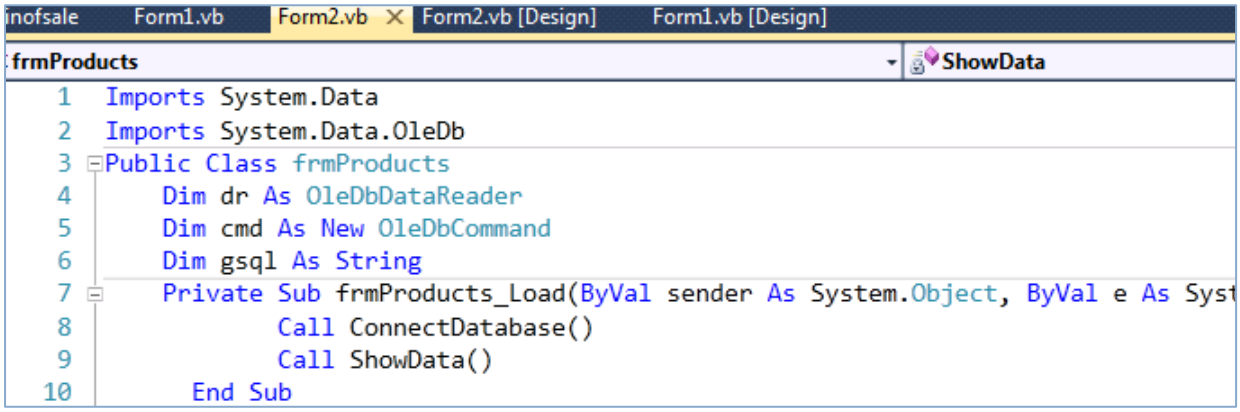
ในขั้นตอนนี้จะสร้างฟอร์มข้อมูลสินค้า เพื่อใช้ในการจัดการข้อมูลสินค้า โดยในการทำงานของฟอร์มข้อมูลสินค้านี้จะมีการทำงานของการผูกติดข้อมูลแบบ Complex Binding Data คือจะมีการทำงานโดยใช้ตารางข้อมูล 2 ตาราง คือ ตาราง Product และตาราง ProductType โดยจะดึงข้อมูลจากตาราง ProductType มาแสดงในส่วนประเภทสินค้า เนื่องจากก่อนหน้านี้เราได้ทำการจัดเก็บข้อมูลประเภทสินค้าไปแล้ว จึงสามารถนำข้อมูลส่วนนี้มาใช้เพื่อกำหนดเป็นรายละเอียดของข้อมูลสินค้าได้ ดังนั้น ให้นักศึกษาสร้างฟอร์มข้อมูลสินค้าดังภาพ



โดยกำหนดคุณสมบัติของคอนโทรล ดังนี้

Control	Properties - Name	Properties – Text
Form	frmProduct	รายการสินค้า
TextBox1	txtProductNo	รหัสสินค้า
TextBox2	txtProductName	รายละเอียด
ComboBox1	cboProductType	ประเภทสินค้า
TextBox3	txtCostPrice	ราคาทุน
TextBox4	txtSalePrice	ราคาขาย
TextBox5	txtQTY	จำนวนในคลัง

จากนั้นให้เขียนชุดคำสั่งที่เหตุการณ์ Form Load ด้วยการดับเบิ้ลคลิกที่ฟอร์มแล้วพิมพ์คำสั่ง call ConnectDatabase() เพื่อเชื่อมต่อกับฐานข้อมูล และคำสั่ง call ShowData() เพื่อเรียกใช้โพรซีเยอร์ที่ชื่อ ShowData มาใช้งาน ซึ่งในฟอร์มข้อมูลสินค้านี้โพรซีเยอร์ ShowData จะใช้สำหรับการเลือกข้อมูลจากตาราง PRODUCT มาแสดงในคอนโทรลที่ได้สร้างขึ้นมาก่อนหน้านี้ ให้นักศึกษาเขียนชุดคำสั่งเพิ่มดังภาพ



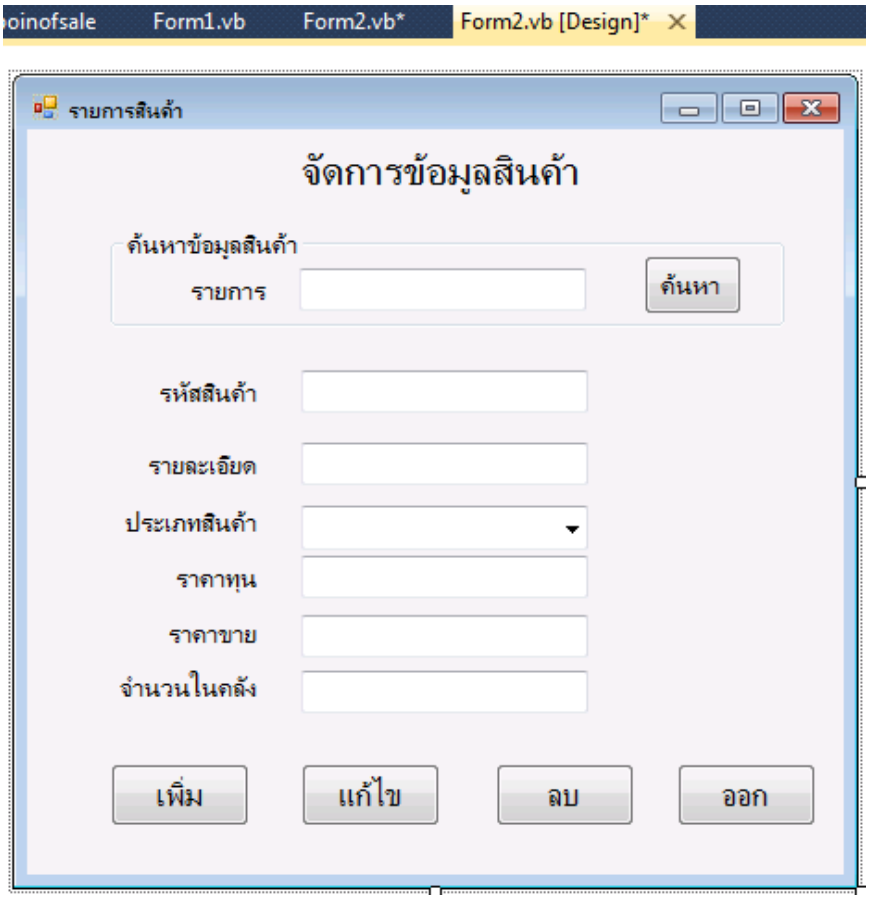

```
11      Private Sub ShowData()
12            gsql = "Select * from PRODUCT"
13            With cmd
14                  .CommandType = CommandType.Text
15                  .CommandText = gsql
16                  .Connection = Cnn
17                  dr = .ExecuteReader
18            End With
19            dr.Read()
20            If dr.HasRows Then
21                  txtProductNo.Text = dr.Item("ProductNo").ToString
22                  txtProductName.Text = dr.Item("ProductName").ToString
23                  cboProductType.Text = dr.Item("ProductTypeNo").ToString
24                  txtCostPrice.Text = dr.Item("CostPrice").ToString
25                  txtSalePrice.Text = dr.Item("SalePrice").ToString
26                  txtQTY.Text = dr.Item("QTY").ToString
27            End If
28            dr.Close()
29      End Sub
30 End Class
```

การเพิ่มการทำงานให้กับฟอร์มข้อมูลสินค้า

จากนี้เราจะเพิ่มการทำงานให้กับฟอร์มข้อมูลสินค้า โดยการสร้างปุ่มการทำงานเพื่อให้ฟอร์มนี้สามารถจัดการกับข้อมูลที่อยู่ในตาราง Product ผ่านหน้าฟอร์มข้อมูลสินค้านี้ ซึ่งประกอบด้วยการทำงานดังนี้

1. ส่วนการเพิ่มข้อมูลลงในตารางข้อมูล
2. การค้นหาข้อมูลจากตารางข้อมูล
3. ส่วนการแก้ไขข้อมูลแล้วปรับปรุงข้อมูลลงในตารางอีกครั้ง
4. การสบข้อมูลออกจากตารางข้อมูล

ให้นักศึกษาเพิ่มปุ่มการทำงานดังภาพ



กำหนดคุณสมบัติของคอนโทรล ดังนี้

Control	Properties - Name	Properties – Text
TextBox	txtSearch	รายการ (ในส่วนค้นหา)
button	btnSearch	ค้นหา
button	btnAdd	เพิ่ม
button	btnEdit	แก้ไข
button	btnDelete	ลบ
button	btnClose	ออก

การสร้างโพรซีเยอร์สำหรับนำข้อมูลจากตาราง PRODUCT_TYPE มาใช้ใน Combobox

จากที่เราได้ทำการจัดเก็บข้อมูลประเภทสินค้าในตาราง PRODUCT_TYPE ซึ่งตาราง PRODUCT_TYPE ใช้สำหรับการอ้างอิงข้อมูล เราจึงสามารถนำข้อมูลส่วนนี้มาใช้เพื่อกำหนดเป็นรายละเอียดของข้อมูลสินค้าได้โดยการสร้างโพรซีเยอร์ในการดึงข้อมูลจากตาราง PRODUCT_TYPE มาใส่ใน ComboBox ตามขั้นตอนดังนี้

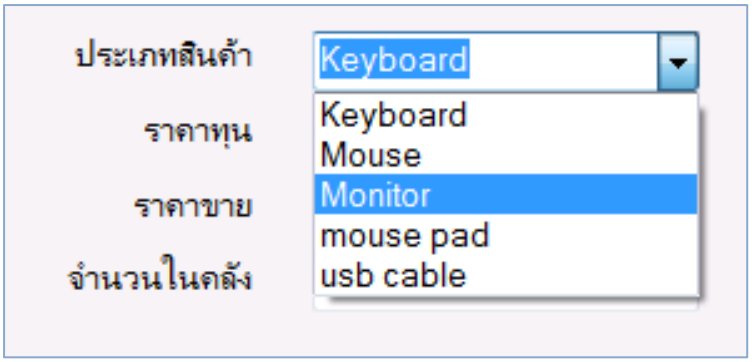
1. สร้าง Sub ชื่อ FillCombo
2. เขียนคำสั่งใน sub FillCombo ดังภาพ

```
31      Private Sub FillCombo()
32            gsql = "select * From Product_Type"
33            Dim da As New OleDbDataAdapter(gsql, Cnn)
34            Dim dt As New DataTable
35            da.Fill(dt)
36            With cboProductType
37                  .DataSource = dt
38                  .DisplayMember = dt.Columns.Item("ProductTypeName").ColumnName
39                  .ValueMember = dt.Columns.Item("ProductTypeNo").ColumnName
40            End With
41      End Sub
42
43  End Class
```

3. เรียกใช้ Sub FillCombo จากเหตุการณ์ Form_Load ในภาพตัวอย่างจะเป็นคำสั่งบรรทัดที่ 9

```
 7      Private Sub frmProducts_Load(ByVal sender As System
System.EventArgs) Handles MyBase.Load
 8            Call ConnectDatabase()
 9            Call FillCombo()
10            Call ShowData()
11       End Sub
```

4. ผลลัพธ์ของการเรียกใช้ FillCombo จะเป็นการนำชื่อประเภทสินค้าทั้งหมดมาแสดงใน ComboBox ที่ชื่อ cboProductType



การแสดงประเภทสินค้าจากในฟอร์มข้อมูลสินค้า

ตาราง Product เป็นตารางเก็บข้อมูลสินค้า ซึ่งมีเขตข้อมูลหนึ่งที่ชื่อว่า ProductTypeNo เขตข้อมูลนี้จะเก็บรหัสของประเภทสินค้า โดยรหัสประเภทสินค้านี้จะอ้างไปถึงชื่อประเภทสินค้าที่อยู่ในตาราง PRODUCT_TYPE ฉะนั้นถ้าเขียนโปรแกรมเพื่อแสดงข้อมูลจากตาราง Product ทั้งหมด ที่ช่องข้อมูลประเภทสินค้าจะแสดงเป็นรหัสประเภทสินค้า แต่เพื่อให้ผู้ใช้งานโปรแกรมเข้าใจได้ง่ายขึ้น เราจึงควรปรับส่วนของการ

แสดงประเภทสินค้าให้เป็นชื่อประเภทสินค้า โดยปรับการใช้คำสั่งที่โพรซีเยอร์ ShowData ในส่วนชุดคำสั่งที่ใช้แสดงข้อมูลจากตาราง Product ให้นักศึกษาปรับคำสั่งเป็นดังภาพ

```
12      Private Sub ShowData()
13            gsql = "Select * from PRODUCT"
14            With cmd
15                  .CommandType = CommandType.Text
16                  .CommandText = gsql
17                  .Connection = Cnn
18                  dr = .ExecuteReader
19            End With
20            dr.Read()
21            If dr.HasRows Then
22                  txtProductNo.Text = dr.Item("ProductNo").ToString
23                  txtProductName.Text = dr.Item("ProductName").ToString
24                  cboProductType.SelectedValue = dr.Item("ProductTypeNo").ToString   'บรรทัดที่ปรับคำสั่งใหม่
25                  txtCostPrice.Text = dr.Item("CostPrice").ToString
26                  txtSalePrice.Text = dr.Item("SalePrice").ToString
27                  txtQTY.Text = dr.Item("QTY").ToString
28            End If
29            dr.Close()
30      End Sub
```

การสร้างคำสั่งสำหรับบันทึกข้อมูลใหม่ (ปุ่มเพิ่ม)

การบันทึกข้อมูลใหม่จะใช้คำสั่ง SQL Insert โดยนำค่าต่างๆ ที่ป้อนใน Textbox ไปเพิ่มในระเบียนใหม่ในตาราง Product ให้นักศึกษาเขียนชุดคำสั่งที่ปุ่ม “เพิ่ม” ดังนี้

```
      Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnAdd.Click

            gsql = "Insert Into Product Values ("
            gsql += "'" & txtProductNo.Text & "',"
            gsql += "'" & txtProductName.Text & "',"
            gsql += "'" & cboProductType.SelectedValue & "',"
            gsql += "'" & txtCostPrice.Text & "',"
            gsql += "'" & txtSalePrice.Text & "',"
            gsql += "'" & txtQTY.Text & "') "
            With cmd
                  .CommandType = CommandType.Text
                  .CommandText = gsql
                  .Connection = Cnn
                  .ExecuteNonQuery()
            End With
            MessageBox.Show("เพิ่มรายการใหม่แล้ว", "เพิ่มข้อมูล", MessageBoxButtons.OK,
MessageBoxIcon.Information)

      End Sub
```

เมื่อทดสอบโปรแกรมและสามารถทำการเพิ่มข้อมูลสินค้าลงในตารางได้แล้ว ให้นักศึกษาเพิ่มชุดคำสั่ง เพื่อป้องกันการป้อนข้อมูลซ้ำ ด้วยคำสั่ง Try . . . Catch เข้าไปดังภาพ

```
44      Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    btnAdd.Click
45          Try
46                  gsql = "Insert Into Product Values ("
47                  gsql += "'" & txtProductNo.Text & "',"
48                  gsql += "'" & txtProductName.Text & "',"
49                  gsql += "'" & cboProductType.SelectedValue & "',"
50                  gsql += "'" & txtCostPrice.Text & "',"
51                  gsql += "'" & txtSalePrice.Text & "',"
52                  gsql += "'" & txtQTY.Text & "') "
53                  With cmd
54                        .CommandType = CommandType.Text
55                        .CommandText = gsql
56                        .Connection = Cnn
57                        .ExecuteNonQuery()
58                  End With
59                  MessageBox.Show("เพิ่มรายการใหม่แล้ว", "เพิ่มข้อมูล", MessageBoxButtons.OK,
    MessageBoxIcon.Information)
60          Catch ex As Exception
61                  MessageBox.Show("มีรหัสสินค้านี้แล้ว ไม่สามารถเพิ่มข้อมูลได้", "เพิ่มข้อมูล", MessageBoxButtons.OK,
    MessageBoxIcon.Information)
62          End Try
63      End Sub
64
```

การสร้างคำสั่งสำหรับการปรับปรุงข้อมูล (ปุ่มแก้ไข)

คำสั่งในการปรับปรุงข้อมูลจะใช้คำสั่ง SQL Update โดยการนำค่าต่างๆ ที่อยู่ในตารางข้อมูลมาปรับปรุงผ่าน Textbox และนำกลับไปปรับปรุงลงในระเบียนข้อมูลเดิมที่มีอยู่ในตาราง สามารถเขียนคำสั่งที่ปุ่มแก้ไข (btnEdit) ดังนี้

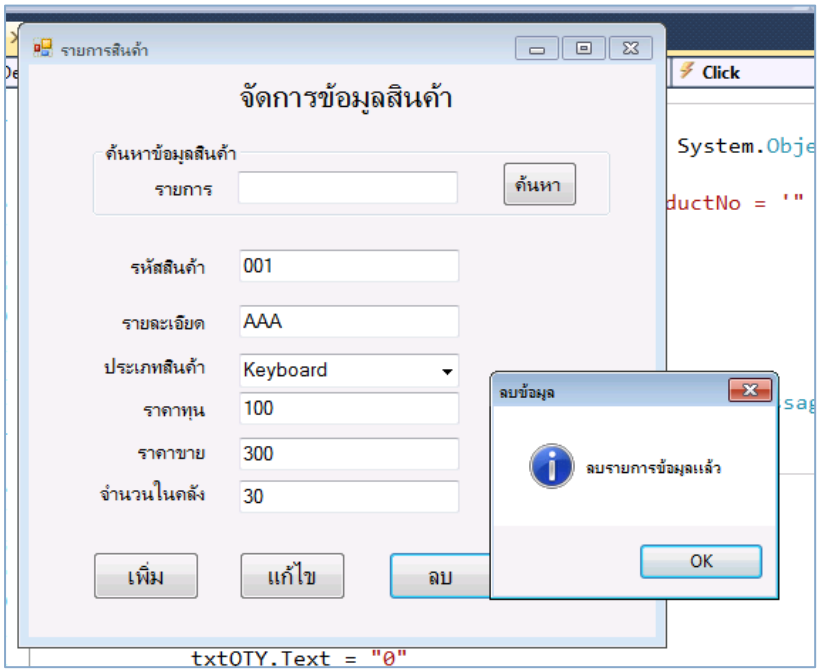
```
64
65      Private Sub btnEdit_Click(ByVal sender As System.Object, ByVal e As System.Even
    btnEdit.Click
66
67              gsql = "Update Product Set "
68              gsql += "ProductName = '" & txtProductName.Text & "',"
69              gsql += "ProductTypeNo = '" & cboProductType.SelectedValue & "',"
70              gsql += "CostPrice = '" & txtCostPrice.Text & "',"
71              gsql += "SalePrice = '" & txtSalePrice.Text & "',"
72              gsql += "QTy = '" & txtQTY.Text & "'"
73              gsql += " where ProductNo = '" & txtProductNo.Text & "'"
74              With cmd
75                    .CommandType = CommandType.Text
76                    .CommandText = gsql
77                    .Connection = Cnn
78                    .ExecuteNonQuery()
79              End With
80              MessageBox.Show("แก้ไขรายการแล้ว", "แก้ไขข้อมูล", MessageBoxButtons.OK,
    MessageBoxIcon.Information)
81      End Sub
82
```

การสร้างคำสั่งสำหรับปุ่มลบข้อมูล

การลบข้อมูลจะใช้คำสั่ง SQL Delete โดยใช้เงื่อนไขคือระเบียนที่ต้องการลบต้องเป็นระเบียนที่มีค่าของเขตข้อมูลตรงกับข้อมูลที่อยู่ใน txtProductNo เนื่องจากการลบข้อมูลส่วนมากจะใช้คีย์หลักเป็นตัวกำหนดระเบียนที่ต้องการลบ โดยสามารถเขียนคำสั่งในปุ่มลบได้ดังนี้

```
104
105     Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As System.EventAr
    btnDelete.Click
106             gsql = "delete from Product where ProductNo = '" & txtProductNo.Text & "'"
107             With cmd
108                     .CommandType = CommandType.Text
109                     .CommandText = gsql
110                     .Connection = Cnn
111                     .ExecuteNonQuery()
112             End With
113             MessageBox.Show("ลบรายการข้อมูลแล้ว", "ลบข้อมูล", MessageBoxButtons.OK,
    MessageBoxIcon.Information)
114
115     End Sub
```

เมื่อทดสอบการคลิกปุ่มลบ จะมี message box แจ้งให้ทราบว่า “ลบรายการข้อมูลแล้ว” และเมื่อคลิกปุ่ม OK แล้ว ข้อมูลในหน้าฟอร์มสินค้ายังคงค้างอยู่ที่หน้าจอโปรแกรม แต่ในความเป็นจริงแล้วข้อมูลได้ถูกลงออกจากตารางไปแล้ว



การปรับปรุงปรุงชุดคำสั่งเพื่อป้องกันการผิดพลาด

- การสร้างกล่องโต้ตอบเพื่อยืนยันการลบข้อมูล เนื่องจากการลบข้อมูลนี้ เมื่อลบข้อมูลไปแล้วจะไม่สามารถเรียกข้อมูลที่ถูกลบกลับมาได้ ดังนั้นจึงควรมีการยืนยันก่อนเกิดการลบจากผู้ใช้งาน ว่าต้องการลบข้อมูลหรือไม่ การปรับปรุงชุดคำสั่งในการสร้างกล่องโต้ตอบเพื่อยืนยันการลบข้อมูลจากผู้ใช้ โดยเพิ่มคำสั่งในบรรทัดที่ 106 – 108 และ บรรทัดที่ 117 ดังภาพ

```
104
105     Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Ha
    btnDelete.Click
106         Dim ans As MsgBoxResult
107         ans = MessageBox.Show("ต้องการลบรายการหรือไม่", "ยืนยันการลบ", MessageBoxButtons.YesNo,
    MessageBoxIcon.Question)
108         If ans = MsgBoxResult.Yes Then
109             gsql = "delete from Product where ProductNo = '" & txtProductNo.Text & "'"
110             With cmd
111                 .CommandType = CommandType.Text
112                 .CommandText = gsql
113                 .Connection = Cnn
114                 .ExecuteNonQuery()
115             End With
116             MessageBox.Show("ลบรายการข้อมูลแล้ว", "ลบข้อมูล", MessageBoxButtons.OK,
    MessageBoxIcon.Information)
117         End If
118
119     End Sub
```

- การล้างค่าในคอนโทรลต่างๆ เพื่อไม่ให้ข้อมูลเก่าที่ลบไปแล้วค้างอยู่ที่หน้าจอ สามารถทำได้โดยการสร้างโพรซีเยอร์เพื่อล้างข้อมูลดังนี้

```
120     Private Sub clearText()
121         txtProductNo.Text = ""
122         txtProductName.Text = ""
123         cboProductType.SelectedValue = ""
124         txtCostPrice.Text = "0"
125         txtSalePrice.Text = "0"
126         txtQTY.Text = "0"
127         txtProductNo.Focus()
128     End Sub
```

จากคำสั่งข้างต้นเป็นการสร้างโพรซีเยอร์ชื่อ clearText ซึ่งทำหน้าที่ในการกำหนดค่าเริ่มต้นที่จะแสดงบนคอนโทรลต่างๆ ให้เป็นค่าว่าง โดยให้นักศึกษากลับไปเพิ่มคำสั่งเรียกใช้โพรซีเยอร์ ClearText ที่ปุ่มลบ ดังภาพ

```
105      Private Sub btnDelete_Click(ByVal sender As System.
         btnDelete.Click
106             Dim ans As MsgBoxResult
107             ans = MessageBox.Show("ต้องการลบรายการหรือไม่", "ยื
         MessageBoxIcon.Question)
108             If ans = MsgBoxResult.Yes Then
109                   gsql = "delete from Product where Produ
110                   With cmd
111                         .CommandType = CommandType.Text
112                         .CommandText = gsql
113                         .Connection = Cnn
114                         .ExecuteNonQuery()
115                   End With
116                   MessageBox.Show("ลบรายการข้อมูลแล้ว", "ลบข้อม
         MessageBoxIcon.Information)
117             End If
118             Call clearText()
119      End Sub
```

การสร้างคำสั่งสำหรับปุ่มออก

ปุ่มออก จะใช้เป็นปุ่มสำหรับปิดหน้าจอ เราสามารถใช้คำสั่งได้โดยดับเบิ้ลคลิกที่ปุ่มออก และเขียนคำสั่ง `Me.Close()`