



Materi Perkuliahan

Pertemuan 06

PEMROGRAMAN VISUAL

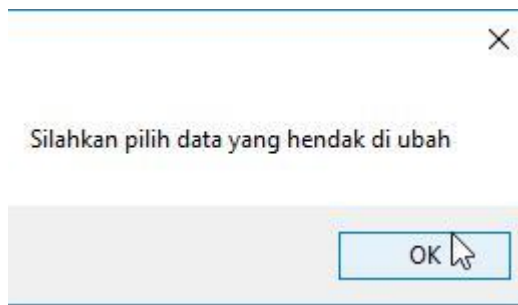
Semester Genap 2020/2021

Edit dan Delete Data

ALUR SKENARIO PROSES EDIT / UBAH DATA

Alur mengubah data adalah sebagai berikut:

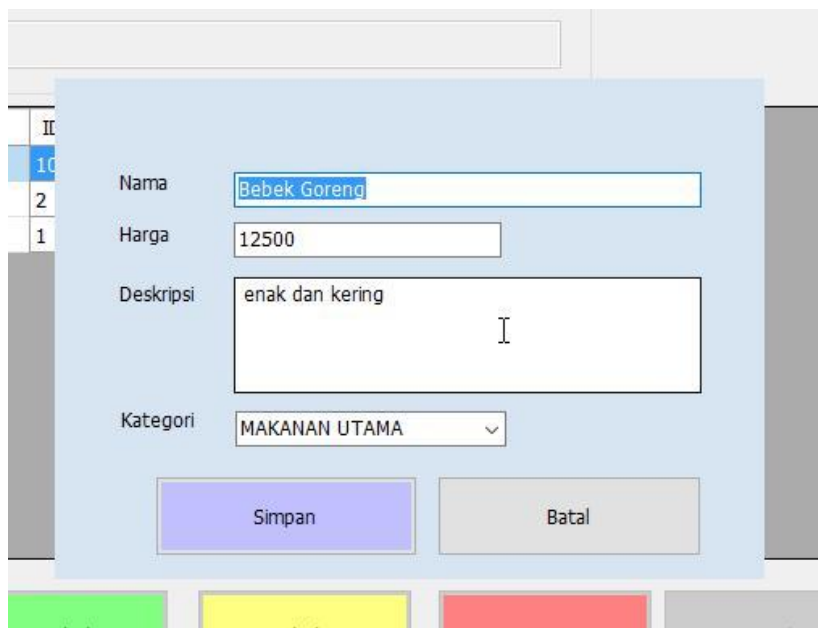
Saat pertama kali Form pengelolaan barang (Form1.cs) dijalankan maka sebelum kita bisa mengedit data kita harus memilih data mana yang akan di edit. Data akan dipilih dari datagrid view dengan meng klik data di datagridview atau menggunakan tombol Page up (↑) atau page down (↓) di keyboard. Selanjutnya jika data sudah terpilih, tekan tombol Ubah. Apabila data belum terpilih dan kita menekan tombol ubah akan muncul peringatan seperti berikut:



Selanjutnya pilihlah data di datagridview yang ditunjukkan dengan baris yang sudah terseleksi seperti contoh berikut:

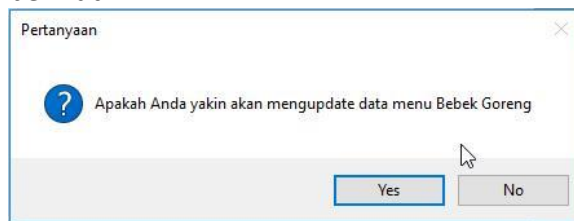
	ID	Nama	Harga	Deskripsi	Kategori
▶	1003	Bebek Goreng	12500	enak dan kering	MAKANAN...
	2	Pizza	25000	italiano yahun	MAKANAN...
	1	Pisang gorgor	7500	enak sekali rasanya	SNACK

Apabila data sudah terpilih, maka tekan tombol Ubah. Maka akan muncul user Control yang menampilkan detil data dari menu yang bisa kita rubah.



Nama	Bebek Goreng
Harga	12500
Deskripsi	enak dan kering
Kategori	MAKANAN UTAMA
<div>Simpan</div> <div>Batal</div>	

Setelah mengubah data dengan benar, tekan tombol Simpan, maka akan keluar peringatan seperti berikut:



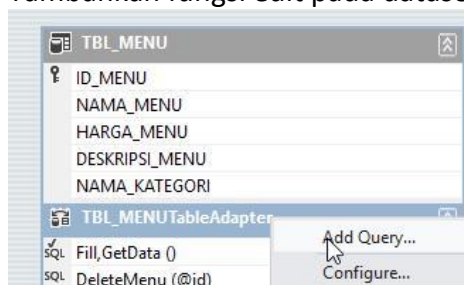
Jika yakin, maka tekan tombol yes, maka kita akan kembali ke form utama dan data akan terupdate

	ID	Nama	Harga	Deskripsi	Kategori
▶	1003	Bebek Goreng Pedas	12500	enak dan kering	MAKANAN...
	2	Pizza	25000	italiano yahun	MAKANAN...
	1	Pisang gorgor	7500	enak sekali rasanya	SNACK

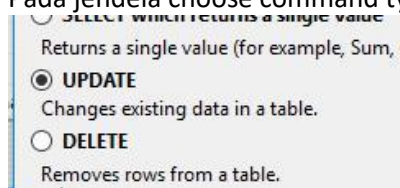
Untuk mewujudkan skenario di atas maka kita akan mengikuti langkah langkah sebagai berikut:

MEMBUAT FUNGSI EDIT PADA DATASET

Tambahkan fungsi edit pada dataset dsResto



Pada jendela choose command type pilih “Use SQL statements” , kemudian pada “Choose a Query Type” pilih



Pada specify a SQL Update statement, buatlah Query sbb:

Specify a SQL UPDATE statement

The UPDATE statement will be used by the query.



Type your SQL statement or use the Query Builder to construct it. What data should be loaded into the table?

What data should the table load?

```
UPDATE TBL_MENU
SET NAMA_MENU = @nama, HARGA_MENU = @harga, DESKRIPSI_MENU = @deskripsi, ID_KATEGORI = @kategori
WHERE (ID_MENU = @id)
```

Beri nama "UpdateMenu" dan OK

Choose Function Name

Choose the name of the func

What would you like to name th

UpdateMenu

Method sudah bertambah



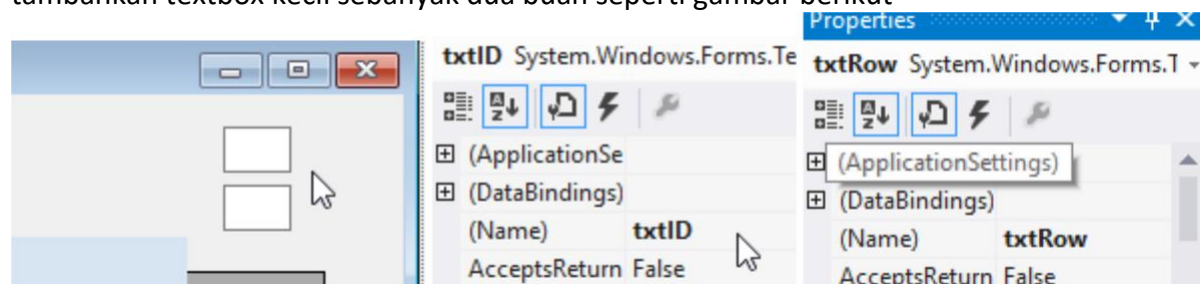
MENAMBAHKAN METHOD PADA MENU CONTROL

Masuk ke MenuControl.CS dan tambahkan method untuk mengubah data sebagai berikut:

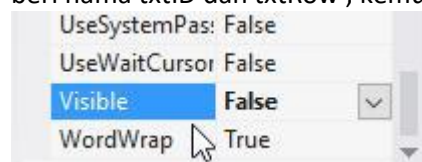
```
public void editMenu(Menu M, int idmenu)
{
    TM.UpdateMenu(M>Nama, M.Harga, M.Deskripsi, M.Kategori, idmenu);
}
```

MENAMBAHKAN CODE EDIT DI FORM1.CS

Kita kembali ke Form1.cs. Kita akan menambahkan beberapa code pada bagian ini. Pertama tambahkan textbox kecil sebanyak dua buah seperti gambar berikut



beri nama txtID dan txtRow , kemudian melalui properties, set visible keduanya sama dengan false



Kedua textbox tersebut akan digunakan untuk menampung id menu dari menu yg akan kita ubah, dan menampung index dari baris data menu tersebut didatagrid

MENAMBAHKAN EVENT MEMILIH DATA PADA DATAGRID VIEW

Selanjutnya kita akan masuk ke bagian code dari Form1.cs dengan menekan F7. Pada skenario pengubahan data, sebelum mengedit data, kita harus memilih data terlebih dahulu dengan memilihnya dari datagrid view. Terkait dengan itu, kita tambahkan fungsi untuk mengambil data dari datagridview pada bagian code dari form1.cs sbb:

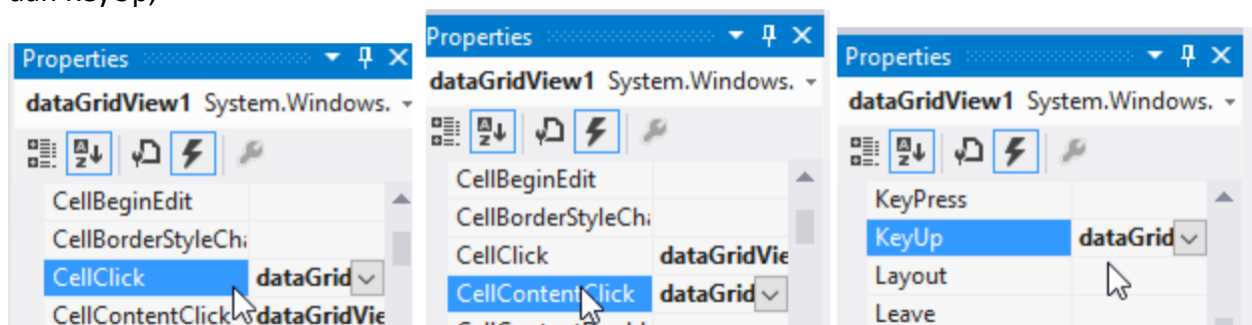
```
private string getKolom(DataGridView dg, int i)
{
    return dg[dg.Columns[i].Index, dg.CurrentRow.Index].Value.ToString();
}

private string getKolomEdit(DataGridView dg, int i)
{
    return dg[dg.Columns[0].Index, dg.Rows[i].Index].Value.ToString();
}

private string getRow(DataGridView dg)
{
    return dg.CurrentRow.Index.ToString();
}
```

Fungsi getKolom berguna untuk mendapatkan data dari kolom yang kita pilih sedangkan getRow adalah fungsi untuk mengembalikan nomer baris yang kita pilih.

Selanjutnya adalah menambahkan tiga event pada datagridview kita yaitu cellClick, cellContentClick dan KeyUp,



Kemudian pada ketiga event yang muncul di bagian code dari Form1.cs, tambah kan code yang sama seperti contoh berikut:

```
private void dataGridview1_CellClick(object sender, DataGridViewCellEventArgs e)
{
    txtID.Text = getKolom(dataGridview1,0);
    txtRow.Text = getRow(dataGridview1);
}

private void dataGridview1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    txtID.Text = getKolom(dataGridview1,0);
    txtRow.Text = getRow(dataGridview1);
}

private void dataGridview1_KeyUp(object sender, KeyEventArgs e)
{
    txtID.Text = getKolom(dataGridview1,0);
    txtRow.Text = getRow(dataGridview1);
}
```

```

private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    txtID.Text = getKolom(dataGridView1, 0);
    txtRow.Text = getRow(dataGridView1);
}

private void dataGridView1_KeyUp(object sender, KeyEventArgs e)
{
    txtID.Text = getKolom(dataGridView1, 0);
    txtRow.Text = getRow(dataGridView1);
}

```

Ketiga event di atas memiliki kegunaan yang sama, yaitu mendapatkan id dari data menu yang kita pilih dengan mendapatkan nilai dari kolom pada indeks ke 0 (Kolom pertama yaitu ID) dan mendapatkan nomer baris data tersebut.

Selanjutnya masih pada bagian code dari form1.cs, tambahkan method berikut:

```

public void EnableEdit()
{
    txtCari.Enabled = true;
    dataGridView1.Enabled = true;
    btnTambah.Enabled = true;
    btnUbah.Enabled = true;
    btnHapus.Enabled = true;
    btnBatal.Enabled = true;

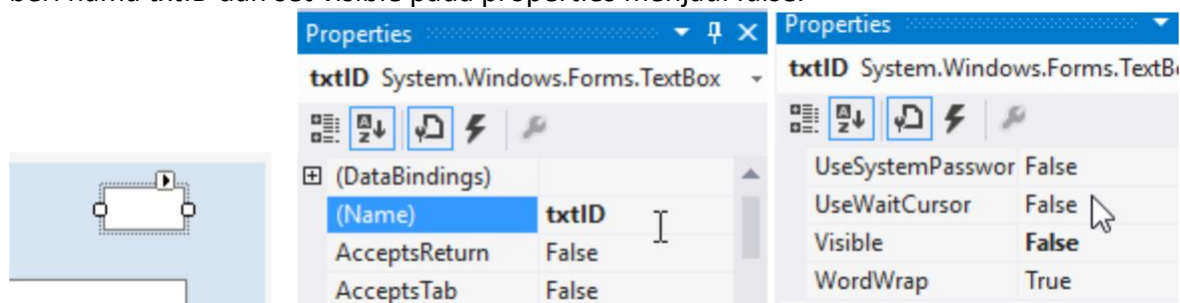
    setDatagridview(this.dataGridView1);
    dataGridView1.Rows[int.Parse(txtRow.Text)].Selected = true;
    txtID.Text = getKolomEdit(dataGridView1, int.Parse(txtRow.Text));
}

```

Pada dasarnya fungsi EnableEdit() tidak berbeda jauh dengan fungsi Enabled sebelumnya. Bedanya fungsi ini akan dipanggil setelah fungsi mengubah data selesai dan akan menyeleksi pada datagridview data yang baru saja di edit.

MENAMBAHKAN KODE UNTUK EDIT PADA UC_MENU.CS

Kita akan masuk ke bagian design dari UC_MENU.CS. Pada bagian ini tambahkan textbox kecil dari beri nama txtID dan set visible pada properties menjadi false.



Textbox ini berfungsi untuk menampung id dari menu yang akan diedit.

Selanjutnya kita akan masuk ke bagian code dari UC_Menu.cs. Pada bagian ini tambahkan class variabel dan method sbb:


```

string temp_menu = "";
public void isiTextBox(string nama, string harga, string deskripsi, string kategori, string id)
{
    txtNama.Text = nama;
    temp_menu = nama;
    txtHarga.Text = harga;
    txtDeskripsi.Text = deskripsi;
    cmbKategori.Text = kategori;
    txtID.Text = id;
}

```

temp_menu dipergunakan untuk menyimpan nama menu sebelum di edit, sedangkan fungsi isiTextBox digunakan untuk menerima inputan data yang dipilih dari Form1.cs untuk ditampilkan di UC_Menu.cs

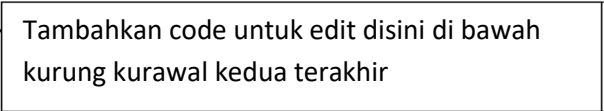
Selanjutnya kita akan melihat event pada button simpan. Sebelumnya pada code tersebut sudah terdapat code untuk menginsert data seperti berikut:

```

private void btnSimpan_Click(object sender, EventArgs e)
{
    if (flagperintah == 1) //tambah data
    {
        if (cektxt() == true)
        {
            errorProvider1.Clear();

            int IDKategori = MC.getIDKategori(cmbKategori.Text);
            Restoran.Entity.Menu M = new Entity.Menu(txtNama.Text, txtDeskripsi.Text, double.Parse(txtHarga.Text), IDKategori);
            MC.addMenu(M);
            cleartxt();
            this.Hide();
            Form1 myParent = (Form1)this.Parent;
            myParent.Enable();
        }
    }
}

```



Tambahkan code didalamnya sebagai bagian dari else dari if yang sudah ada sebelumnya sbb:

```

else
{
    if (cektxt() == true)
    {
        errorProvider1.Clear();

        int IDKategori = MC.getIDKategori(cmbKategori.Text);
        Restoran.Entity.Menu M = new Entity.Menu(txtNama.Text, txtDeskripsi.Text, double.Parse(txtHarga.Text), IDKategori);

        DialogResult dr = MessageBox.Show("Apakah Anda yakin akan mengupdate data menu " + temp_menu, "Pertanyaan",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if (dr == DialogResult.Yes)
        {
            MC.editMenu(M, int.Parse(txtID.Text));
        }
        cleartxt();
        this.Hide();
        Form1 myParent = (Form1)this.Parent;
        myParent.EnableEdit();
    }
}

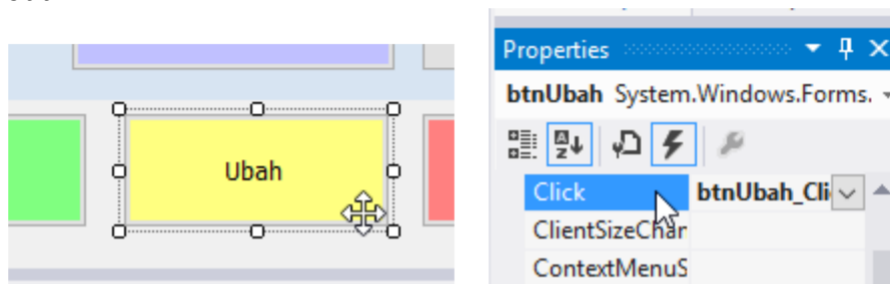
```

Logika dari code program di atas adalah, jika flag perintah tidak sama dengan 1 (yg artinya edit), kita akan cek apakah textbox ada yg kosong tidak (dengan cektxt()). Kemudian kita buat objek Menu M (sama dengan penjelasan pada edit). Kemudian kita buat dialog result yang akan memastikan apakah kita akan mengedit data menu, jika ya, maka kode: MC.editMenu(M,int.parse(txtID.text)); akan di

eksekusi, yaitu mengedit nilai dari menu dengan atribut dari objek M dan id sesuai dengan yang disimpan pada txtID. Karena parameter id pada fungsi editMenu adalah int, maka kita harus mengconvert txtID.text menjadi integer menggunakan perintah: `int.Parse(string)`.

Setelah selesai langkah selanjutnya sama saja dengan bagian edit, yaitu mengclear text box di UC_MENU, menghide user control ini, dan memanggil parent form dari UC_MENU yaitu Form dan memanggil fungsi `EnableEdit()`;

Selanjutnya kita kembali ke Form1.cs dan pada button ubah di Form1.cs , tambahkan event click sbb:



```
private void btnUbah_Click(object sender, EventArgs e)
{
}

```

Tambahkan code didalamnya sebagai berikut:

```
private void btnUbah_Click(object sender, EventArgs e)
{
    if (txtID.Text == "")
    {
        MessageBox.Show("Silahkan pilih data yang hendak di ubah");
        dataGridView1.Focus();
    }
    else
    {
        uC_Menu1.setFlag(2);

        string nama = getKolom(dataGridView1, 1);
        string harga = getKolom(dataGridView1, 2);
        string deskripsi = getKolom(dataGridView1, 3);
        string kategori = getKolom(dataGridView1, 4);
        uC_Menu1.isiTextBox(nama, harga, deskripsi, kategori, txtID.Text);
        uC_Menu1.Visible = true;
        txtID.Clear();
        disable();
    }
}

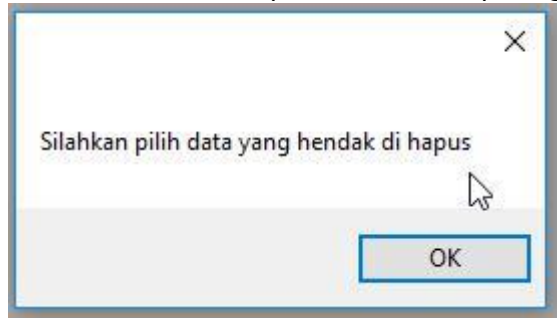
```


Alur logika dari code diatas adalah sebagai berikut:

Jika data belum dipilih (yang artinya txtID.text == "") maka akan keluar peringatan untuk memilih data. Jika sudah dipilih, kita akan menset flag dari uc_Menu1 sama dengan 2 yang artinya perintah untuk edit. Selanjutnya kita akan mendapatkan data detail Menu dari baris yang kita pilih dari datagridview menggunakan fungsi getKolom yang sudah kita buat sebelumnya dan mengirimkan nilai tersebut ke uc_Menu1 melalui fungsi isiTextBox. Kemudian menampilkan uc_Menu1 dan mengclear txtID di form1.cs dan mendisablenya.

ALUR SKENARIO PROSES DELETE /HAPUS DATA

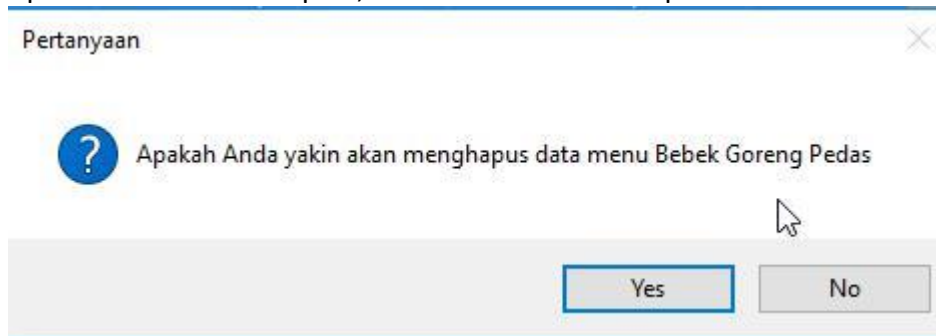
Alur menghapus data adalah sebagai berikut, pilih dahulu data yang akan dihapus, jika kita langsung menekan tombol hapus akan keluar peringatan sbb:



Selanjutnya pilih menu yang hendak dihapus yang ditandai dengan dengan baris yang sudah terseleksi seperti contoh berikut:

	ID	Nama	Harga	Deskripsi	Kategori
▶	1003	Bebek Goreng	12500	enak dan kering	MAKANAN...
	2	Pizza	25000	italiano yahun	MAKANAN...
	1	Pisang gorgor	7500	enak sekali rasanya	SNACK

Apabila data sudah terpilih, maka tekan tombol hapus dan akan muncul konfirmasi peringatan sbb:



Jika Iya, tekan YES maka data yang dipilih akan terhapus

Pencarian

	ID	Nama	Harga	Deskripsi	Kategori
▶	2	Pizza	25000	italiano yahun	MAKANAN...
	1	Pisang gorgor	7500	enak sekali rasanya	SNACK

Langkah untuk mewujudkan code diatas adalah sebagai berikut:

BUAT METHOD HAPUS DATA PADA DATASET TBL_MENU

Langkah nya sama, pada TBL_MENUDataAdapter pilih add query dan Pada jendela choose command type pilih "Use SQL statements" , kemudian pada "Choose a Query Type" pilih DELETE

☐ UPDATE
 Changes existing data in a table.

☒ DELETE
 Removes rows from a table.

Kemudian selanjutnya tambahkan query delete sbb:

Specify a SQL DELETE statement

The DELETE statement will be used by the query

Type your SQL statement or use the Query Builder

What data should the table load?

```
DELETE FROM TBL_MENU
WHERE (ID_MENU = @id)
```

Setelah selesai beri nama "DeleteMenu"

Choose Function Name

Choose the name of the function

What would you like to name the new function?

DeleteMenu

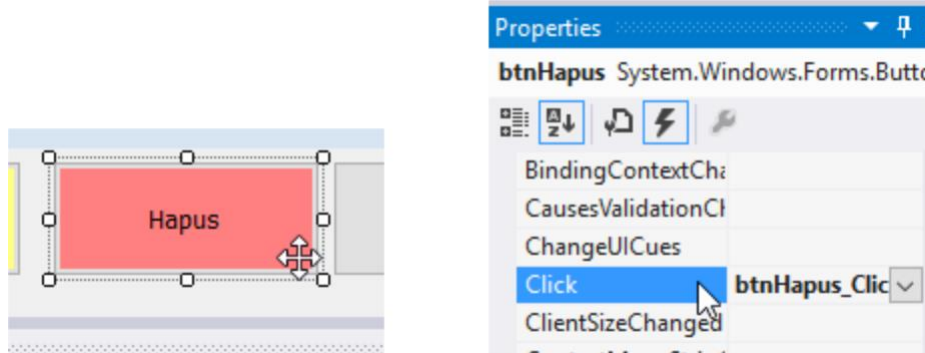
MENAMBAHKAN METHOD HAPUS PADA MENU CONTROL

Masuk ke MenuControl.CS dan tambahkan method untuk menghapus data sebagai berikut:

```
public void deleteMenu(int idmenu)
{
    TM.DeleteMenu(idmenu);
}
```

MENAMBAHKAN CODE HAPUS DI FORM1.CS

Selanjutnya kembali ke Form1.cs dan tambahkan event click pada button hapus sbb:



Maka akan muncul event click pada bagian code dari Form1 sbb:

```
private void btnHapus_Click(object sender, EventArgs e)
{
}
```

Didalamnya tambahkan code sebagai berikut:

```
private void btnHapus_Click(object sender, EventArgs e)
{
    if (txtID.Text == "")
    {
        MessageBox.Show("Silahkan pilih data yang hendak di hapus");
        dataGridView1.Focus();
    }
    else
    {
        DialogResult dr = MessageBox.Show("Apakah Anda yakin akan menghapus data menu "
            + getKolom(dataGridView1, 1),
            "Pertanyaan", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        if (dr == DialogResult.Yes)
        {
            MC.deleteMenu(int.Parse(txtID.Text));
        }
        txtID.Clear();
        this.Enable();
    }
}
```

Logika codenya adalah sebagai berikut:

Jika data di datagridview belum terpilih (ditunjukkan dengan txtID.text == "") maka akan keluar peringatan. Jika sudah terpilih, maka munculkan dialog result untuk memastikan bahwa anda yakin akan menghapus data tersebut. Jika setuju hapus, maka dipanggil fungsi untuk menghapus data yaitu MC.deleteMenu dengan parameternya adalah id dari menu yang hendak kita hapus. Setelah itu kita clearkan program kita dan panggil fungsi enable.

MODIFIKASI FUNGSI ENABLE UNTUK MENCEGAH ERROR JIKA DATA YANG DIHAPUS TINGGAL SATU SAJA Kita hendak memodifikasi fungsi enable pada form1.cs agar jika data tinggal satu dan kita hapus tidak akan error karena pada kode hapus tsb, kita memanggil fungsi Enable() yang didalamnya terdapat code

```
dataGridView1.Rows[0].Selected = true;
txtID.Text = getKolom(dataGridView1, 0);
```

Jika data terakhir kita hapus, maka code di atas akan menyebabkan error karena data baris pertama (rows index ke 0) sudah tidak ada. Maka untuk mencegah tidak error, tambahkan sbb:

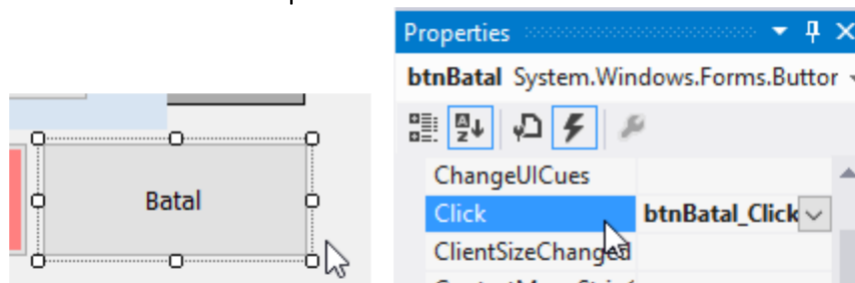
```
public void Enable()
{
    txtCari.Enabled = true;
    dataGridView1.Enabled = true;
    btnTambah.Enabled = true;
    btnUbah.Enabled = true;
    btnHapus.Enabled = true;
    btnBatal.Enabled = true;

    setDataGridView(this.dataGridView1);
    if (dataGridView1.RowCount > 0)
    {
        dataGridView1.Rows[0].Selected = true;
        txtID.Text = getKolom(dataGridView1, 0);
    }
}
```

Dengan menambahkan kondisi, yaitu jika jumlah baris pada datagrid view lebih besar dari 0, maka kita mencegah program memilih row yang tidak ada. Jadi Row akan dipilih jika ada data minimal satu pada datagrid view.

MENAMBAHKAN EVENT PADA BUTTON BATAL

Tambahkan event click pada button batal sbb:



Akan muncul event click dan tambahkan kode didalamnya yang memanggil fungsi Enable() sbb:

```
private void btnBatal_Click(object sender, EventArgs e)
{
    this.Enable();
}
```

Cukup sampai disini.. bersambung.. thanks 😊