

Laporan Pembuatan Aplikasi Manajemen Marketplace PETONA



Oleh

Nazwa Dafa Ramadhan Arifin Putra	20051397023
Mohammad Dzakiyyul Ashfiya' El Arif	20051397031
Gagah Rizky Mulyawan	20051397045

Program Studi Manajemen Informatika

Fakultas Vokasi

Universitas Negeri Surabaya

2022

Deskripsi Aplikasi

PETONA Desktop Application merupakan sebuah aplikasi desktop yang dirancang untuk memperlancar proses manajerial pada website PETONA, melalui aplikasi desktop ini nantinya seluruh gambaran proses yang terjadi pada website PETONA dapat dipantau dengan mudah dan juga semua hasil dari proses-proses yang terjadi dapat digambarkan melalui sebuah laporan yang secara otomatis tersedia pada aplikasi desktop ini nantinya. PETONA Desktop Application juga memungkinkan merekap seluruh feedback masukan dari para pengguna dimana hal ini akan berguna sebagai landasan dalam pengembangan website PETONA kedepannya.

Teknis penggunaan aplikasi

PETONA Desktop Application terdiri dari 3 bagian yaitu :

1. Home

Pada bagian home ini admin dari petona dapat mengakses menu input barang, pemesanan, dan pelanggan :

a. Menu input barang

Pada menu ini admin dapat memasukkan barang apa saja yang nantinya siap untuk diperjual-belikan dengan para user website PETONA nantinya dan juga memasukkan jumlah ketersediaan produk.

b. Menu pemesanan

Pada menu ini admin dapat mengisi form yang ada untuk mengisikan pemesanan apa saja yang telah dilakukan oleh user website PETONA, melalui menu juga nanti akan dihasilkan laporan keuangan yang menunjukkan semua hasil proses bisnis yang terjadi melalui website PETONA ini.

c. Menu pelanggan

Pada menu ini admin dapat melihat pengguna yang telah masuk dan membuat akun pada website PETONA, admin juga dapat memonitor pengguna telah melakukan pembelian apa saja pada marketplace website PETONA.

2. Shop

Pada bagian menu ini akan menampilkan semua produk-produk yang telah muncul pada website marketplace PETONA dan jumlah ketersediaan produk, jika semisal terdapat produk

yang telah habis ketersediaannya maka admin dapat menghilangkan produk tersebut dari marketplace nya, pada menu shop juga semua harga dari produk ditampilkan sehingga jika terdapat human error dimana terjadi salah input harga dapat terdeteksi secepat mungkin, dan pada menu ini juga akan ditampilkan jumlah ketersediaan produk saat itu.

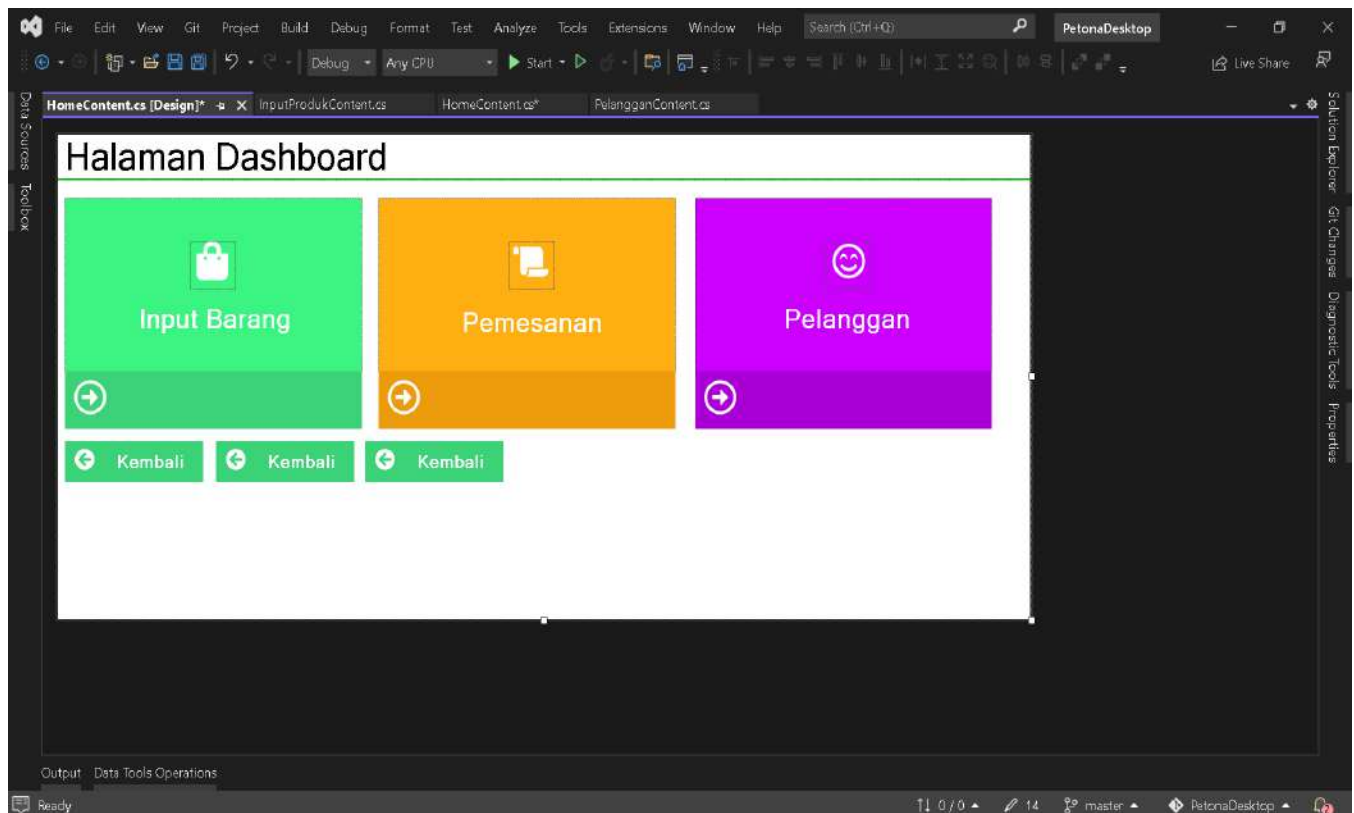
3. Feedback

Pada bagian menu ini akan ditampilkan semua saran dan masukan yang diberikan user website, melalui saran dan masukan pada feedback ini nantinya akan mempermudah dalam proses pengembangan website kedepan.

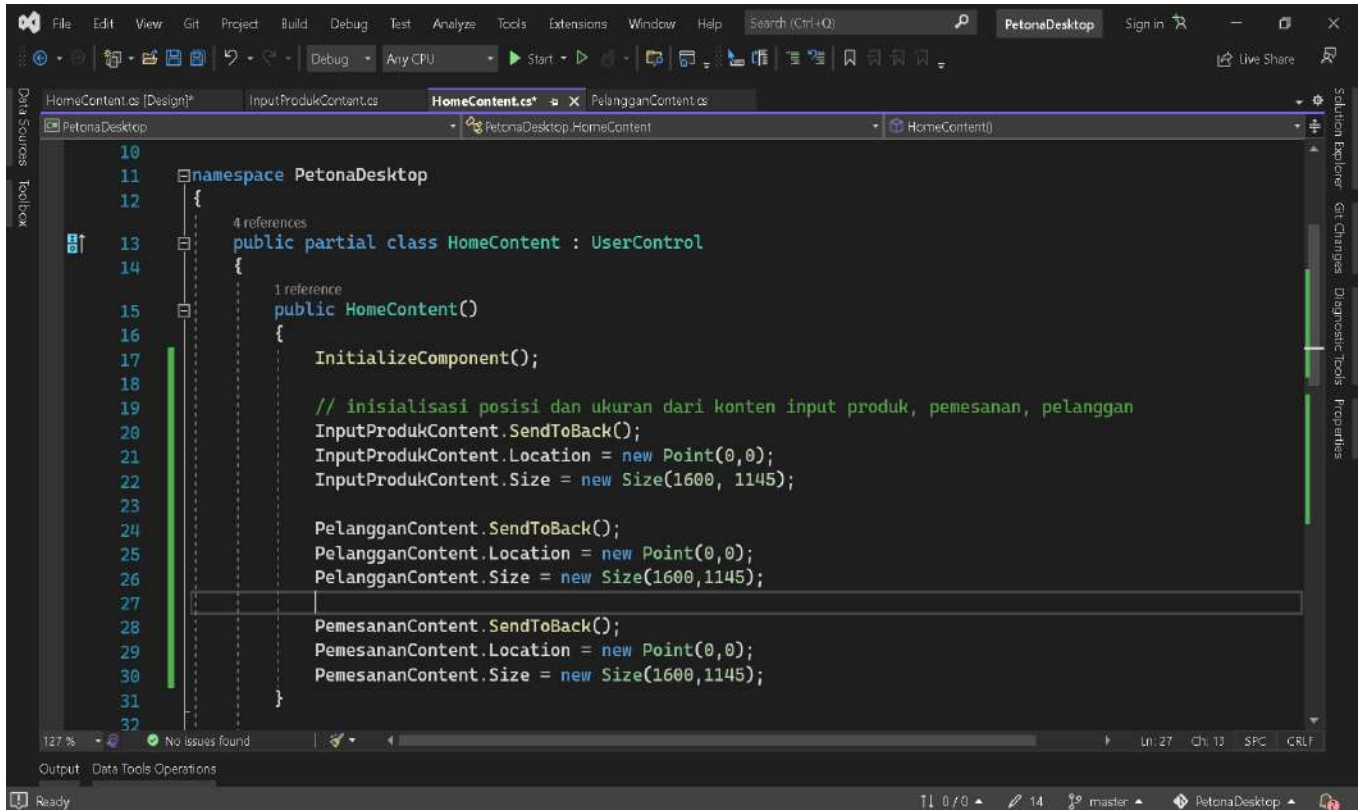
Kontribusi Setiap Anggota

1. Nazwa Dafa Ramadhan Arifin Putra

Membuat halaman home yang berisikan 3 pilihan yaitu Input Barang, Pemesanan, dan Pelanggan

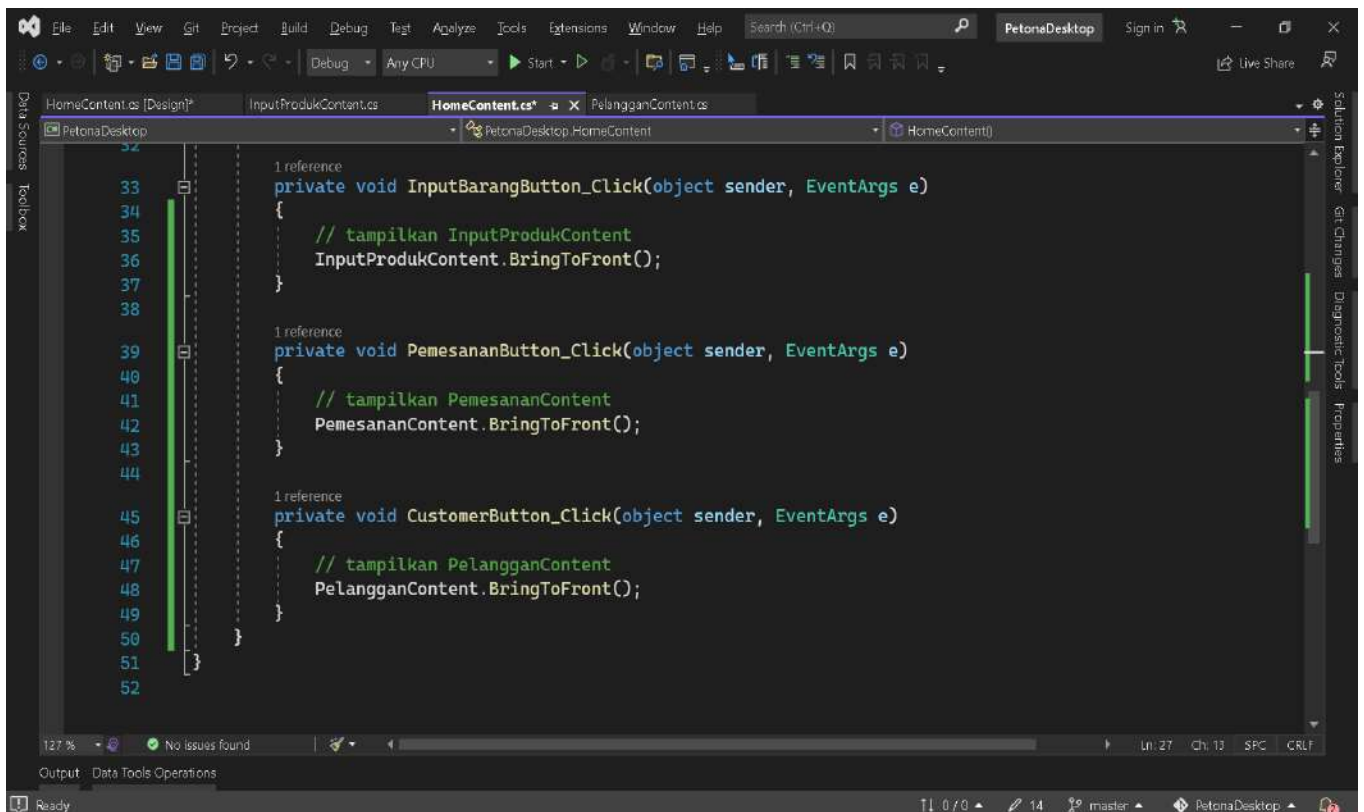


Inisialisasi Posisi dan ukuran konten Input Produk, Pemesanan, dan Pelanggan



```
10
11 namespace PetonaDesktop
12 {
13     4 references
14     public partial class HomeContent : UserControl
15     {
16         1 reference
17         public HomeContent()
18         {
19             InitializeComponent();
20
21             // inisialisasi posisi dan ukuran dari konten input produk, pemesanan, pelanggan
22             InputProdukContent.SendToBack();
23             InputProdukContent.Location = new Point(0,0);
24             InputProdukContent.Size = new Size(1600, 1145);
25
26             PelangganContent.SendToBack();
27             PelangganContent.Location = new Point(0,0);
28             PelangganContent.Size = new Size(1600,1145);
29
30             PemesananContent.SendToBack();
31             PemesananContent.Location = new Point(0,0);
32             PemesananContent.Size = new Size(1600,1145);
33         }
34     }
35 }
```

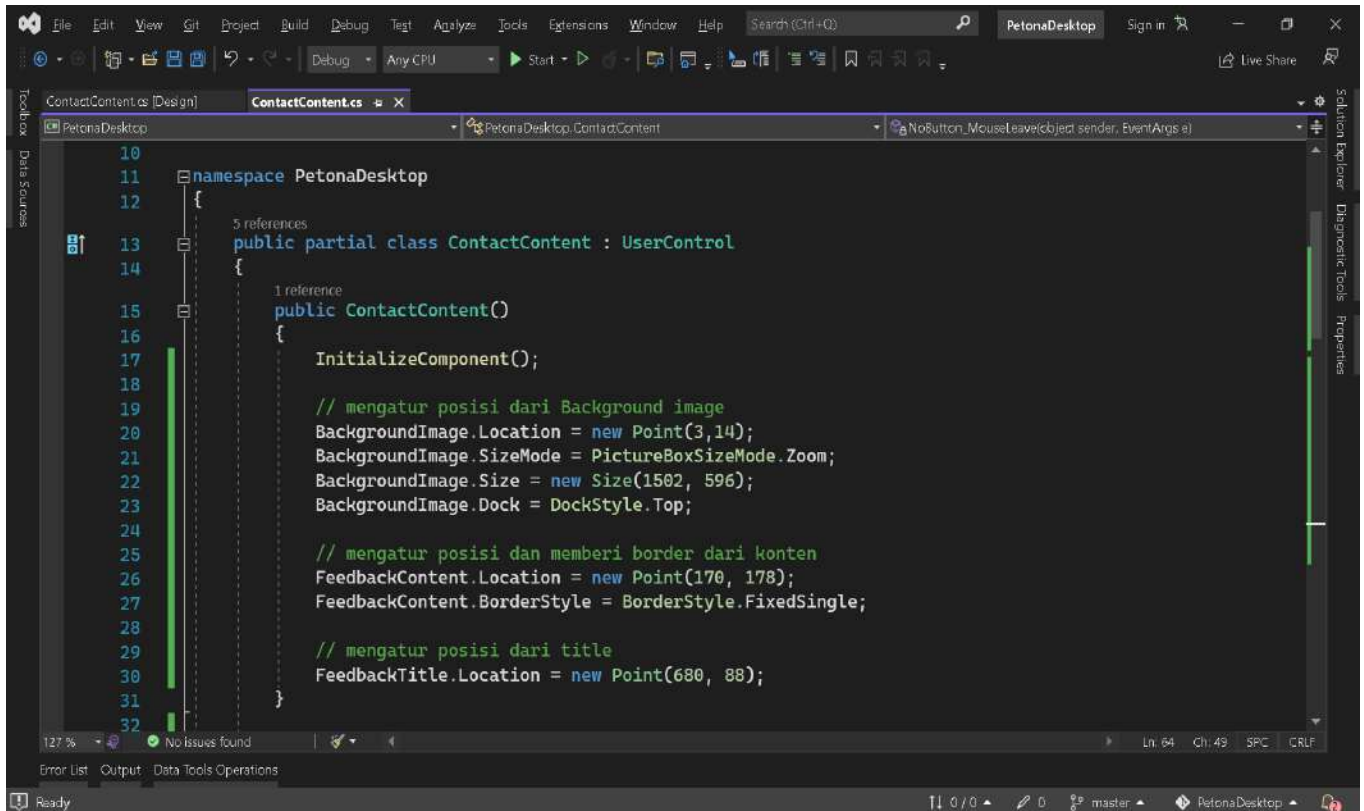
Ketika panel Input barang, Pemesanan, atau Pelanggan ditekan maka akan membuka masing-masing tampilan dengan mengakses user control baru



```
33
34
35     1 reference
36     private void InputBarangButton_Click(object sender, EventArgs e)
37     {
38         // tampilkan InputProdukContent
39         InputProdukContent.BringToFront();
40     }
41
42     1 reference
43     private void PemesananButton_Click(object sender, EventArgs e)
44     {
45         // tampilkan PemesananContent
46         PemesananContent.BringToFront();
47     }
48
49     1 reference
50     private void CustomerButton_Click(object sender, EventArgs e)
51     {
52         // tampilkan PelangganContent
53         PelangganContent.BringToFront();
54     }
55 }
```

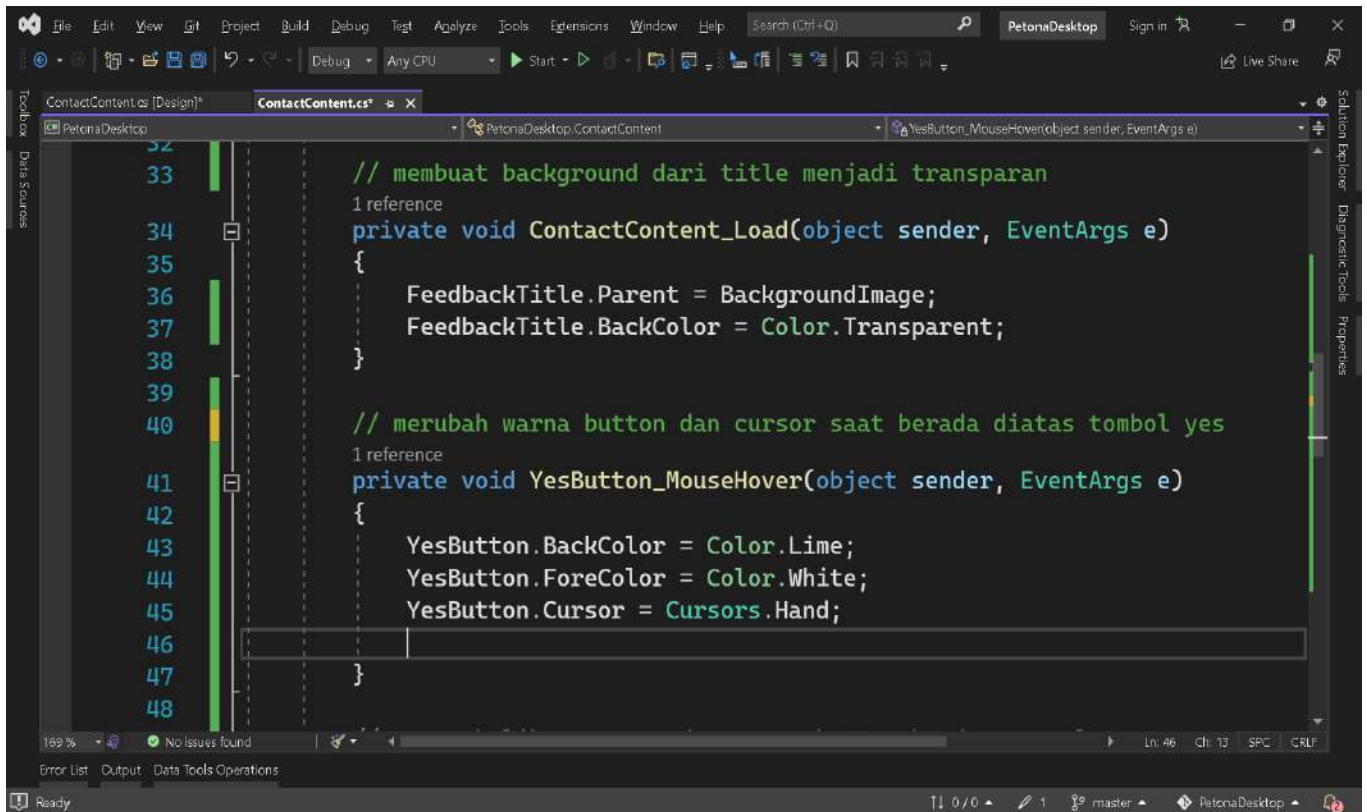
2. Gagah Rizky Mulyawan

Mengatur posisi dari background image, memberi border dari konten, dan title



```
10 namespace PetonaDesktop
11 {
12     5 references
13     public partial class ContactContent : UserControl
14     {
15         1 reference
16         public ContactContent()
17         {
18             InitializeComponent();
19
20             // mengatur posisi dari Background image
21             BackgroundImage.Location = new Point(3, 14);
22             BackgroundImage.SizeMode = PictureBoxSizeMode.Zoom;
23             BackgroundImage.Size = new Size(1502, 596);
24             BackgroundImage.Dock = DockStyle.Top;
25
26             // mengatur posisi dan memberi border dari konten
27             FeedbackContent.Location = new Point(170, 178);
28             FeedbackContent.BorderStyle = BorderStyle.FixedSingle;
29
30             // mengatur posisi dari title
31             FeedbackTitle.Location = new Point(680, 88);
32         }
33     }
```

Membuat background dari title menjadi transparan dan merubah warna button dan cursor saat berada diatas tombol yes

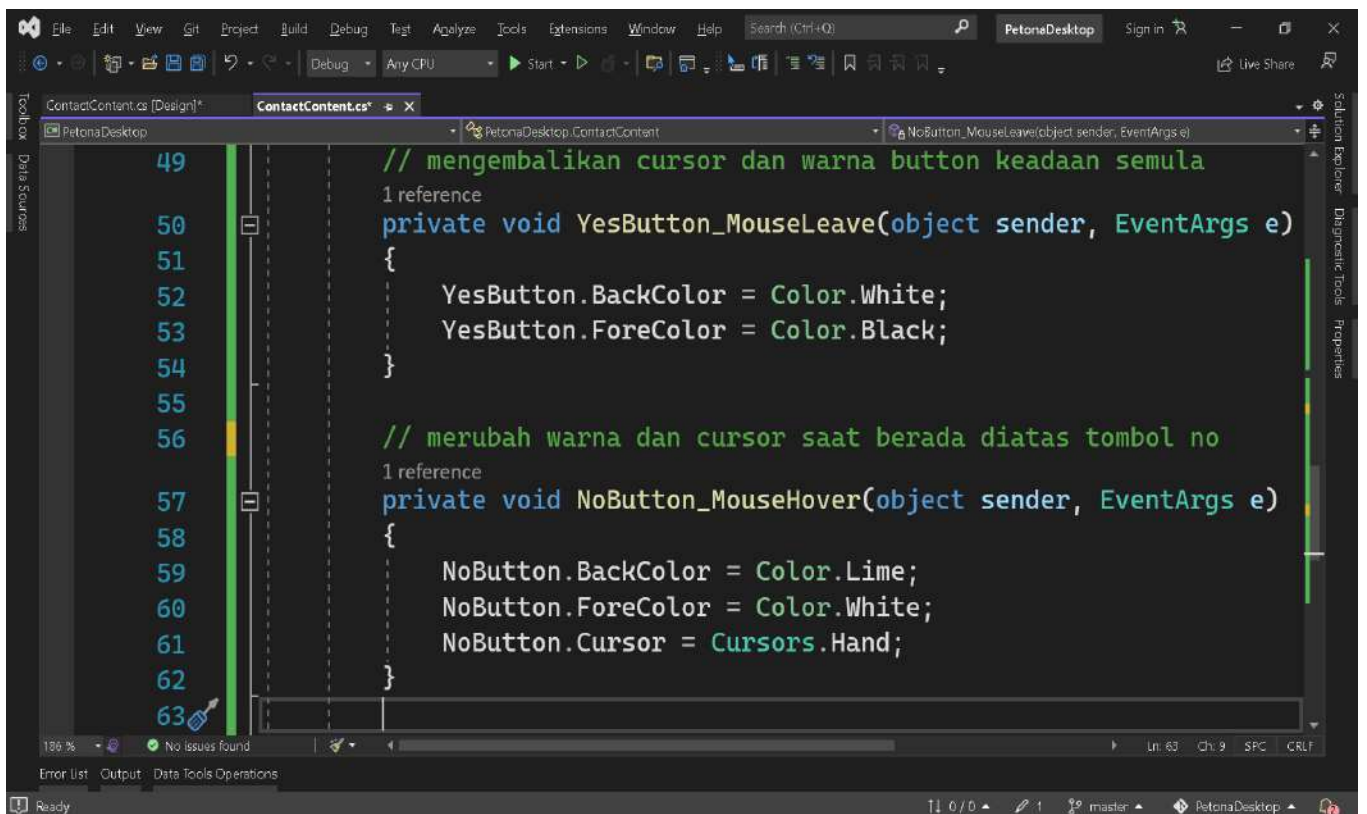


The screenshot shows the Visual Studio IDE with a C# file named `ContactContent.cs` open. The code is written in a dark theme. The `ContactContent_Load` method sets the parent of `FeedbackTitle` to `BackgroundImage` and sets its background color to `Color.Transparent`. The `YesButton_MouseHover` method sets the background color of `YesButton` to `Color.Lime`, the foreground color to `Color.White`, and the cursor to `Cursors.Hand`. The interface includes a menu bar, a toolbar, a Solution Explorer on the right, and a status bar at the bottom.

```
// membuat background dari title menjadi transparan
1 reference
private void ContactContent_Load(object sender, EventArgs e)
{
    FeedbackTitle.Parent = BackgroundImage;
    FeedbackTitle.BackColor = Color.Transparent;
}

// merubah warna button dan cursor saat berada diatas tombol yes
1 reference
private void YesButton_MouseHover(object sender, EventArgs e)
{
    YesButton.BackColor = Color.Lime;
    YesButton.ForeColor = Color.White;
    YesButton.Cursor = Cursors.Hand;
}
```

Mengembalikan cursor dan warna button keadaan semula, merubah warna dan cursor saat berada diatas tombol no

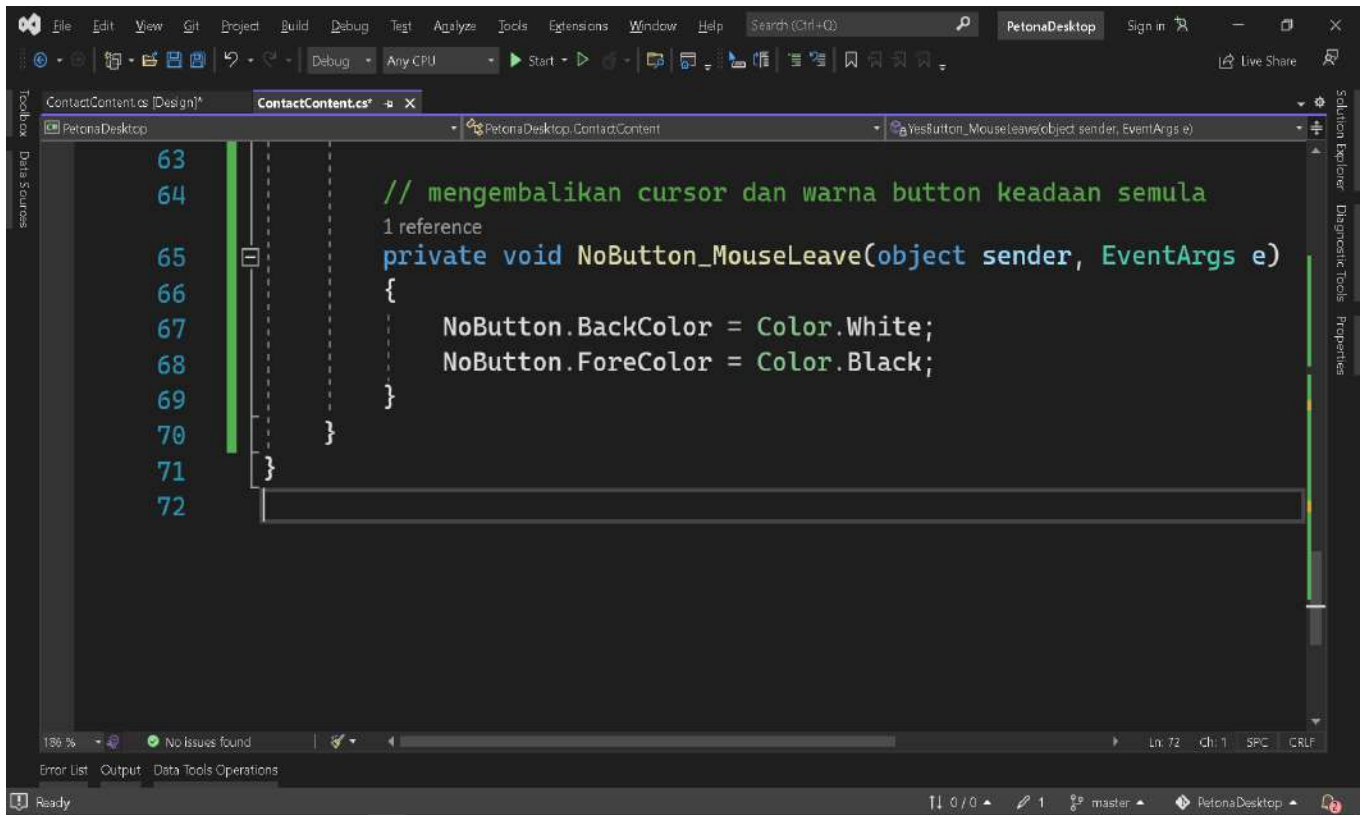


The screenshot shows the Visual Studio IDE with a C# file named `ContactContent.cs` open. The code is written in a dark theme. The `YesButton_MouseLeave` method sets the background color of `YesButton` back to `Color.White` and the foreground color back to `Color.Black`. The `NoButton_MouseHover` method sets the background color of `NoButton` to `Color.Lime`, the foreground color to `Color.White`, and the cursor to `Cursors.Hand`. The interface includes a menu bar, a toolbar, a Solution Explorer on the right, and a status bar at the bottom.

```
// mengembalikan cursor dan warna button keadaan semula
1 reference
private void YesButton_MouseLeave(object sender, EventArgs e)
{
    YesButton.BackColor = Color.White;
    YesButton.ForeColor = Color.Black;
}

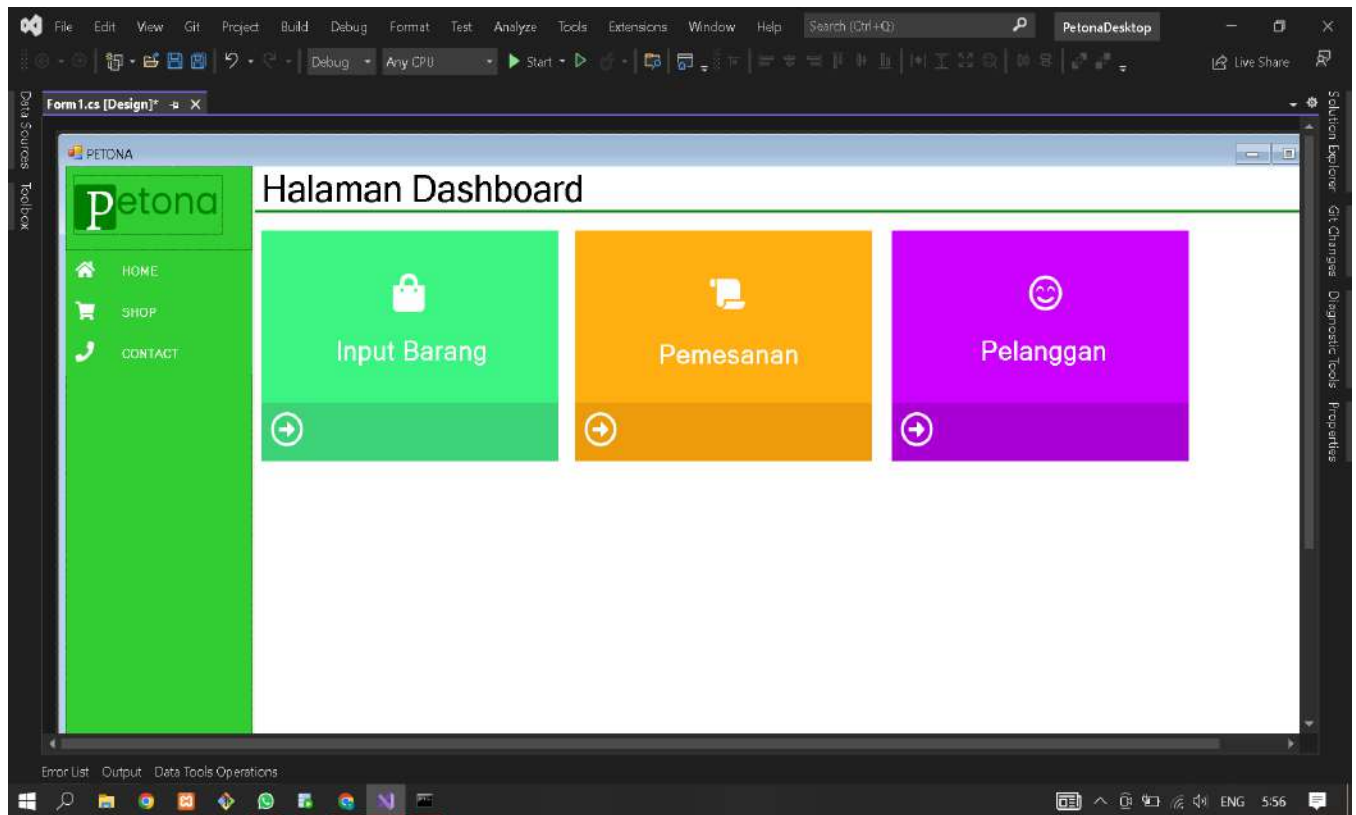
// merubah warna dan cursor saat berada diatas tombol no
1 reference
private void NoButton_MouseHover(object sender, EventArgs e)
{
    NoButton.BackColor = Color.Lime;
    NoButton.ForeColor = Color.White;
    NoButton.Cursor = Cursors.Hand;
}
```

Mengembalikan cursor dan warna button keadaan semula

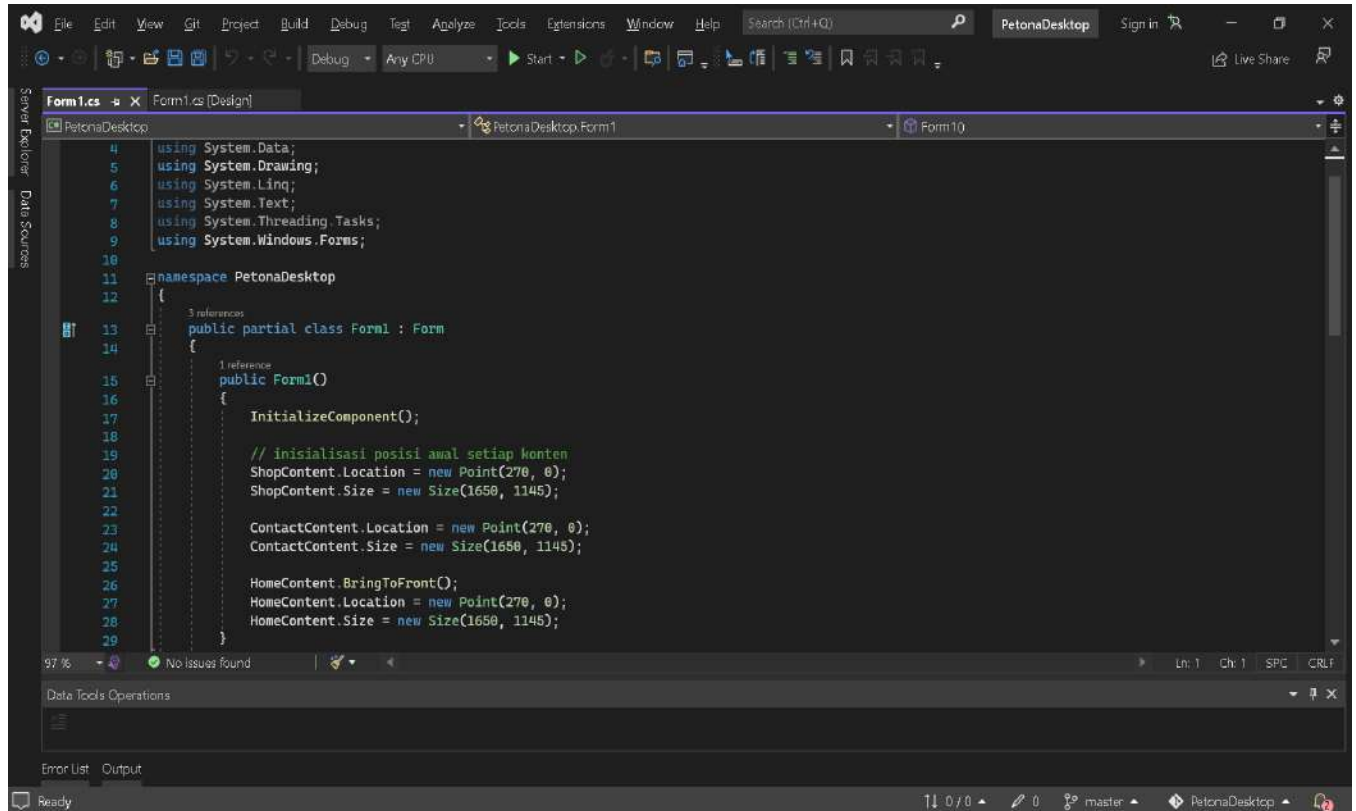


3. Mohammad Dzakiyyul Ashfiya' El Arif

Membuat Navbar Panel yang berisikan 1 picturebox dan 3 button untuk Home, Shop, dan Contact

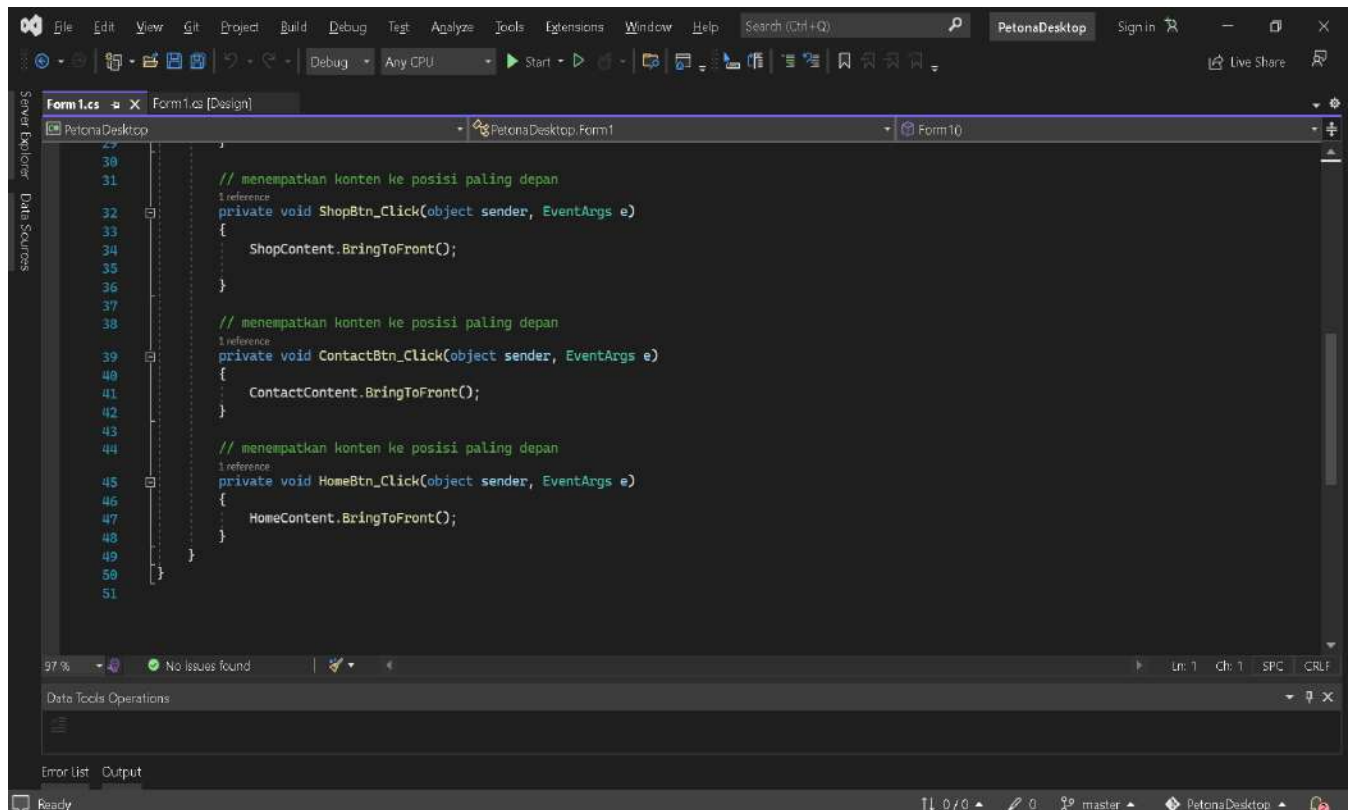


Inisialisasi posisi awal setiap konten agar posisi dan ukuran konten memenuhi tepat disamping navbar panel.



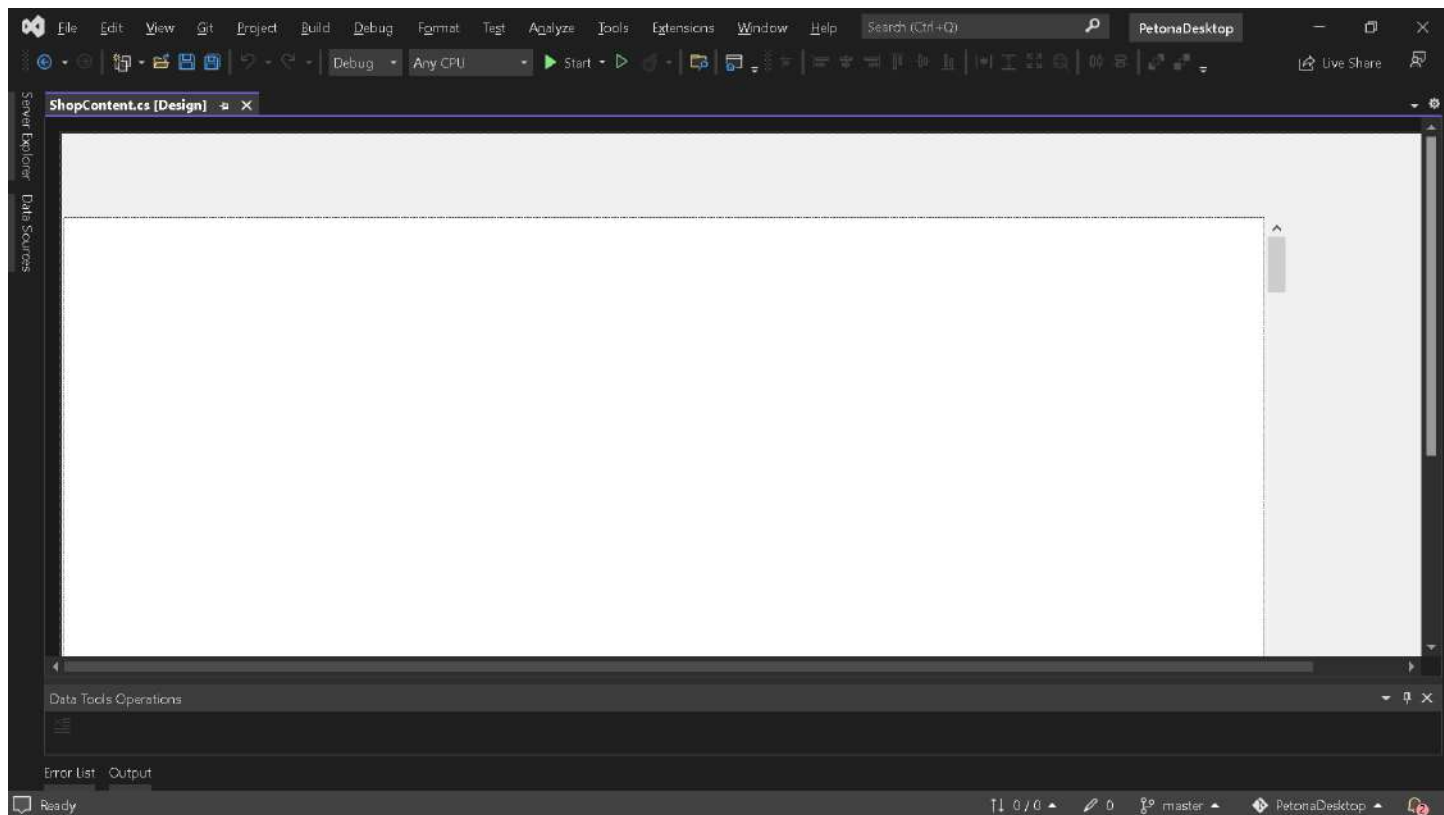
```
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace PetonaDesktop
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18
19             // inisialisasi posisi awal setiap konten
20             ShopContent.Location = new Point(270, 0);
21             ShopContent.Size = new Size(1650, 1145);
22
23             ContactContent.Location = new Point(270, 0);
24             ContactContent.Size = new Size(1650, 1145);
25
26             HomeContent.BringToFront();
27             HomeContent.Location = new Point(270, 0);
28             HomeContent.Size = new Size(1650, 1145);
29         }
30     }
31 }
```

3 tombol klik untuk Home, Shop, dan Contact untuk membuat active content pada halaman sesuai tombol yang diklik

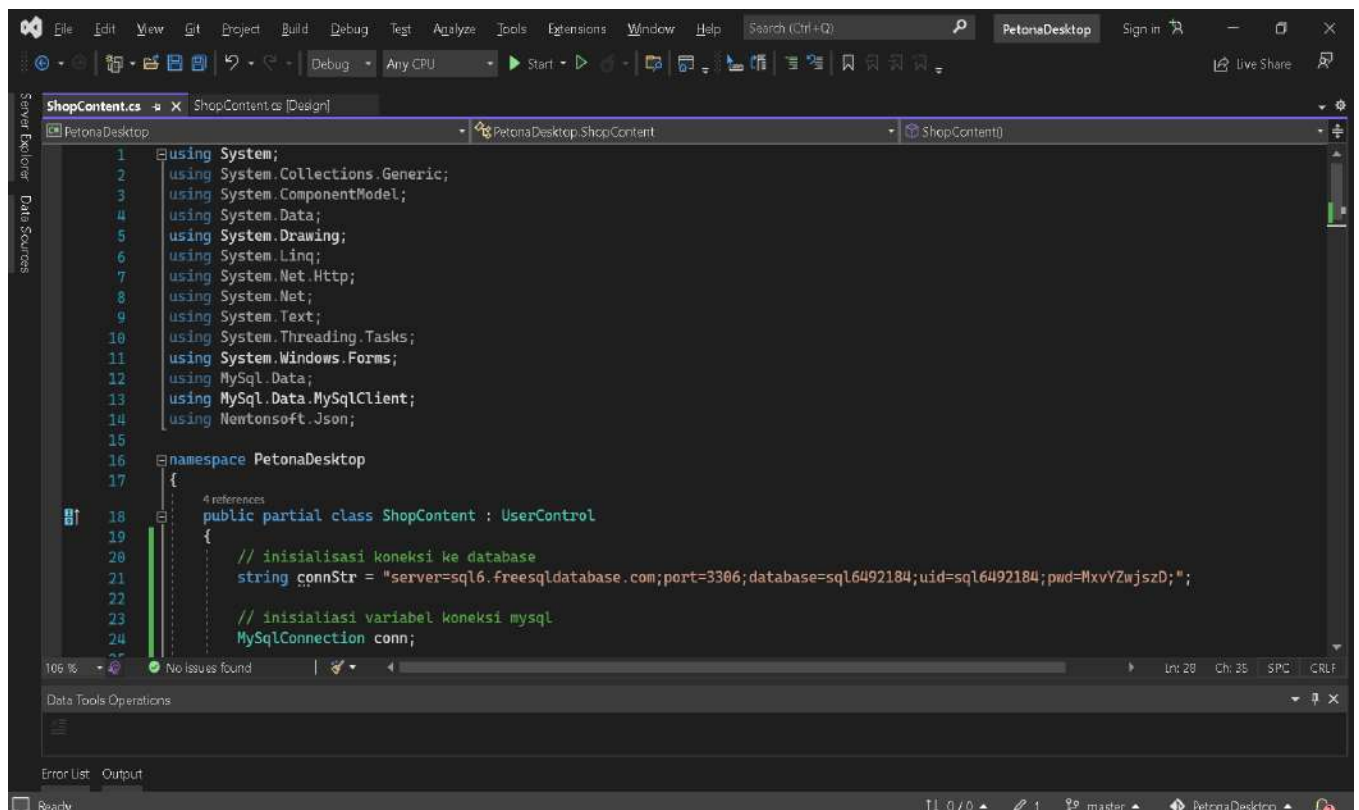


```
30
31 // menempatkan konten ke posisi paling depan
32 private void ShopBtn_Click(object sender, EventArgs e)
33 {
34     ShopContent.BringToFront();
35 }
36
37 // menempatkan konten ke posisi paling depan
38 private void ContactBtn_Click(object sender, EventArgs e)
39 {
40     ContactContent.BringToFront();
41 }
42
43 // menempatkan konten ke posisi paling depan
44 private void HomeBtn_Click(object sender, EventArgs e)
45 {
46     HomeContent.BringToFront();
47 }
48
49
50
51
```

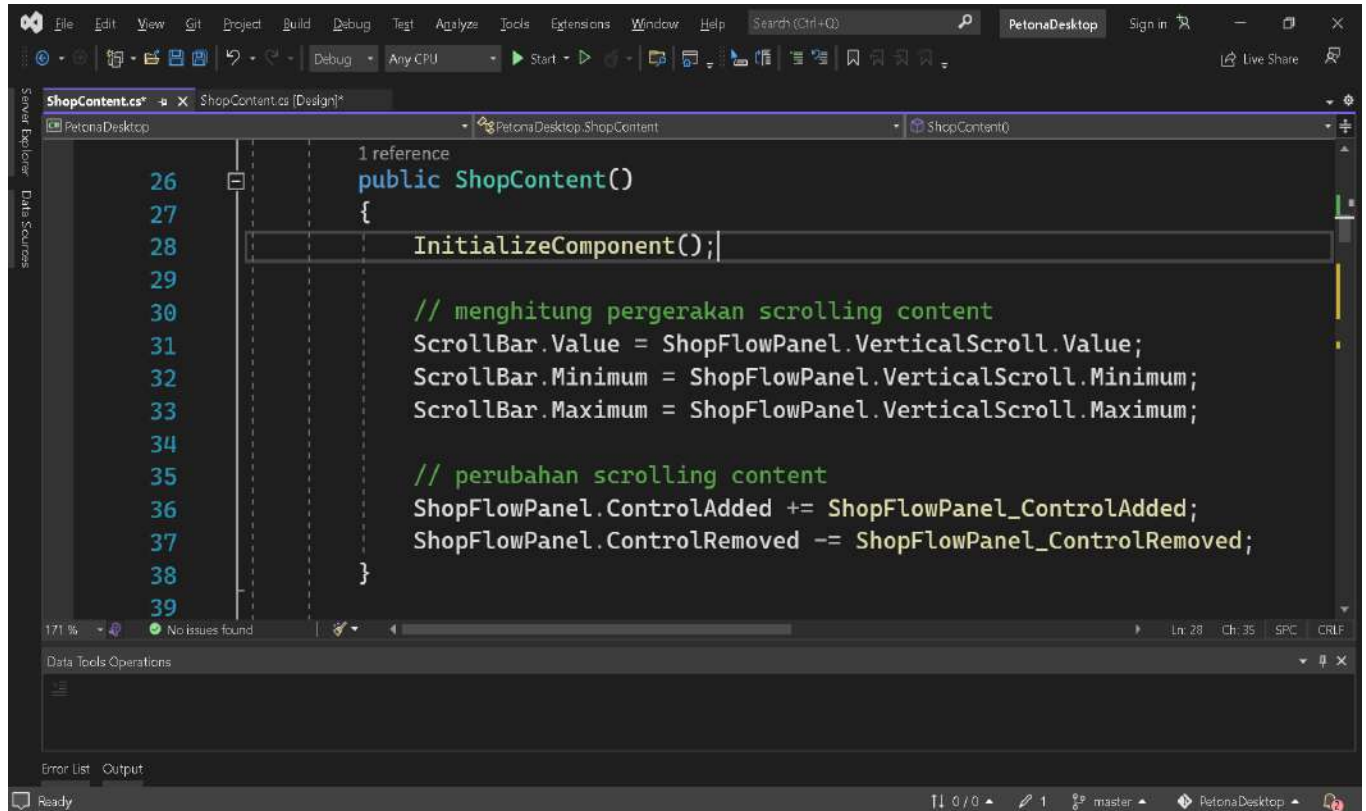
Design halaman Shop dengan membuat User Control(Windows Form) baru yang berisikan 1 Panel dengan nama ShopFlowPanel sebagai tempat dimana produk akan ditampilkan dan 1 Vertical Scroll Bar untuk scrolling produk ketika produk yang dimiliki banyak



Inisialisasi koneksi dan variabel koneksi ke database mysql



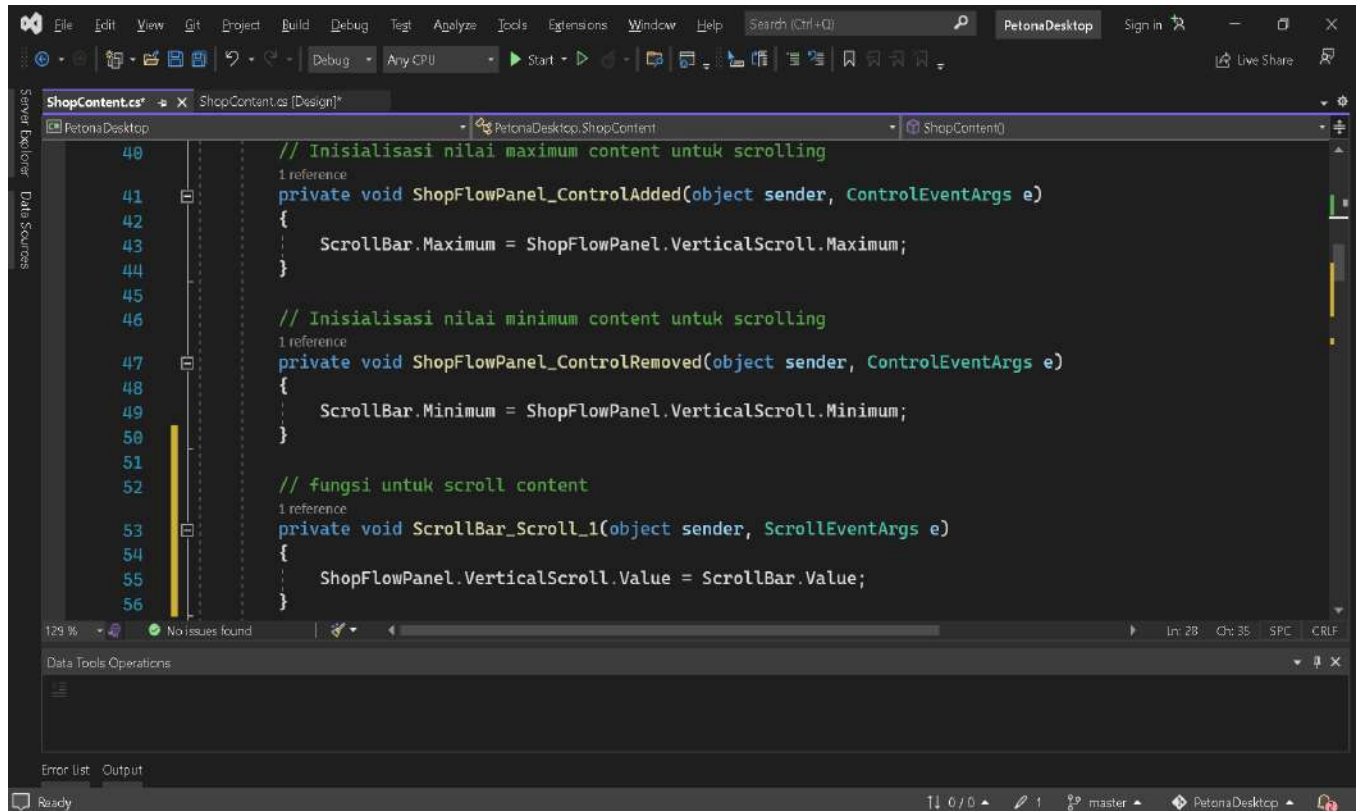
Membuat fungsi untuk scrolling halaman produk dimana value dari scrollbar akan berubah sesuai dengan penambahan control baru pada ShopFlowPanel



```
1 reference
public ShopContent()
{
    InitializeComponent();

    // menghitung pergerakan scrolling content
    ScrollBar.Value = ShopFlowPanel.VerticalScroll.Value;
    ScrollBar.Minimum = ShopFlowPanel.VerticalScroll.Minimum;
    ScrollBar.Maximum = ShopFlowPanel.VerticalScroll.Maximum;

    // perubahan scrolling content
    ShopFlowPanel.ControlAdded += ShopFlowPanel_ControlAdded;
    ShopFlowPanel.ControlRemoved -= ShopFlowPanel_ControlRemoved;
}
```

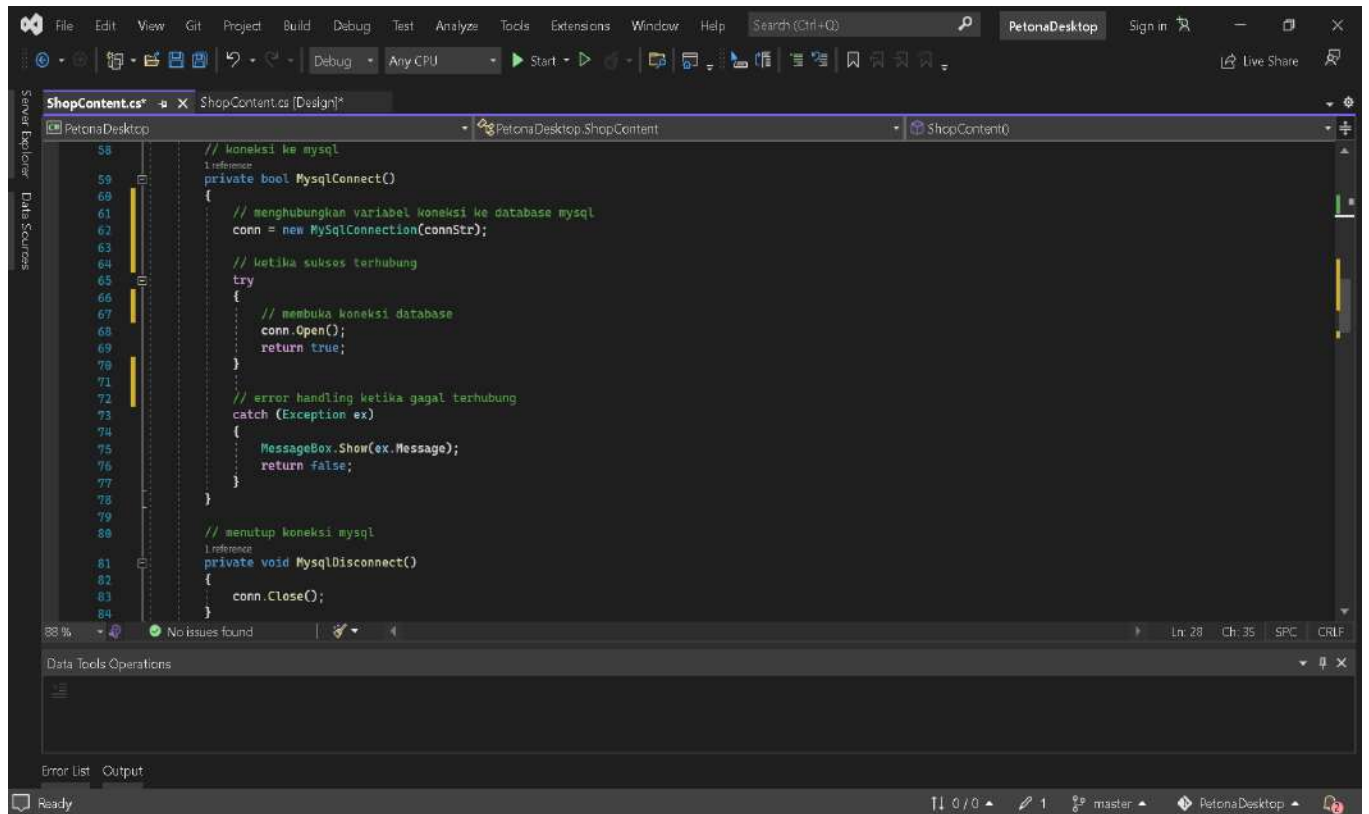


```
// Inisialisasi nilai maximum content untuk scrolling
1 reference
private void ShopFlowPanel_ControlAdded(object sender, ControlEventArgs e)
{
    ScrollBar.Maximum = ShopFlowPanel.VerticalScroll.Maximum;
}

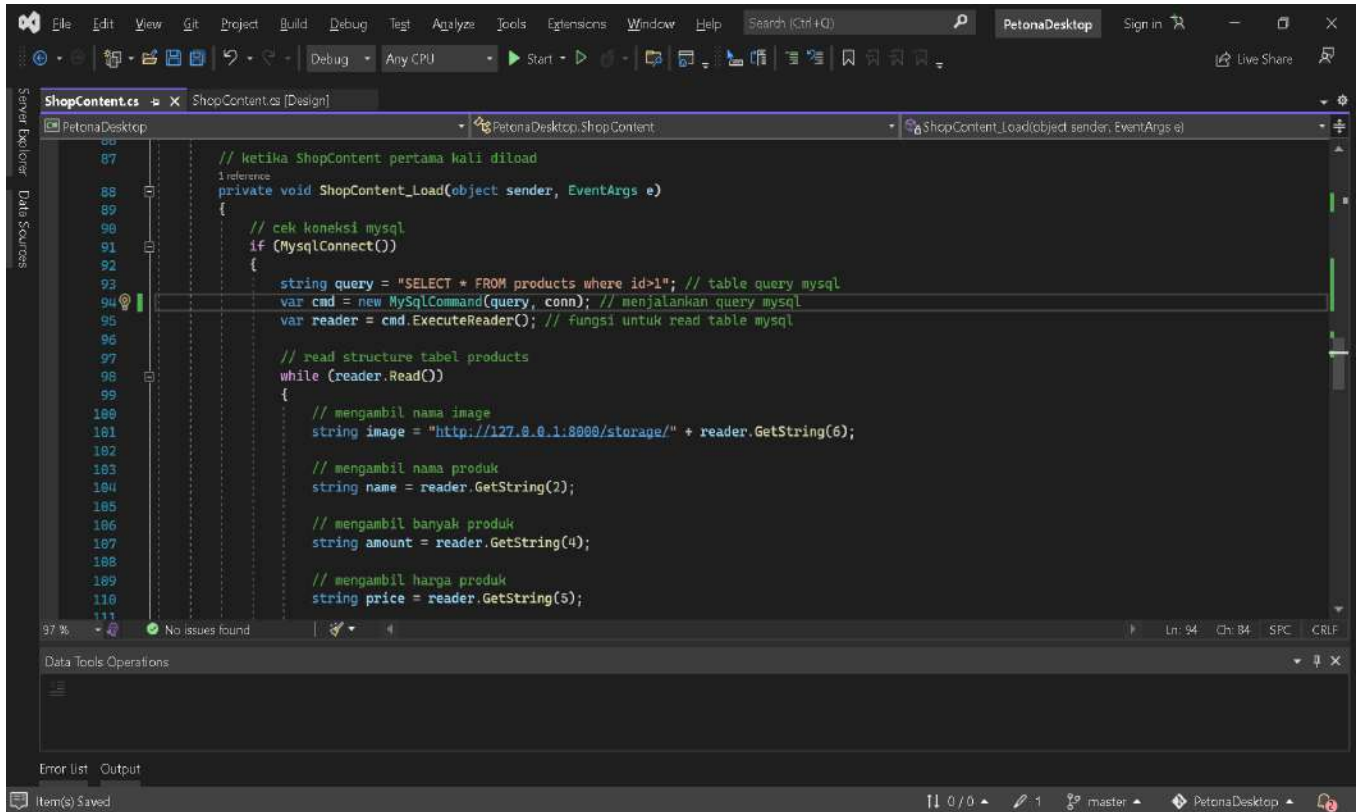
// Inisialisasi nilai minimum content untuk scrolling
1 reference
private void ShopFlowPanel_ControlRemoved(object sender, ControlEventArgs e)
{
    ScrollBar.Minimum = ShopFlowPanel.VerticalScroll.Minimum;
}

// fungsi untuk scroll content
1 reference
private void ScrollBar_Scroll_1(object sender, ScrollEventArgs e)
{
    ShopFlowPanel.VerticalScroll.Value = ScrollBar.Value;
}
```

Menghubungkan, membuka dan menutup koneksi ke database MySQL

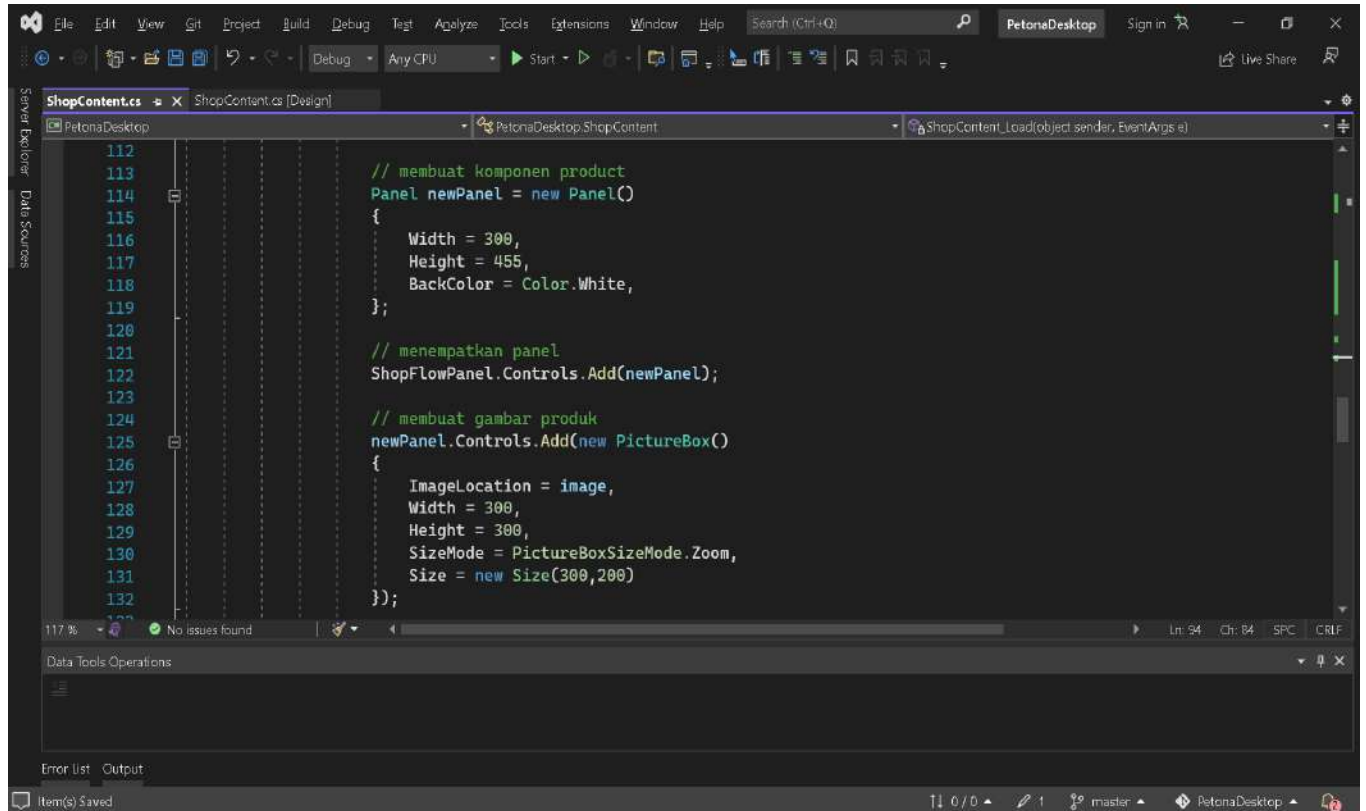


Ketika halaman Shop pertama kali di load dan koneksi ke database sukses maka variabel reader akan membaca structure tabel pada database untuk diambil data nama image, nama produk, banyak produk, dan harga produk. Lalu hasil reader tadi akan disimpan ke dalam masing-masing variable . Untuk nama image diambil dari api yang sudah diaktifkan dengan url <http://127.0.0.1:8000/storage/> ditambah dengan nama image yang ada di dalam database.



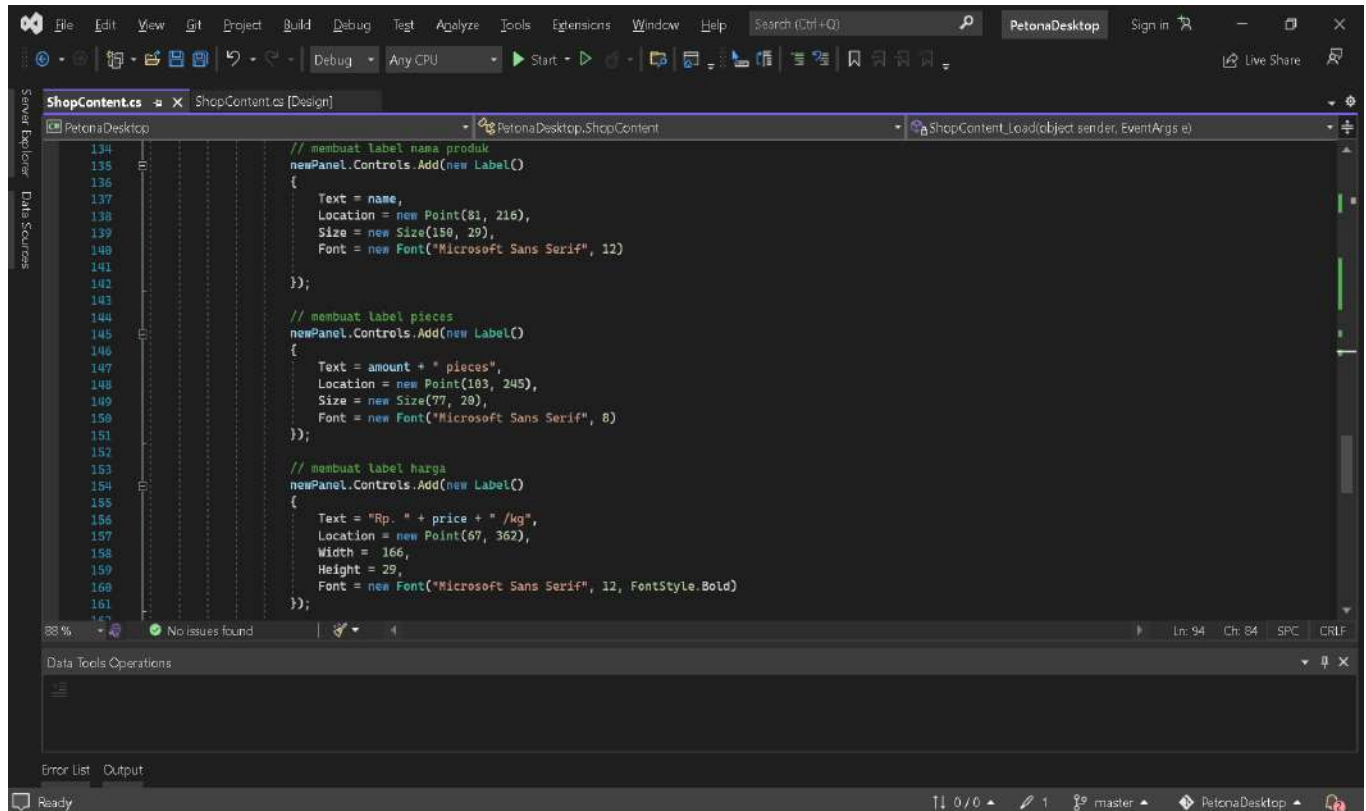
```
87 // ketika ShopContent pertama kali diload
88 1 reference
89 private void ShopContent_Load(object sender, EventArgs e)
90 {
91     // cek koneksi mysql.
92     if (MysqlConnect())
93     {
94         string query = "SELECT * FROM products where id>1"; // table query mysql
95         var cmd = new MySqlCommand(query, conn); // menjalankan query mysql
96         var reader = cmd.ExecuteReader(); // fungsi untuk read table mysql
97
98         // read structure tabel products
99         while (reader.Read())
100         {
101             // mengambil nama image
102             string image = "http://127.0.0.1:8000/storage/" + reader.GetString(6);
103
104             // mengambil nama produk
105             string name = reader.GetString(2);
106
107             // mengambil banyak produk
108             string amount = reader.GetString(4);
109
110             // mengambil harga produk
111             string price = reader.GetString(5);
112         }
113     }
114 }
```


Membuat komponen produk dengan panel yang berisikan 1 picturebox untuk gambar produk dan 3 label untuk nama, banyak produk, dan harga produk



The screenshot shows the Visual Studio IDE with the PetonaDesktop project open. The ShopContent.cs file is in Design view. The code defines a new Panel and adds a PictureBox to it. The panel has a width of 300, height of 455, and white background. The PictureBox is located at (81, 216) with a width of 300, height of 300, and is set to Zoom mode.

```
112
113
114 // membuat komponen produk
115 Panel newPanel = new Panel()
116 {
117     Width = 300,
118     Height = 455,
119     BackColor = Color.White,
120 };
121
122 // menempatkan panel
123 ShopFlowPanel.Controls.Add(newPanel);
124
125 // membuat gambar produk
126 newPanel.Controls.Add(new PictureBox()
127 {
128     ImageLocation = image,
129     Width = 300,
130     Height = 300,
131     SizeMode = PictureBoxSizeMode.Zoom,
132     Size = new Size(300,200)
133 });
```



The screenshot shows the Visual Studio IDE with the PetonaDesktop project open. The ShopContent.cs file is in Design view. The code adds three labels to the newPanel. The first label is for the product name, the second for the amount, and the third for the price. Each label has specific text, location, size, and font properties.

```
134 // membuat label nama produk
135 newPanel.Controls.Add(new Label()
136 {
137     Text = name,
138     Location = new Point(81, 216),
139     Size = new Size(150, 29),
140     Font = new Font("Microsoft Sans Serif", 12)
141 });
142
143 // membuat label pieces
144 newPanel.Controls.Add(new Label()
145 {
146     Text = amount + " pieces",
147     Location = new Point(103, 245),
148     Size = new Size(77, 20),
149     Font = new Font("Microsoft Sans Serif", 8)
150 });
151
152 // membuat label harga
153 newPanel.Controls.Add(new Label()
154 {
155     Text = "Rp. " + price + " /kg",
156     Location = new Point(67, 362),
157     Width = 166,
158     Height = 29,
159     Font = new Font("Microsoft Sans Serif", 12, FontStyle.Bold)
160 });
161
```


Membuat tombol didalam panel untuk produk dimana tombol ini akan digunakan untuk memasukkan produk ke dalam troli belanja lalu menutup koneksi ke database dengan fungsi MysqlDisconnect()

The screenshot displays the Visual Studio Code editor with a C# file named `ShopContent.cs` open. The code is as follows:

```

162
163
164 // membuat tombol
165 newPanel.Controls.Add(new Button()
166 {
167     Text = "Masukkan Troli",
168     Width = 202,
169     Height = 31,
170     Location = new Point(46, 399),
171     Font = new Font("Microsoft Sans Serif", 8, FontStyle.Bold),
172     ForeColor = Color.White,
173     FlatStyle = FlatStyle.Flat,
174     BackColor = Color.Green,
175 });
176
177
178 // keluar koneksi mysql
179 MysqlDisconnect();
180
181
182

```

The interface includes a menu bar at the top with options like File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, and Help. A toolbar below the menu bar contains icons for various actions. The sidebar on the left shows the Explorer, Search, and Run and Debug views. The bottom status bar indicates the current file is `ShopContent.cs` and shows the status 'Ready'.

Hasil dari Halaman Shop

