

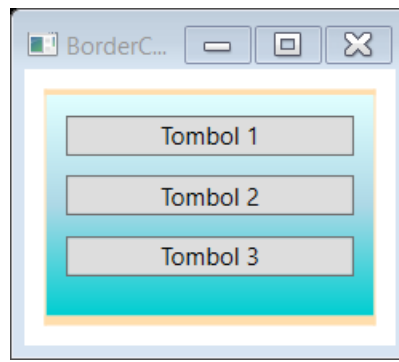
Pertemuan 11

Border Control

Pada modul ke 12 ini kita akan mempelajari beberapa fungsi kontrol yang kerap digunakan pada aplikasi desktop. Salah satu contoh nya adalah penggunaan border untuk menghiasi tampilan aplikasi. Seperti fungsi border pada umumnya, control ini digunakan untuk membentuk batas serta latar dari objek lain sehingga tampak jelas batasnya. Kontrol ini diperlukan sebab control Panel yang dipelajari diatas belum mendukung pembentukan garis tepi untuk objeknya. Silahkan bentuk sebuah proyek WPF baru, beri nama "BelajarControl1" → Tambahkan sebuah file Window WPF bernama "BorderControl". Kemudian tambahkan kode XAML berikut ini.

```
Title="BorderControl" Height="175" Width="200">
<Grid Margin="10">
  <Border BorderBrush="NavajoWhite" BorderThickness="1,3,1,5">
    <Border.Background>
      <LinearGradientBrush StartPoint="0.5,0" EndPoint="0.5,1">
        <GradientStop Color="LightCyan" Offset="0.0" />
        <GradientStop Color="LightBlue" Offset="0.5" />
        <GradientStop Color="DarkTurquoise" Offset="1.0" />
      </LinearGradientBrush>
    </Border.Background>
    <StackPanel Margin="10">
      <Button>Tombol 1</Button>
      <Button Margin="0,10">Tombol 2</Button>
      <Button>Tombol 3</Button>
    </StackPanel>
  </Border>
</Grid>
```

Setelah selesai mengetikkan kode di atas, kemudian jalankan aplikasi untuk melihat hasilnya. Silahkan ubah warna latar sesuai kreativitas anda.

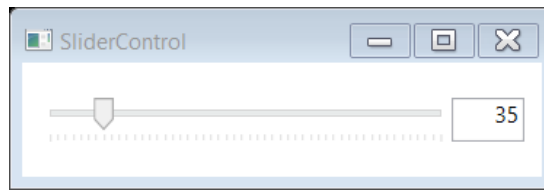


Slider Control

Contoh kedua adalah penggunaan control Slider yang memungkinkan kita untuk mengatur nilai properti atau nilai lainnya dengan range tertentu dengan cara menggeser panelnya. Silahkan tambahkan Window baru, beri nama "SliderControl".

```
Title="SliderControl" Height="100" Width="300">
<DockPanel VerticalAlignment="Center" Margin="10">
  <TextBox Text="{Binding ElementName=slValue, Path=Value,
    UpdateSourceTrigger=PropertyChanged}" DockPanel.Dock="Right"
    TextAlignment="Right" Width="40" />
  <Slider Maximum="255" TickPlacement="BottomRight" TickFrequency="5"
    IsSnapToTickEnabled="True" Name="slValue" />
</DockPanel>
```

Jalankan aplikasi tersebut untuk melihat hasilnya.



Untuk contoh kedua kita akan membentuk sebuah file Window baru bernama "SliderControl2". Pada contoh ini kita akan menggunakan nilai dari Slider yang kita geser pada aplikasi untuk menentukan warna latar dari Window yang sedang aktif.

```
Title="SliderControl2" Height="200" Width="300">
<StackPanel Margin="10" VerticalAlignment="Center">
  <DockPanel VerticalAlignment="Center" Margin="10">
    <Label DockPanel.Dock="Left" FontWeight="Bold">R:</Label>
    <TextBox Text="{Binding ElementName=slColorR, Path=Value, UpdateSourceTrigger=PropertyChanged}"
      DockPanel.Dock="Right" TextAlignment="Right" Width="40" />
    <Slider Maximum="255" TickPlacement="BottomRight" TickFrequency="5" IsSnapToTickEnabled="True"
      Name="slColorR" ValueChanged="ColorSlider_ValueChanged" />
  </DockPanel>

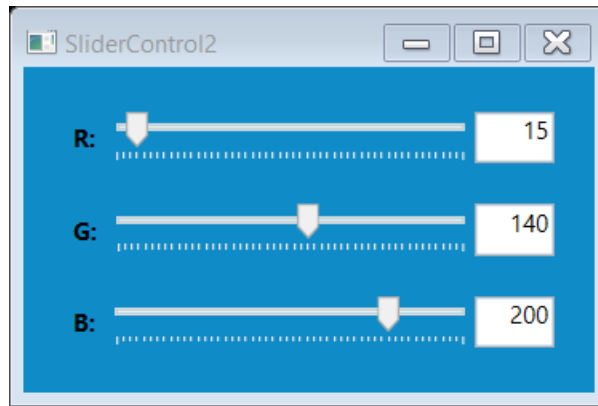
  <DockPanel VerticalAlignment="Center" Margin="10">
    <Label DockPanel.Dock="Left" FontWeight="Bold">G:</Label>
    <TextBox Text="{Binding ElementName=slColorG, Path=Value, UpdateSourceTrigger=PropertyChanged}"
      DockPanel.Dock="Right" TextAlignment="Right" Width="40" />
    <Slider Maximum="255" TickPlacement="BottomRight" TickFrequency="5" IsSnapToTickEnabled="True"
      Name="slColorG" ValueChanged="ColorSlider_ValueChanged" />
  </DockPanel>

  <DockPanel VerticalAlignment="Center" Margin="10">
    <Label DockPanel.Dock="Left" FontWeight="Bold">B:</Label>
    <TextBox Text="{Binding ElementName=slColorB, Path=Value, UpdateSourceTrigger=PropertyChanged}"
      DockPanel.Dock="Right" TextAlignment="Right" Width="40" />
    <Slider Maximum="255" TickPlacement="BottomRight" TickFrequency="5" IsSnapToTickEnabled="True"
      Name="slColorB" ValueChanged="ColorSlider_ValueChanged" />
  </DockPanel>
</StackPanel>
```

Tambahkan kode C# sederhana ini pada file SliderControl2.xaml.cs untuk mendapatkan nilai yang telah kita set pada UI.

```
private void ColorSlider_ValueChanged(object sender, RoutedPropertyChangedEventArgs<double> e)
{
    Color color = Color.FromRgb((byte)s1ColorR.Value, (byte)s1ColorG.Value, (byte)s1ColorB.Value);
    this.Background = new SolidColorBrush(color);
}
```

Kemudian jalankan aplikasi, maka latar belakang dari tampilan akan berubah warna sesuai dengan kombinasi nilai RGB yang anda tentukan.



Progress Bar Control

Jika pada Slider kita mengatur nilai sesuai kebutuhan, progress bar control digunakan untuk menampilkan progress yang sedang berlangsung dari aplikasi yang dijalankan. Dengan memberikan nilai minimum dan maksimum yang kian bertambah, hal ini akan memberikan gambaran seberapa jauh perkembangan dari proses yang sedang berlangsung. Tambahkan sebuah file WPF baru, beri nama „ProgressBar“, lalu tambahkan file XAML berikut ini.

```
Title="ProgressBar" Height="100" Width="300"
ContentRendered="Window_ContentRendered">
<Grid Margin="20">
    <ProgressBar Minimum="0" Maximum="100" Name="pbStatus" />
    <TextBlock Text="{Binding ElementName=pbStatus, Path=Value, StringFormat={}{0:0}%"
        HorizontalAlignment="Center" VerticalAlignment="Center" />
</Grid>
```

Langkah selanjutnya adalah membuka file .cs untuk memasukkan logic dari program yang akan dibentuk. Silahkan ketikkan kode program berikut.

```
using System;
using System.ComponentModel;
using System.Threading;
using System.Windows;

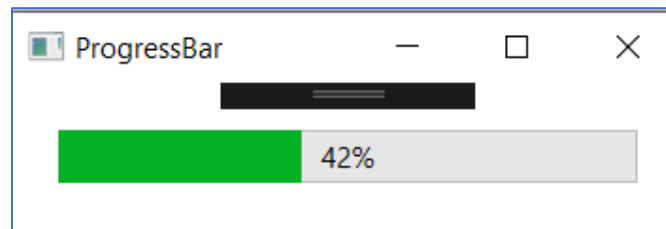
namespace BelajarControl1
{
    /// Interaction logic for ProgressBar.xaml
    /// </summary>
    public partial class ProgressBar : Window
    {
        public ProgressBar()
        {
            InitializeComponent();
        }

        private void Window_ContentRendered(object sender, EventArgs e)
        {
            BackgroundWorker worker = new BackgroundWorker();
            worker.WorkerReportsProgress = true;
            worker.DoWork += worker_DoWork;
            worker.ProgressChanged += worker_ProgressChanged;
            worker.RunWorkerAsync();
        }

        void worker_DoWork(object sender, DoWorkEventArgs e)
        {
            for (int i = 0; i < 100; i++)
            {
                (sender as BackgroundWorker).ReportProgress(i);
                Thread.Sleep(100);
            }
        }

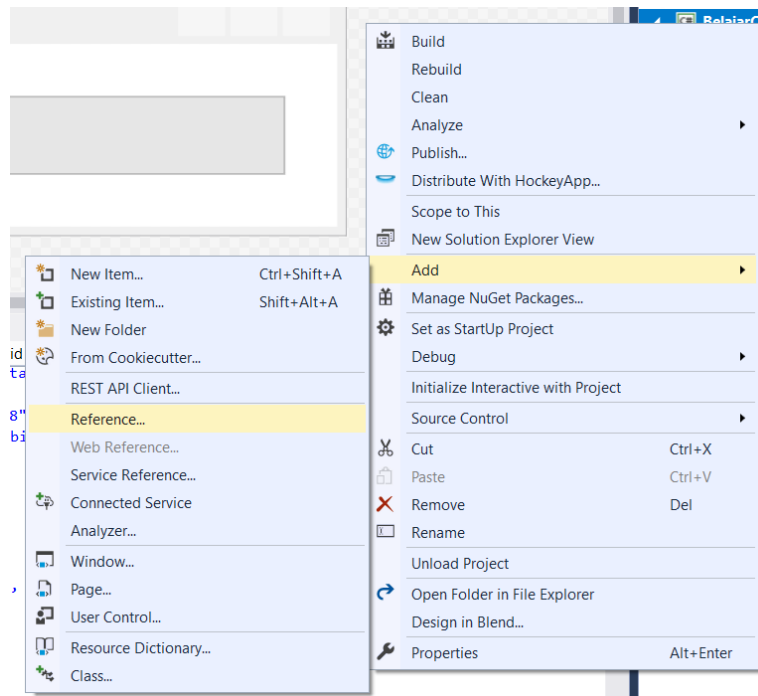
        void worker_ProgressChanged(object sender, ProgressChangedEventArgs e)
        {
            pbStatus.Value = e.ProgressPercentage;
        }
    }
}
```

Kemudian jalankan aplikasi untuk melihat hasilnya. Jika benar, maka aplikasi akan berjalan seperti pada contoh di bawah ini.

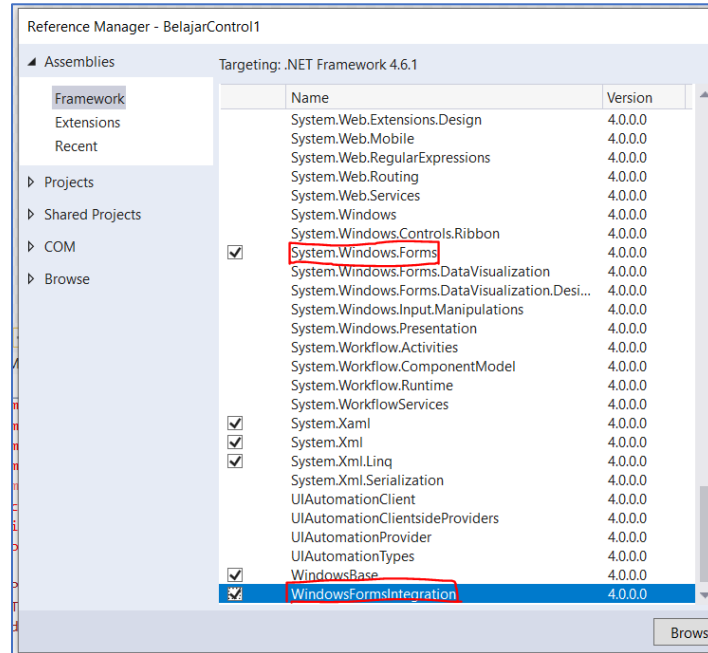


Web Browser Control

Kontrol jenis ini digunakan untuk membentuk sebuah web browser kecil dengan menggunakan WPF. Akan tetapi, fungsi yang disediakan oleh WPF dalam hal ini lebih terbatas dan sulit jika dibandingkan dengan WinForms. Untuk menggunakan fungsi ini, kita harus menambahkan referensi ke dalam aplikasi yang dibentuk yaitu WindowsFormsIntegration dan System.Windows.Forms. Buat sebuah file Window baru, beri nama „WebBrowserControl“. Kemudian klik kanan pada nama proyek → Add → Reference.



Kemudian akan muncul sebuah kotak dialog, expand bagian Assemblies → pilih framework → centang bagian System.Windows.Forms dan WindowsFormsIntegration → Ok.



Setelah menambahkan dua buah referensi tersebut, silahkan buka kembali file WebBrowserControl.xaml dan ketikkan kode berikut ini.

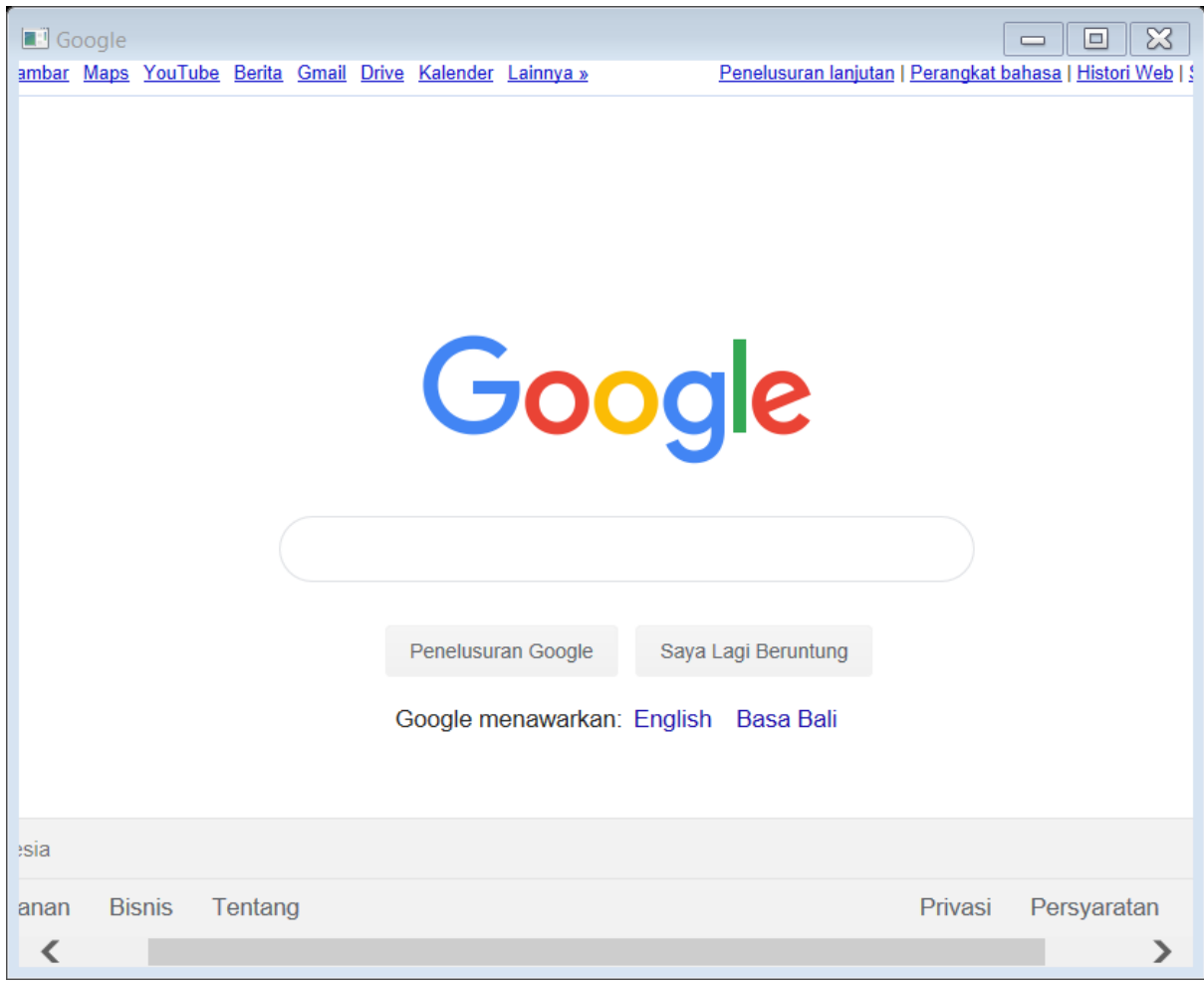
```
xmlns:wf="clr-namespace:System.Windows.Forms;assembly=System.Windows.Forms"
mc:Ignorable="d"
Title="WebBrowserControl" Height="350" Width="450">
<Grid>
  <WindowsFormsHost Name="wfhSample">
    <WindowsFormsHost.Child>
      <wf:WebBrowser DocumentTitleChanged="wbWinForms_DocumentTitleChanged" />
    </WindowsFormsHost.Child>
  </WindowsFormsHost>
</Grid>
```

Lalu buka file .cs untuk menambahkan alamat web yang ingin ditampilkan pada aplikasi. Pada contoh ini kita menggunakan google.com sebagai halaman browser yang ditampilkan.

```
public WebBrowserControl()
{
    InitializeComponent();
    (wfhSample.Child as System.Windows.Forms.WebBrowser).Navigate("http://www.google.com");
}

private void wbWinForms_DocumentTitleChanged(object sender, EventArgs e)
{
    this.Title = (sender as System.Windows.Forms.WebBrowser).DocumentTitle;
}
```

Kemudian jalankan aplikasi dan lihat hasilnya yang seperti pada contoh berikut ini.



Analisa :

1. Silahkan Analisa baris kode program bagian logic (C#) yang terdapat pada setiap contoh program di atas. Beri penjelasan fungsi dari baris kode program/method yang digunakan pada aplikasi tersebut.