

Apa itu WPF?

WPF adalah singkatan singkat dari Windows Presentation Foundation. Ini adalah kerangka kerja ideal untuk membangun aplikasi Windows. Gagasan di balik pengenalan kerangka WPF adalah bahwa pengguna dapat mengembangkan antarmuka pengguna yang kaya fitur yang terdiri dari 3D, animasi, warna yang kaya dengan kompleksitas kode minimum.

WPF dapat dianggap sebagai evolusi dari WinForms. Ini menghasilkan sistem tampilan grafis yang mudah dioperasikan untuk windows. WPF menawarkan pemisahan yang mudah antara UI dan logika bisnis.

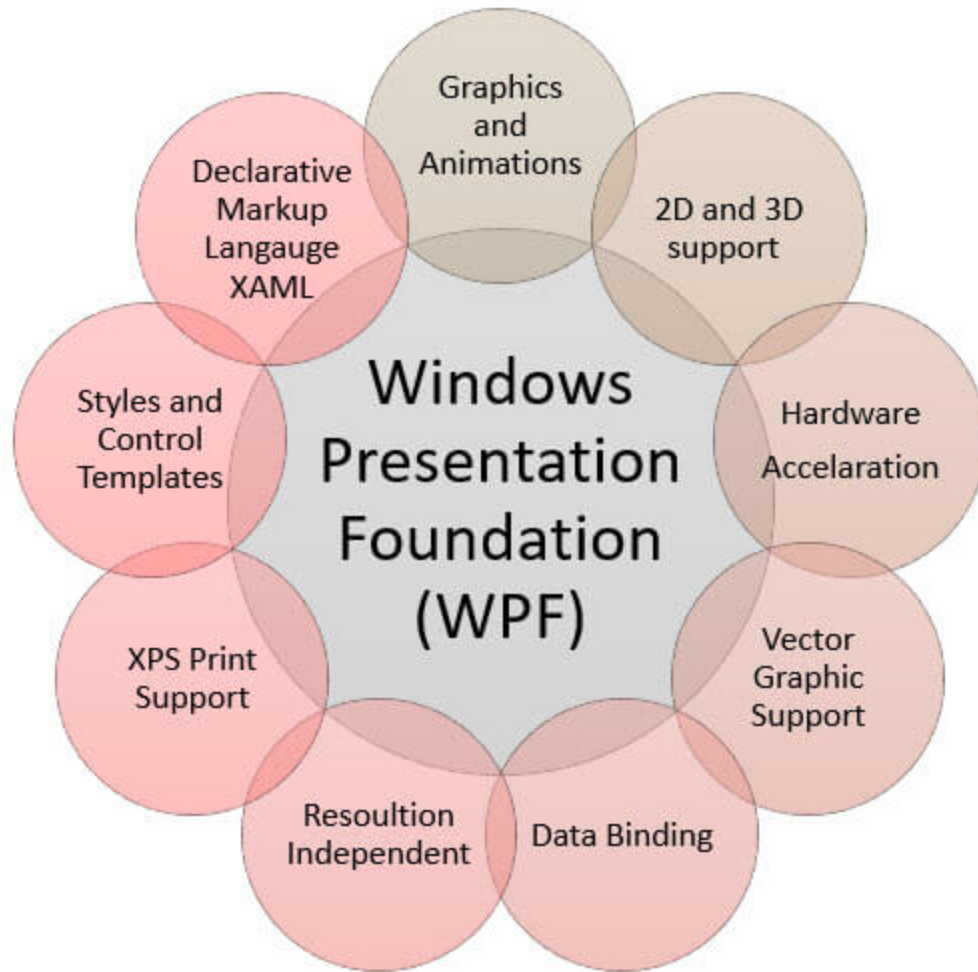
WPF adalah mesin rendering berbasis vektor yang menggunakan akselerasi perangkat keras kartu grafis modern yang membuat UI lebih cepat dan sangat terukur. Aplikasi WPF dapat digunakan dalam sistem Anda sebagai program desktop mandiri atau di-host sebagai objek tertanam di situs web.

Mengapa WPF?

Ada banyak alasan untuk menggunakan platform WPF:

- Platform ideal untuk digunakan jika Anda harus berurusan dengan berbagai jenis media
- WPF memungkinkan Anda untuk membuat antarmuka pengguna dengan tampilan yang bisa diubah (***skinned UI***). WPF juga digunakan ketika perlu memuat sebagian UI atau ingin mengikat data XML
- WPF memungkinkan untuk mengambil manfaat dari pustaka/library .NET karena dibangun di atas teknologi .NET
- Resolusi independen, sehingga aplikasi yang dikembangkan menggunakan WPF dapat dijalankan pada perangkat seluler atau monitor pada PC.
- WPF dibangun di atas DirectX bukan WinAPI, oleh karena itu, WPF menawarkan kekuatan rendering grafis yang lebih besar
- Mendukung model "***flow dokumen***" yang memungkinkan pembuatan "***desktop publishing***" dengan tata letak
- UI dan Code program atau designer dan pengembang untuk mengerjakan dua model terpisah
- Dalam kerangka kerja WPF Anda dapat mendesain tool(komponen control) secara grafis pada dokumen XML
- Memungkinkan Anda menggunakan akselerasi perangkat keras untuk menggambar GUI, untuk kinerja yang lebih baik

Fitur WPF



Jenis dan Infrastruktur Inti WPF

Kelas-kelas dalam WPF dibagi menjadi empat jenis:

- *UIElement*
- *FrameworkElement*
- *ContentElement*
- *FrameworkContentElement*

Kelas-kelas ini, dikenal sebagai kelas elemen dasar. Ini memberikan dasar untuk model penulisan antarmuka pengguna.

Antarmuka pengguna WPF terdiri dari elemen yang dirangkai dalam hierarki pohon. Ini disebut elemen tree. Tree elemen adalah cara intuitif untuk tata letak antarmuka pengguna. Ini adalah struktur yang memungkinkan Anda untuk mendapatkan fitur layanan UI yang kuat.

XAML

Extensible Application Markup Language yang disebut "XAML", atau "zammel" adalah cara deklaratif untuk mendefinisikan antarmuka pengguna.

Berikut adalah definisi XAML dari tombol sederhana:

```
<Button
  FontSize = "16"
  HorizontalAlignment = "Center"
  VerticalAlignment = " Center "> Hello XAML</Button>
```

Keuntungan menggunakan bahasa XAML:

- XAML memisahkan UI dari logika program
- XAML adalah metode termudah untuk mewakili antarmuka pengguna
- XAML bekerja secara efektif dengan tool (komponen control)

Komponen Kontrol

Kerangka WPF mendukung banyak kontrol yang berguna seperti:

- Komponen editing kontrol seperti TextBox, CheckBox, RadioButton
- List komponen seperti ListBox, ListView, TreeView
- Informasi pengguna seperti Label, ProgressBar, ToolTip
- Tindakan seperti Menu, Tombol, dan ToolBar
- Tampilan seperti Border, Image, dan Viewbox
- Kotak dialog umum seperti OpenFileDialog dan PrintDialog.
- Container seperti TabContro, ScrollBar, dan GroupBox
- Layout seperti DocPanel, StackPanel, dan Grid
- Navigasi seperti Frame dan Hyperlink

Tampilan kontrol dapat disesuaikan dengan Style dan template tanpa pemrograman. Anda juga dapat membuat kontrol khusus dengan menurunkan kelas baru dari kelas dasar yang sesuai.

Tampilan

WPF menyediakan fitur untuk menyesuaikan tampilan aplikasi Anda. Ini memungkinkan Anda untuk mengatur objek dan nilai untuk hal-hal seperti font, latar belakang, dll.

Fitur style memungkinkan Anda untuk melakukan standarisasi pada tampilan spesifik untuk seluruh produk. Ini memungkinkan Anda untuk mengganti tampilan default sambil mempertahankan perilaku default-nya.

Template data memungkinkan Anda untuk mengontrol visualisasi default dari data yang terikat. Dengan bantuan tema, Anda dapat dengan mudah memvisualisasikan style dari sistem operasi.

Tata Letak & Panel

Layout membantu Anda dalam menentukan posisi dan mengatur ukuran dengan benar. Ini adalah bagian dari proses menyusun presentasi untuk pengguna. Kerangka kerja WPF memudahkan proses tata letak dan menawarkan kemampuan beradaptasi yang lebih baik dari tampilan UI.

Infrastruktur tata letak yang ditawarkan oleh berbagai kelas adalah 1) StackPanel 2) DockPanel 3) WrapPanel 4) Grid dan 5) Canvas

Grafik

WPF menyediakan sistem grafis yang lebih baik

- WPF menggunakan unit independen dari perangkat keras, memungkinkan resolusi dan independensi perangkat. Oleh karena itu, setiap piksel, yang tidak tergantung pada perangkat, secara otomatis berskala dengan sistem pengaturan titik-per-inci
- WPF menggunakan double bukannya float dan mendukung jajaran warna yang lebih luas
- Mesin grafis WPF dirancang sedemikian rupa sehingga dapat memanfaatkan perangkat keras grafis saat tersedia

Dokumen dan Pencetakan

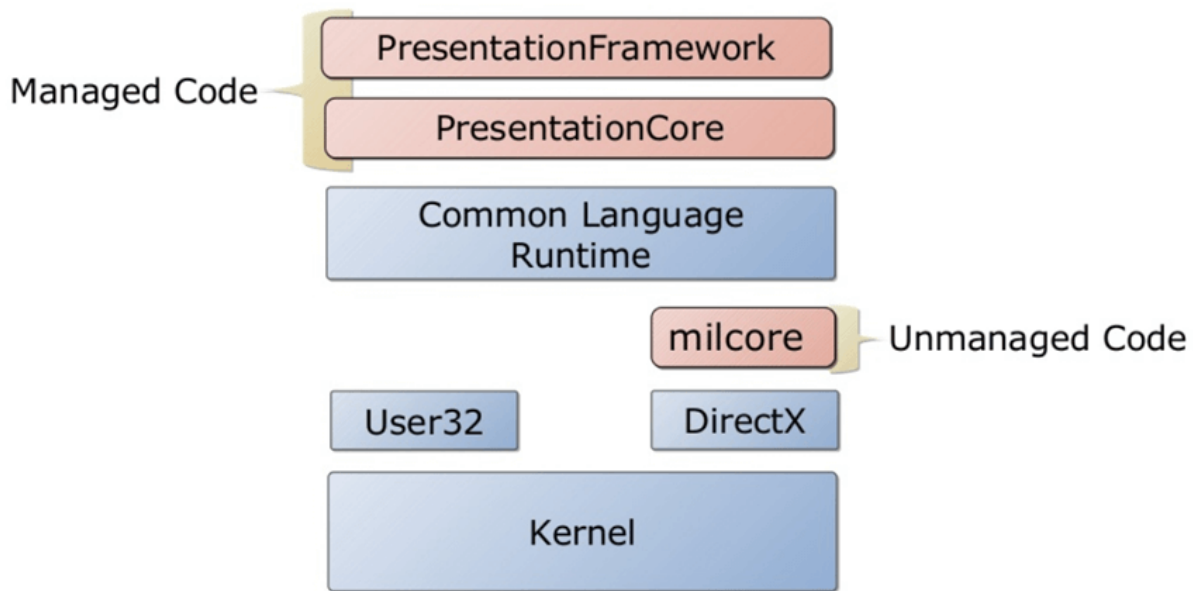
Kerangka WPF menawarkan tiga jenis dokumen:

- **Document Control - Document:** Jenis dokumen ini mendukung presentasi WYSIWYG. Fixed
- **Flow Dokumen:** Memungkinkan Anda untuk menyesuaikan dan mereflow konten berdasarkan variabel run-time seperti ukuran jendela dan resolusi perangkat.
- **Dokumen XPS:** Merupakan representasi makalah dari kertas elektronik dalam format berbasis XML. XPS adalah format dokumen sumber terbuka dan lintas platform.

WPF memungkinkan kontrol yang lebih baik atas sistem cetak. Ini termasuk pencetakan jarak jauh dan antrian. Selain itu, dokumen XPS dapat dicetak langsung tanpa mengubahnya menjadi format cetak.

Arsitektur WPF

WPF adalah bagian dari framework .NET. Ini mengandung kode yang dikelola dan tidak dikelola. Komponen penting arsitektur WPF dijelaskan pada gambar di bawah ini:



Komponen terpenting dari bagian arsitektur WPF adalah -

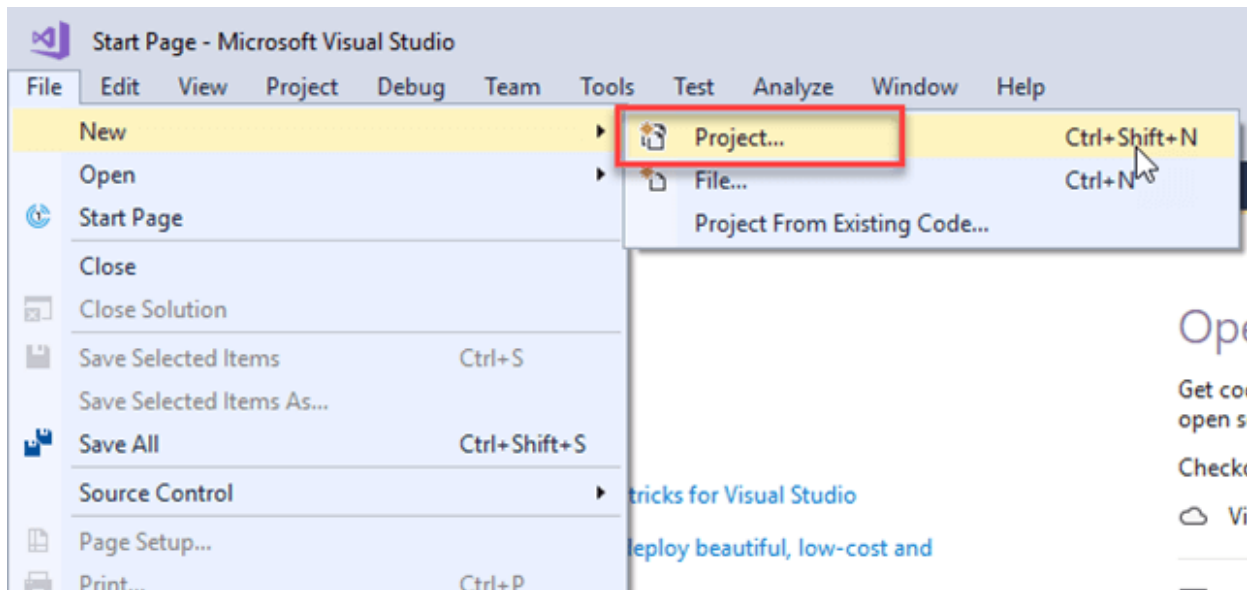
- **PresentationFramework** : Membantu Anda membuat elemen tingkat atas seperti Kontrol, style, tata letak, jendela, dll.
- **PresentationCore** : Ini memegang jenis dasar seperti UIElement, Visual dari semua Kontrol dan bentuk diturunkan dari PresentationFramework.dll.
- **CLR** : Itu membuat proses pengembangan produktif dengan menawarkan fitur-fitur seperti manajemen memori, penanganan kesalahan, dll.
- **Milcore**: Milcore adalah bagian dari kode tidak terkelola yang menawarkan integrasi dengan DirectX.
- **DirectX**: Ini adalah API tingkat rendah yang memungkinkan rendering gambar WPF. DirectX berinteraksi dengan driver dan menyajikan konten.
- **User32**: Ini adalah API inti yang banyak digunakan oleh banyak program. Ia mengelola memori dan proses. User32 membantu Anda memutuskan elemen mana yang akan ditempatkan di layar.
- **Kernel**: Input berasal sebagai sinyal pada driver perangkat mode kernel dan dialihkan ke proses yang benar dan dengan menghubungkan dengan kernel Windows dan User32.

Instalasi WPF

Membangun WPF Pertama Anda

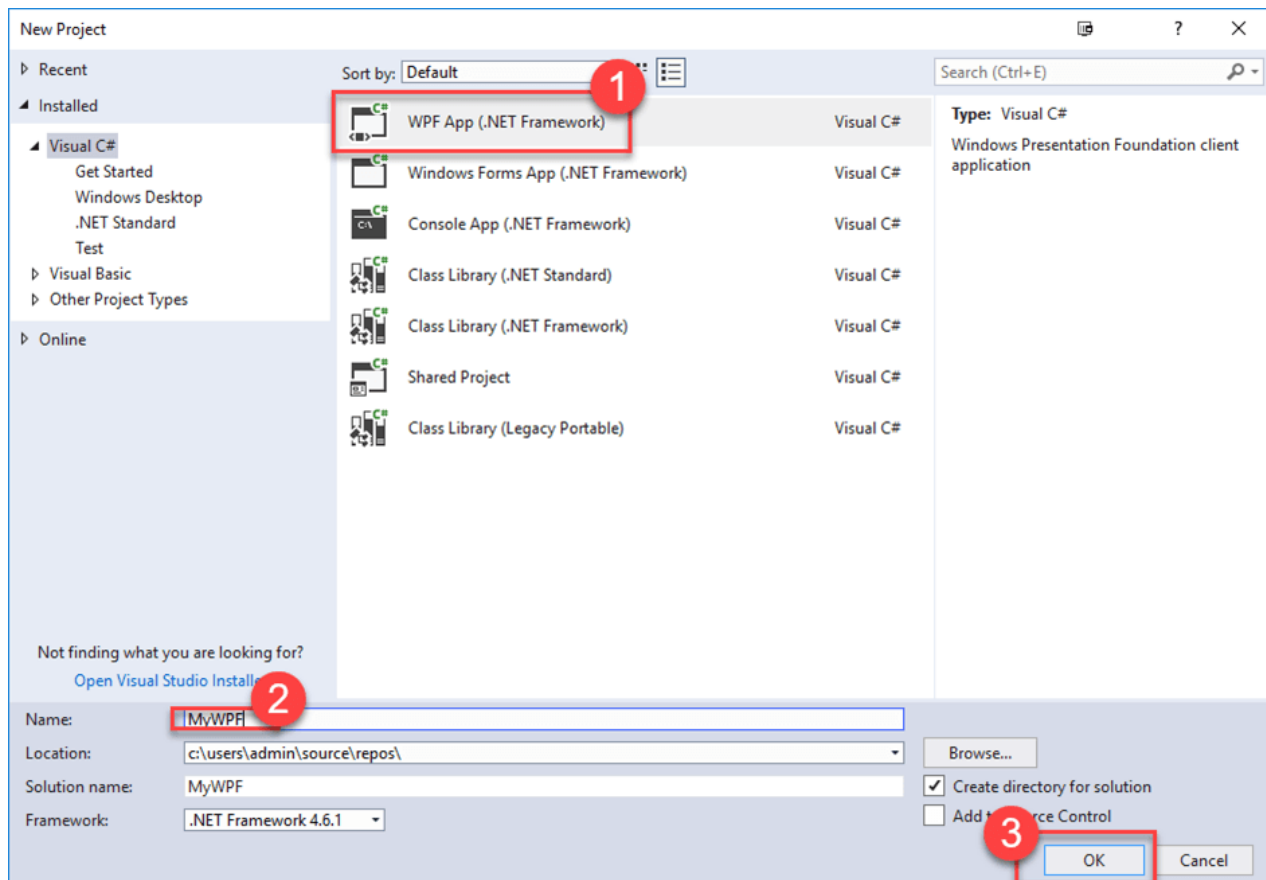
Pada bagian ini, kami akan mengembangkan aplikasi WPF dasar. Jadi, mari kita mulai implementasi sederhana dengan mengikuti langkah-langkah yang diberikan di bawah ini.

Langkah 1) Di Visual Studio Buka File> Project



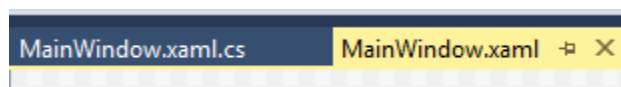
Langkah 2) Di jendela proyek baru

1. Pilih Aplikasi WPF
2. Masukkan Nama sebagai "MyWPF"
3. Klik OK



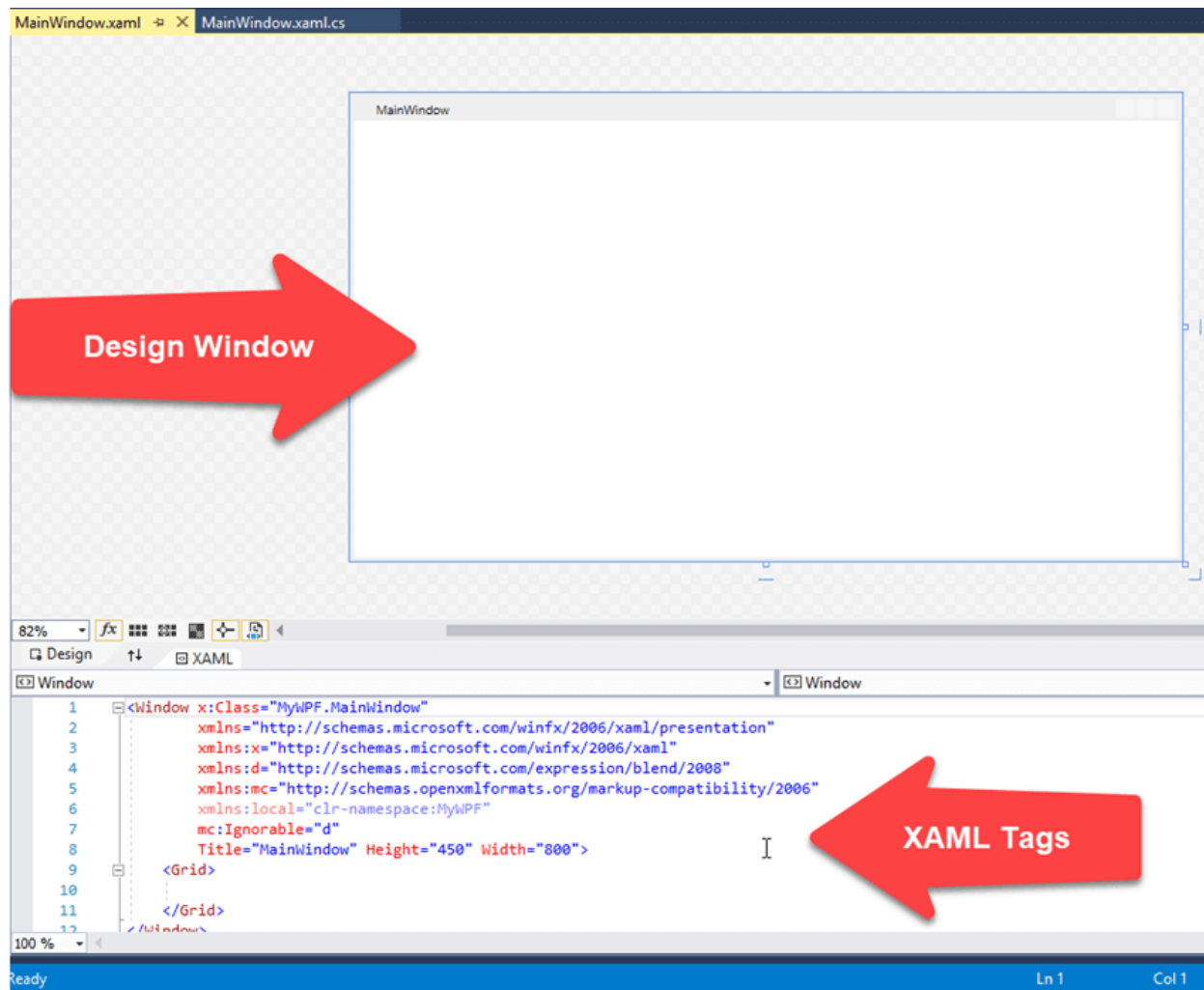
Langkah 3) Visual Studio secara default membuat dua file

- File XAML (MainWindow.xaml)
- File CS (MainWindow.xaml.cs)



MainWindow.xaml memiliki

1. Jendela Desain
2. File XAML

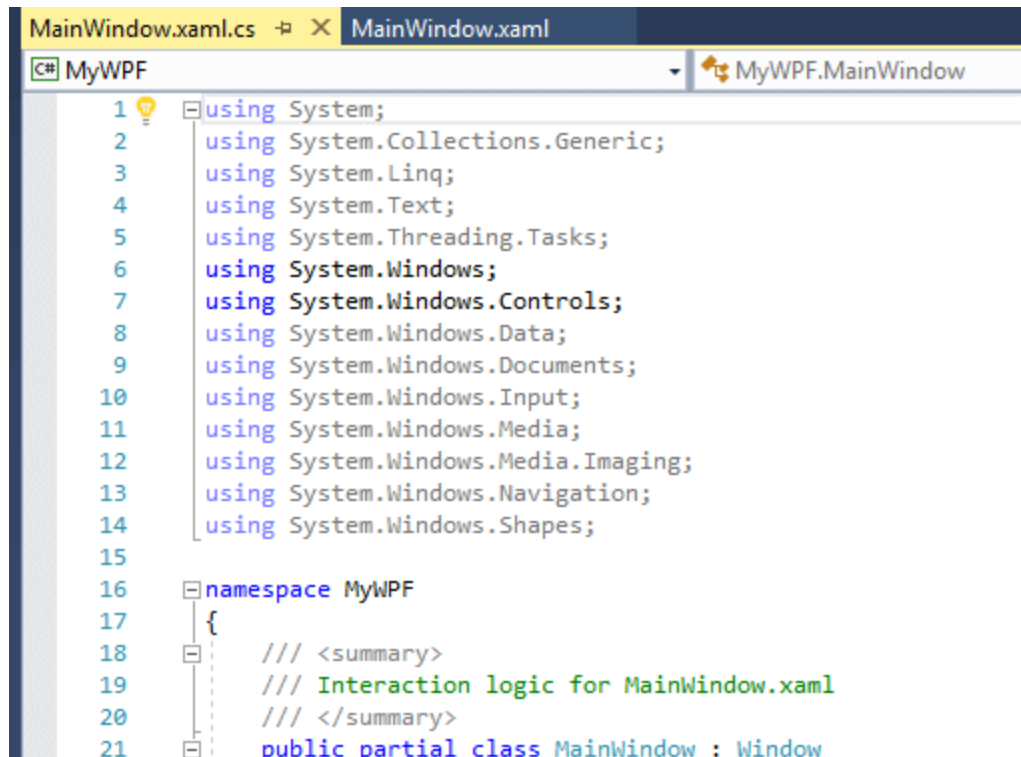


Di jendela XAML, tag berikut ditulis sebagai default



Grid adalah elemen pertama secara default.

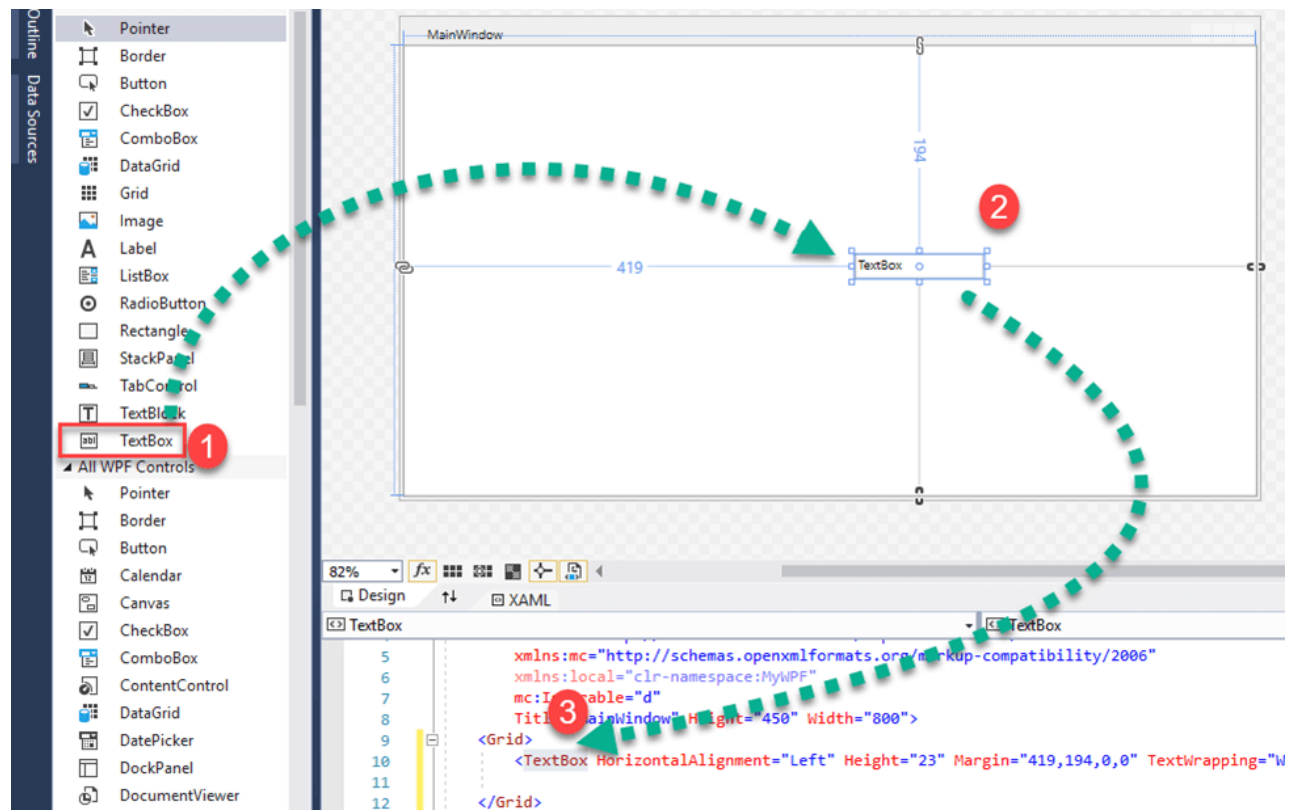
`MainWindow.xaml.cs` berisi kode yang sesuai di belakang file desain XAML



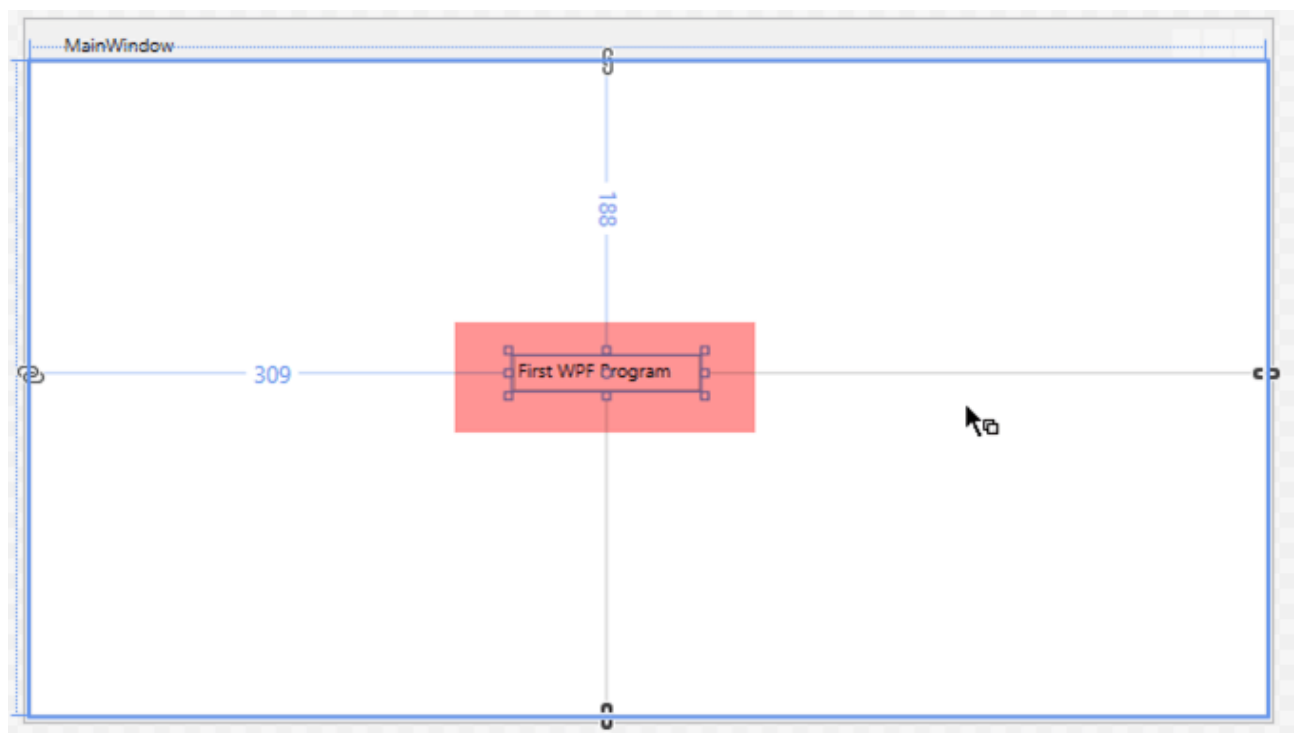
```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using System.Windows;
7 using System.Windows.Controls;
8 using System.Windows.Data;
9 using System.Windows.Documents;
10 using System.Windows.Input;
11 using System.Windows.Media;
12 using System.Windows.Media.Imaging;
13 using System.Windows.Navigation;
14 using System.Windows.Shapes;
15
16 namespace MyWPF
17 {
18     /// <summary>
19     /// Interaction logic for MainWindow.xaml
20     /// </summary>
21     public partial class MainWindow : Window
```

Langkah 4) Pada Toolbox,

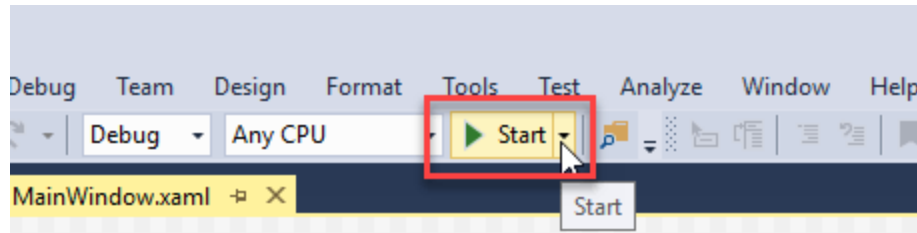
1. Drag n drop elemen TextBox ke jendela desain
2. TextBox akan muncul di jendela desain
3. Anda akan melihat kode XAML untuk TextBox ditambahkan



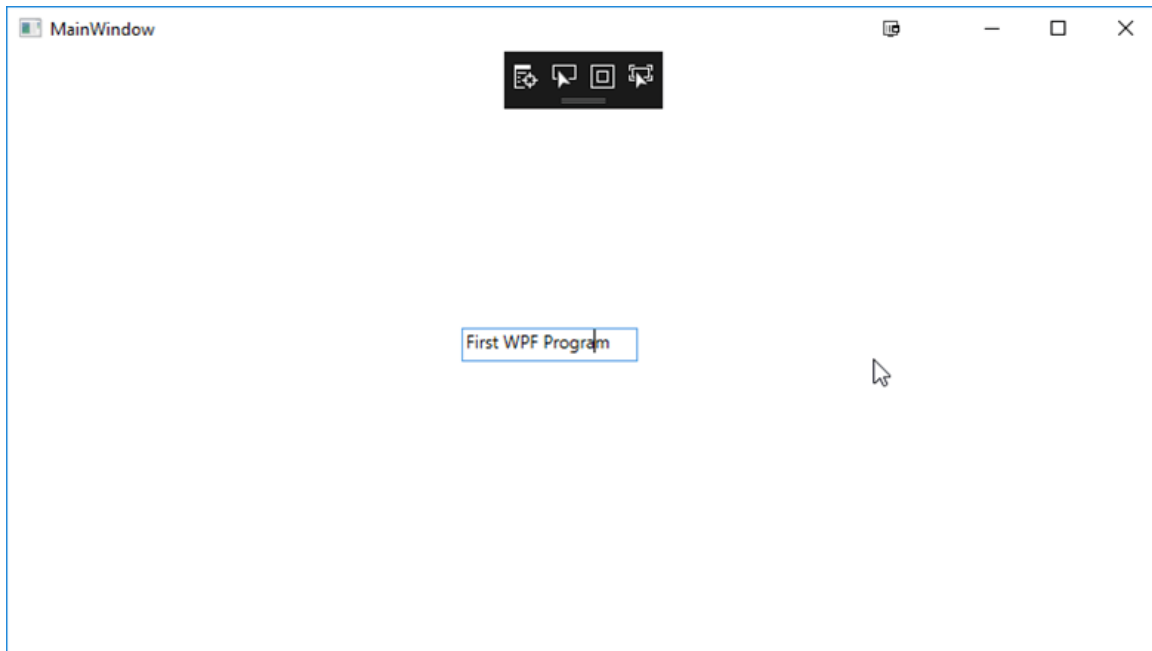
Langkah 5) Ubah teks menjadi "First Program WPF"



Langkah 6) Jalankan Aplikasi



Langkah 7) Anda akan melihat Window at Output



WPF vs WinForms

WPF	WinForms
Ini adalah kerangka kerja yang lebih baru, sehingga lebih selaras dengan standar saat ini.	Ini lebih tua, jadi lebih dicoba dan diuji.
Ini fleksibel dan kaya fitur. Anda dapat mendesain aplikasi yang sangat kaya tanpa coding atau membeli kontrol.	Tidak terlalu kaya fitur
Pengembang kontrol pihak ketiga difokuskan pada kompatibilitas dengan	Ada banyak kontrol pihak ketiga yang dapat Anda beli atau dapatkan secara gratis.

WPF karena ini adalah masa depan.

XAML memungkinkan Anda membuat dan mengedit GUI Anda dengan mudah. Ini memungkinkan pekerjaan dibagi antara seorang desainer (XAML) dan seorang programmer (C #, ASP.net, dll.).

Di Windows Forms semua kode ditulis di satu tempat.

WPF dapat membuat antarmuka pengguna untuk aplikasi Windows dan aplikasi web seperti Silverlight dan XBAP.

Tidak ada dukungan seperti itu

Suite kontrol dalam kotak WPF terbatas

Kontrol dalam kotak sangat kuat
