

## **BAB II**

### **PEMBAHASAN**

#### **2.1 RANCANGAN PROJECT WORK**

Disaat pemerintah memberlakukan peraturan untuk selalu tetap dirumah untuk mengurangi penyebaran wabah virus corona, SMK Telkom Sandhy Putra Jakarta ikut andil mendukung kegiatan tersebut. Hal itu membuat mereka mengharuskan membatalkan segala kegiatan yang menimbulkan terjadinya tatap muka. Mereka pun menggantinya dengan berbagai kegiatan yang bisa dilakukan tanpa harus melakukan tatap muka. *Project work* ini termasuk solusi agar tidak terjadinya tatap muka.

Dibanding melakukan Praktik Kerja Lapangan (PKL) yang seharusnya dilaksanakan, *project work* ini bisa mengurangi kegiatan bertatap muka walaupun dengan terpaksa siswa siswi tidak bisa merasakan rasanya bekerja di industri.

*Project work* ini mengharuskan siswa siswinya untuk tetap berpikir kreatif dalam membangun suatu produk walaupun disaat tengahnya pandemi ini. Mereka diharuskan untuk membuat suatu produk secara berkelompok sesuai dengan jurusan yang mereka tempati dan melaksanakannya dirumah masing-masing dengan komunikasi lewat daring atau *online*. Mereka juga mempunyai guru sebagai pembimbing yang akan membimbing mereka disaat *project work* sedang berlangsung. Masing-masing kelompok memiliki dua pembimbing, yaitu pembimbing akademis yang melakukan pemeriksaan terhadap penulisan laporan dan pembimbing teknis yang melakukan pemeriksaan terhadap project yang sedang mereka kerjakan.

Akan tetapi, sebelum mereka melaksanakan pengerjaan *project work* yang sesuai dengan kelompok masing-masing, mereka dibekali ilmu terlebih dahulu oleh pelatih yang sudah disediakan oleh sekolah untuk mempermudah dalam pengerjaan *project* nanti

Dalam pengerjaan *project work* ini, penulis memiliki kelompok yang mempunyai tugas masing-masing didalamnya yaitu berupa *system analyst* yang bertugas untuk membuat perencanaan setelah ide sudah ditentukan agar *project* bisa dilaksanakan dengan perencanaan tersebut, *programmer* yang bertugas untuk membangun aplikasi dari perencanaan yang telah dibuat sampai bisa dapat digunakan oleh siapapun, dan *marketing* yang bertugas untuk membikin suatu yang bisa membuat aplikasi yang sudah bangun bisa diperkenalkan atau dipromosikan kepada yang dituju agar bisa menjadi suatu produk yang laris. Disini penulis bertugas sebagai programmer.

### 2.1.1 LATAR BELAKANG PEMILIHAN JUDUL



Gambar 2.1 Logo OrHor  
kiri



Gambar 2.2 Logo OrHor  
kanan

Setelah berpikir lama tentang ide yang akan diimplementasikan ke dalam sebuah produk berupa aplikasi yang berbasis dekstop pada sistem operasi windows, kami sekelompok menyetujui akan membangun aplikasi yang bisa membuat sesama penyuka bisa berinteraksi satu sama lain khususnya kepada penyuka horror. Lalu, kami membuat judul *project work* ini dengan nama “Aplikasi Khusus Pencinta Horror” serta dengan nama aplikasinya yaitu “OrHor”.

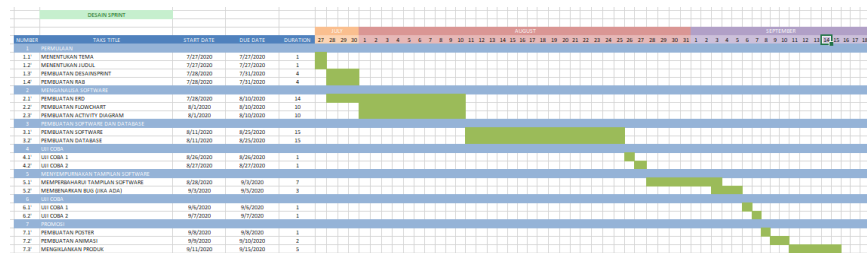
Latar belakang terbentuknya judul ini adalah dikarenakan kami menyadari bahwa beberapa orang yang menyukai akan suatu hal tidak bisa berkomunikasi untuk saling berbagi pengalaman dan hiburan secara khusus kepada sesama penyuka sejenis yang lainnya.

Dengan ide produk ini, bisa membuat orang yang memiliki satu ketertarikan memiliki komunikasi dengan yang memiliki ketertarikan yang sama di luar sana tanpa harus bertemu langsung layaknya sosial media.

Berdasarkan pemikiran kami, penyampaian akan suatu hal kepada orang yang memiliki ketertarikan yang sama akan terjalin hubungan komunikasi yang baik dan menyenangkan sehingga produk ini akan sangat diminati kepada khususnya penyuka berjenis horror. Disini kami juga memiliki pemikiran tambahan yang tidak terlalu dominan tentang aplikasi ini, yaitu bisa saling membantu terhadap seseorang yang sedang terkena ancaman dari suatu hal yang ghaib. Dan juga kami membuatnya agar aplikasi ini memiliki pembatasan usia sampai 18 tahun dikarenakan kami peduli terhadap kondisi psikologi orang terhadap sesuatu yang sadis, menegangkan dan menyeramkan.

### 2.1.2 JADWAL DAN RANCANGAN KERJA

Jadwal waktu yang kami sekelompok lakukan ketika pengerjaan *project* adalah berdasarkan perhitungan per hari. Dibawah ini terdapat gambar dari *design sprint* untuk penjadwalan kegiatan yang seharusnya kami lakukan sehari-hari. Tapi, sayangnya penjadwalan yang kami bikin tidak terlaksana dengan baik dikarenakan terdapat berbagai permasalahan sehingga untuk penjadwalan yang sebenarnya tidak sesuai dengan yang ada di *design sprint*.



Gambar 2.3 OrHor Design Sprint

## 2.2 LANDASAN TEORI

### 2.2.1 PROTOTYPE



Gambar 2.4 OrHor Prototype

*Prototype* adalah sebuah sample, model atau perilsan sebuah produk yang dibuat untuk *testing* sebuah konsep yang dibuat. *Prototype* digunakan untuk mengevaluasi *design* baru untuk meningkatkan tingkat akurasi dari analisis dan user sistem.

Dalam suatu ilmu *design process*, *prototype* adalah bagian yang krusial dan membutuhkan suatu latihan untuk digunakan dalam semua *design disciplines*. Arsitektur, engineer, desainer industri dan desainer servis membuat *prototype* untuk *testing design* mereka sebelum diluncurkan ke suatu produksi.

*Prototype* membuat designer bisa memvalidasi konsep design mereka terlebih dahulu sebelum dikirim ke produksi dengan ditujukan kepada pengguna asli dan mendapatkan laporan secepat mungkin dari mereka untuk diperbaiki lagi kedepannya.

Dalam pembuatan *prototype* bagian desainer industri atau desainer servis, terdapat beberapa aplikasi yang membantu desainer dalam pembuatan *design* khususnya dibagian *User Interface/User Experience (UI/UX) design*. Aplikasi tersebut dinamakan sebagai *prototyping tools*. Disini penulis menggunakan *prototyping tools* dengan nama Adobe XD.

#### a. ADOBE XD



Gambar 2.5 Logo Adobe XD

Adobe XD adalah *user experience design tool* berbasis vektor yang digunakan untuk keperluan pembangunan aplikasi *website* dan aplikasi *mobile* dan dibangun dan dipublish oleh Adobe Inc yang tersedia di sistem operasi windows dan macOS.

### 2.2.2 DEKSTOP APPLICATION DEVELOPER PLATFORM (WINDOWS)

Tidak memiliki penyebutan yang baku dalam hal ini. Digunakan agar *developer* bisa membangun aplikasi berbasis dekstop di sistem operasi windows. Memiliki banyak jenis yang bisa *developer* pilih untuk digunakan dalam pembuatan aplikasi. Disini penulis menggunakan *Windows Presentation Foundation (WPF)*.

### a. WINDOWS PRESENTATION FOUNDATION

Ini adalah *graphic subsystem* yang membantu pengguna untuk membuat tampilan saat mereka bekerja dengan windows *app development*. Didalam library dari WPF ini, terdapat .Net framework yang tercantum didalamnya. Salah satu fitur yang terdapat dalam WPF adalah menyatukan elemen dari *User Interface* (UI) yang dapat digunakan untuk merender 3D dan 2D, dokumen yang adaptif, *vector graphics*, *pre-rendered media* dan animasi berjalan.

### 2.2.3 .NET (DOT NET)



Gambar 2.6 Logo .NET

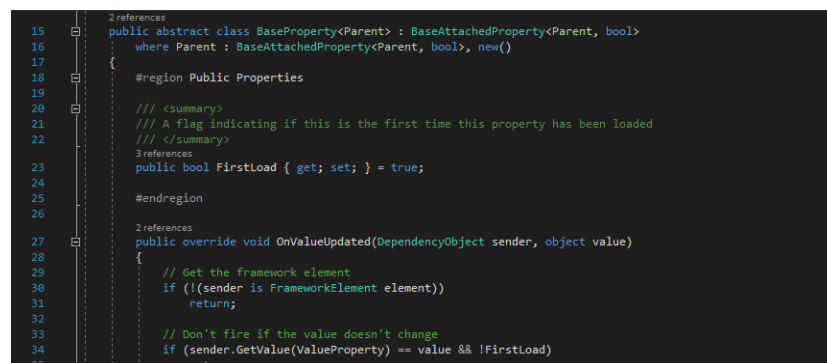
.NET adalah *open source developer platform* yang dibuat oleh Microsoft untuk membuat berbagai macam tipe aplikasi. .NET memiliki sifat yaitu gratis, *cross-platform*, dan *open source project*. Bisa juga digunakan untuk beberapa *programming languages* seperti C#, F# dan Visual Basic dan *libraries* untuk membuat *website*, mobile, dekstop, *games* dan IoT aplikasi.

.NET memiliki beberapa jenis implementasi yaitu .NET Core, .Net Framework dan Xamarin/Mono. Disini penulis menggunakan .Net Framework karena berkaitan erat dengan Windows Presentation Foundation (WPF).

### a. .NET FRAMEWORK

.NET Framework adalah *software development* framework untuk membuat dan menjalankan aplikasi di windows dan termasuk original implementasi dari .NET. Dapat melakukan pendukungan untuk menjalankan *websites*, *services*, desktop aplikasi dan masih banyak lagi di windows.

### 2.2.4 PROGRAMMING LANGUAGE



```

15 2 references
16 public abstract class BaseProperty<Parent> : BaseAttachedProperty<Parent, bool>
17     where Parent : BaseAttachedProperty<Parent, bool>, new()
18 {
19     #region Public Properties
20     /// <summary>
21     /// A flag indicating if this is the first time this property has been loaded
22     /// </summary>
23     3 references
24     public bool FirstLoad { get; set; } = true;
25     #endregion
26
27 2 references
28     public override void OnValueUpdated(DependencyObject sender, object value)
29     {
30         // Get the framework element
31         if (!sender is FrameworkElement element)
32             return;
33
34         // Don't fire if the value doesn't change
35         if (sender.GetValue(ValueProperty) == value && !FirstLoad)
36             return;

```

Gambar 2.7 Programming Language Aplikasi OrHor  
(C#)

*Programming* adalah sebuah instruksi yang diberikan kepada komputer untuk dieksekusi. Maka *programming language* adalah *tools* atau *languages* yang digunakan untuk menulis instruksi-instruksi agar komputer dapat mengikutinya.

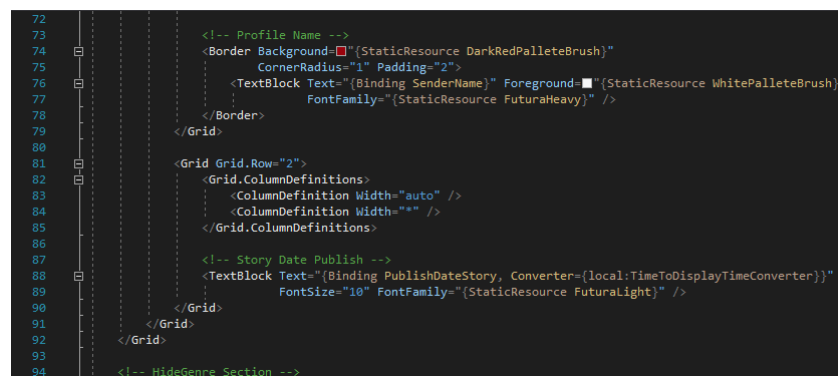
Diluar sana, terdapat banyak *programming languages* yang bisa digunakan dalam membuat suatu aplikasi atau produk yaitu diantaranya adalah Python, C#, C++, Java, dan masih banyak lagi. Disini penulis sendiri menggunakan bahasa C# dikarenakan sangat berkaitan dengan .NET Framework dan Windows Presentation Foundation (WPF).

### a. C#

C# adalah salah satu *programming language* yang sangat populer. Berdasarkan yang penulis lihat di suatu *website* bernama tiobe yang mengurutkan tingkat kepopuleran dari *programming languages*, C# mendapati urutan ke lima dari seluruh *programming languages*.

Dapat digunakan ke berbagai macam hal yaitu web aplikasi, dekstop aplikasi, mobile aplikasi, *game development*, *Virtual Reality* (VR), *Augmented Reality* (AR) dan *back end services*. Dalam hal ini, penulis menggunakannya sebagai dekstop aplikasi.

## 2.2.5 CODING LANGUAGE



```

72
73
74 <!-- Profile Name -->
75 <Border Background="{StaticResource DarkRedPalleteBrush}"
76         CornerRadius="1" Padding="2">
77     <TextBlock Text="{Binding SenderName}" Foreground="{StaticResource WhitePalleteBrush}"
78               FontFamily="{StaticResource FuturaHeavy}" />
79 </Border>
80 </Grid>
81
82 <Grid Grid.Row="2">
83     <Grid.ColumnDefinitions>
84         <ColumnDefinition Width="auto" />
85         <ColumnDefinition Width="*" />
86     </Grid.ColumnDefinitions>
87
88     <!-- Story Date Publish -->
89     <TextBlock Text="{Binding PublishDateStory, Converter={local:TimeToDisplayTimeConverter}}"
90               FontSize="10" FontFamily="{StaticResource FuturaLight}" />
91 </Grid>
92 </Grid>
93
94 <!-- HideGenre Section -->

```

Gambar 2.8 Coding Language Aplikasi OrHor (XAML)

Hanya sebagai perbedaan pendapat di antara kalangan *programmer* dalam hal ini. Memiliki pengertian yang sama dengan *programming language*. Namun, menurut penulis dan *programmer-programmer* yang sepaham, ada perbedaan diantara *programming language* dengan *coding language*.



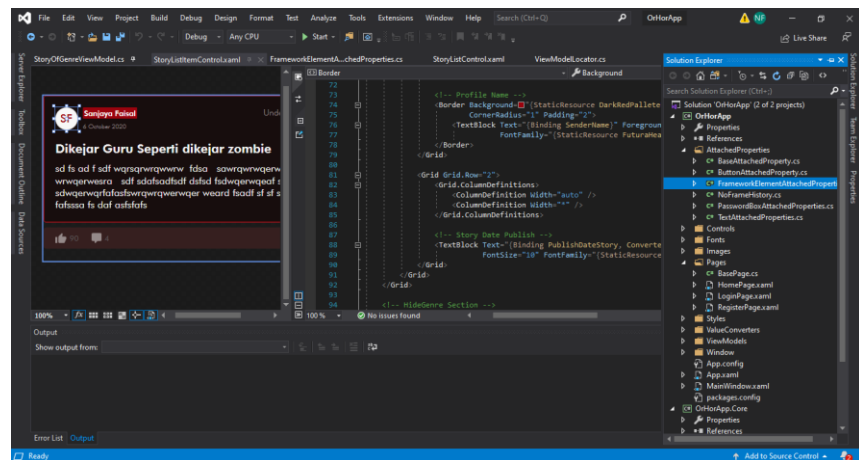
Perbedaannya terletak dimana *programming language* merupakan *language* yang seharusnya berisi sesuatu yang logikal seperti terdapat *decision*, *loop* atau semacamnya. Sementara *coding language* memiliki perbedaan yang terletak dimana ini hanya versi ringan dari *programming language* dan tidak memiliki semacam sesuatu yang logikal. Contoh *language* yang termasuk *coding language* yaitu *Hyper Text Markup Language* (HTML), *Cascading Style Sheet* (CSS), *Extensible Markup Language* (XML), *Extensible Application Markup Language* (XAML) dan semacamnya. Bahkan salah satu dari mereka memiliki pengertian bahwa itu termasuk *markup language*. Disini penulis menggunakan *Extensible Application Markup Language* (XAML) dikarenakan berkaitan erat dengan *Windows Presentation Foundation* (WPF).

#### a. **EXTENSIBLE APPLICATION MARKUP LANGUAGE**

Ekstensible Application Markup Language (XAML) termasuk varian dari Microsoft yang diambil berdasarkan *Extensible Markup Language* (XML) dimana *developers* dan desainer bisa berbagi dan mengedit konten secara bebas tanpa melakukan semacam kompilasi. XAML juga memiliki manfaat untuk membangun sebuah *User Interface* (UI) dengan membuat prosedural *code* agar dapat menghasilkan sesuatu daripada mendeklarnya.

Dengan mendefinisikan UI dengan cara membentuk prosedural code, membuat *developers* bisa membuat UI tersebut menjadi lebih atraktif, statik, responsif, dan masih banyak lagi. Dan juga XAML ini memiliki sintaks yang hampir sama dengan kebanyakan *markup language*.

## 2.2.6 INTEGRATED DEVELOPMENT ENVIRONMENT



Gambar 2.9 IDE Aplikasi OrHor  
(Visual Studio 2019)

*Integrated Development Environment* (IDE) dapat meningkatkan produktifitas seorang programmer dengan cara mengkombinasi segala aktifitas menulis *software* menjadi sebuah aplikasi, seperti mengedit *source code*, membangun sesuatu yang di eksekusi, dan *debugging*.

IDE ini memiliki fitur yang sangat berguna untuk mempercepat tingkat produktifitas seorang *programmer* seperti sintaks *highlighting*, *autocomplete* dan *debugging*. Terdapat berbagai macam IDE diluar sana dan masing-masing dari mereka memiliki kegunaan yang berbeda seperti Microsoft Visual Studio, NetBeans, PyCharm dan masih banyak lagi. Disini penulis menggunakan Microsoft Visual Studio dikarenakan sangat berkaitan erat terhadap .NET Framework, Windows *Presentation Foundation* (WPF), C# *programming language* dan XAML *coding language*.

### a. MICROSOFT VISUAL STUDIO



Gambar 2.10 Logo  
Visual Studio

Microsoft Visual Studio adalah *Integrated Development Environment* (IDE) yang dibuat oleh Microsoft yang digunakan untuk membuat web aplikasi, dekstop aplikasi, mobile aplikasi dan masih banyak lagi.

Visual Studio berisi *code editor* dengan dukungan berupa IntelliSense dan built-in tools berupa *code profiler*, desainer untuk membuat *Graphical User Interface* (GUI) aplikasi, web desainer, *class* desainer dan *database* skema desainer. Visual Studio juga mendukung 36 *programming language* yang berbeda berupa C#, F#, Visual Basic, C++, dan masih banyak lagi.

### 2.2.7 SISTEM OPERASI

Sistem Operasi (SO) atau dalam bahasa Inggris *Operating System* (OS) adalah *software* yang berperilaku sebagai tampilan antara pengguna dengan *hardware* komputer.

Dalam komputer, harus memiliki setidaknya satu OS didalamnya untuk menjalankan segala program yang ada. OS membantu berkomunikasi dengan komputer tanpa harus mengetahui cara berbicara dengan *language* komputer.

Terdapat berbagai macam OS yang bisa digunakan yaitu Windows, Linux, MacOS dan masih banyak lagi. Disini penulis menggunakan OS Windows dengan versi ke 10 atau disebut sebagai Windows 10. Penulis menggunakan OS ini dikarenakan sangat berkaitan erat dengan project yang sedang dikerjakan dan sangat berhubungan dengan semua *tools* yang ada.

**a. WINDOWS 10**



Gambar 2.11 Logo Windows 10

Seris dari OS yang dibangun oleh Microsoft dan dirilis sebagai bagian dari keluarga Windows NT OS.

Fitur yang paling dikenal adalah dukungan untuk aplikasi universal yang termasuk ekspansi dari *Metro-style* yang pertama kali di perkenalkan di seri Windows 8. Aplikasi universal di desain agar berjalan di seluruh produk yang dibuat oleh Microsoft seperti *Personal Computer* (PC)s, *tablets*, *smartphones* dan masih banyak lagi.

## 2.3 PENGEMBANGAN APLIKASI PENCINTA KHUSUS HORROR (OrHor)

### 2.3.1 PERSIAPAN

#### a. ALAT

- Komputer bisa berupa laptop atau *Personal Computer* (PC) dengan minimum persyaratan rekomendasi dari penulis yaitu *Random Access Memory* (RAM) 4 Giga Byte (GB), *processor* dengan kecepatan 2.0 Giga Hertz (GHz), *video card* yang mendukung resolusi 1366 x 768, penyimpanan mencapai 60 GB bebas ruang dan Sistem Operasi rilis Windows 10 versi 1703. Sebagai alat untuk berinteraksi dengan aplikasi-aplikasi yang dibutuhkan dan pelaksanaan *testing* dalam *project*.
- Internet, buku atau sumber apapun sebagai tempat menyelesaikan suatu *bug* atau *error* yang ada didalam aplikasi.
- Adobe XD versi 27.1.12.4 atau yang terbaru *prototyping tool*. Sebagai alat untuk meningkatkan akurasi analisis *design* dan memudahkan desainer dalam membentuk sebuah *design* yang bisa digunakan langsung sebelum dikirim ke produksi.
- Microsoft Visual Studio 2019 versi 16.7.4 atau yang terbaru *Integrated Development Environment* (IDE). Sebagai alat untuk menulis *software*, *debugging*, *testing* dan membangun seluruh *code* menjadi aplikasi yang bisa dieksekusi.

**b. BAHAN**

- *Windows Presentation Foundation (WPF) Desktop application developer platform.* Digunakan agar mendapat dukungan untuk membuat aplikasi di Sistem Operasi (SO) Windows.
- *Microsoft .NET Framework versi 4.8.04084* atau yang terbaru. Digunakan sebagai pendukung untuk *Windows Presentation Foundation (WPF)*.
- *C# programming language* dan *Extensible Application Markup Language (XAML) coding language.* Digunakan untuk memberikan instruksi agar bisa dipahami oleh komputer dan merupakan kebutuhan untuk menggunakan *Windows Presentation Foundation (WPF)*.

**2.3.2 PELAKSANAAN**

Sebelum menuju penjelasan tentang pelaksanaan, disini penulis ingin memberitahu kepada pembaca bahwa pelaksanaan ini tidak bisa dijelaskan secara lengkap. Dikarenakan *project* ini masih dalam pengerjaan proses sampai 30%. Penulis meminta maaf atas kejadian yang tidak diharapkan ini.

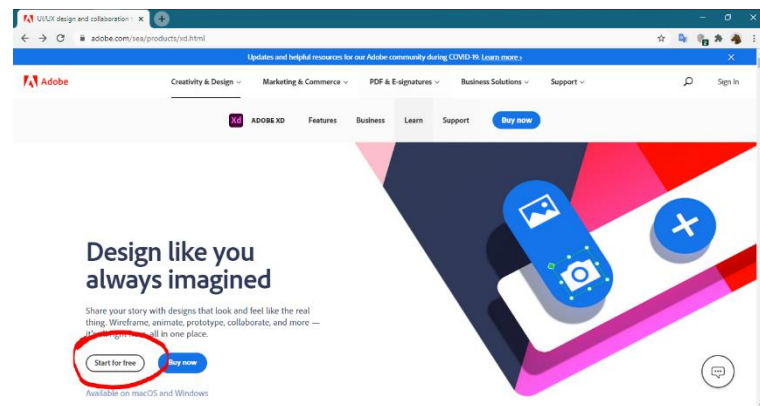
## a. PROTOTYPING

Berikut akan dijelaskan bagaimana cara pembuatan *prototype* di aplikasi Adobe XD.

1. Unduh dan install aplikasi Adobe XD di *official website*.

Berikut link pengunduhan :

<https://www.adobe.com/sea/products/xd.html>



Gambar 2.12 Adobe XD Official Website

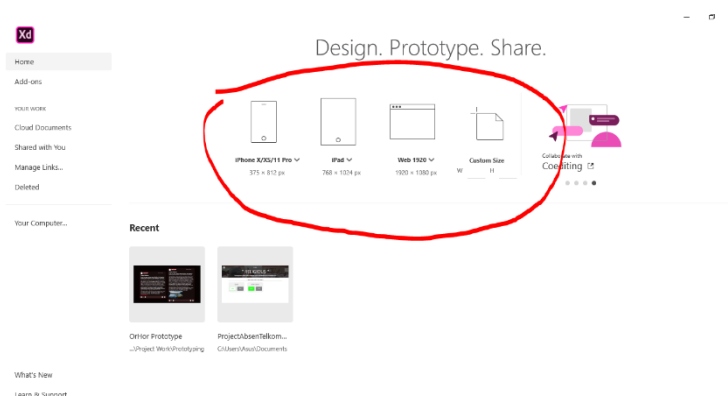
Berikut link jika membutuhkan bantuan dalam pengunduhan :

<https://helpx.adobe.com/xd/get-started.html>

Berikut link jika memiliki masalah terhadap pengunduhan :

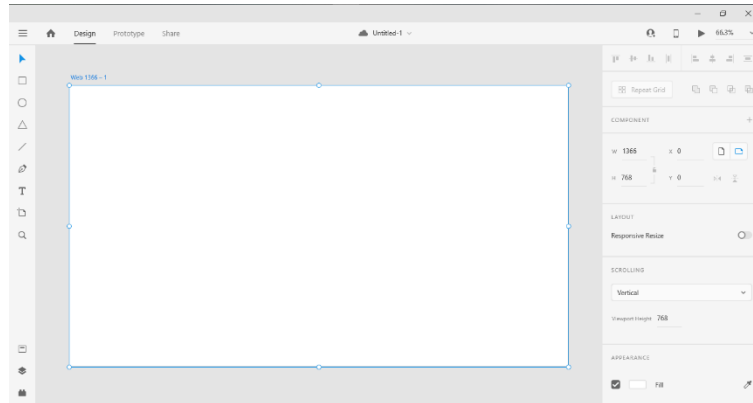
<https://helpx.adobe.com/download-install.html>

2. Buka Adobe XD yang sudah selesai diinstal dan pengguna akan ditujukan kepada tampilan awal dari aplikasi Adobe XD. Lalu, pilih salah satu resolusi yang di butuhkan untuk memulai project.



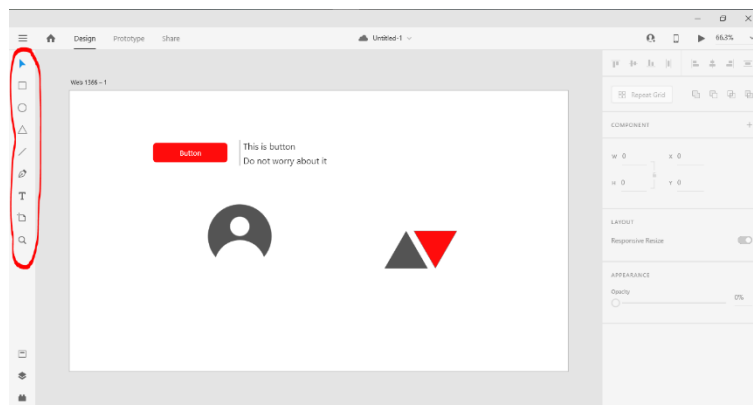
Gambar 2.13 Adobe XD Tampilan Awal

- Setelah memilih resolusi, akan ditujukan ke tampilan awal dalam pembuatan *prototype* yang akan dilakukan disebut sebagai *Artboard*.



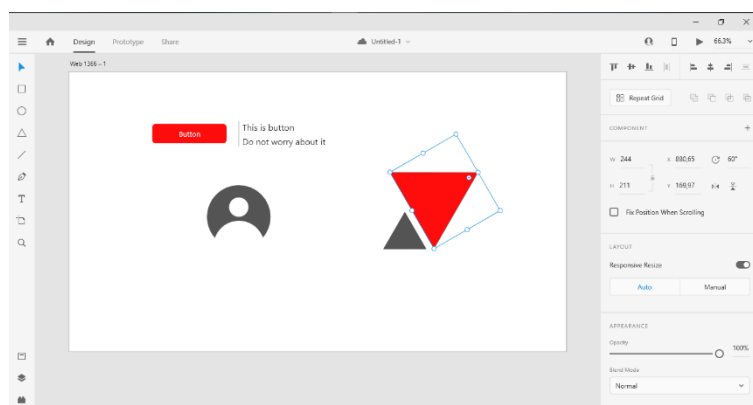
Gambar 2.14 Adobe XD Artboard

- Disamping kiri, terdapat berbagai *tools* yang bisa digunakan untuk membuat *prototype*. Berikut juga terdapat contoh dari hasil penggunaannya.



Gambar 2.15 Adobe XD Tools

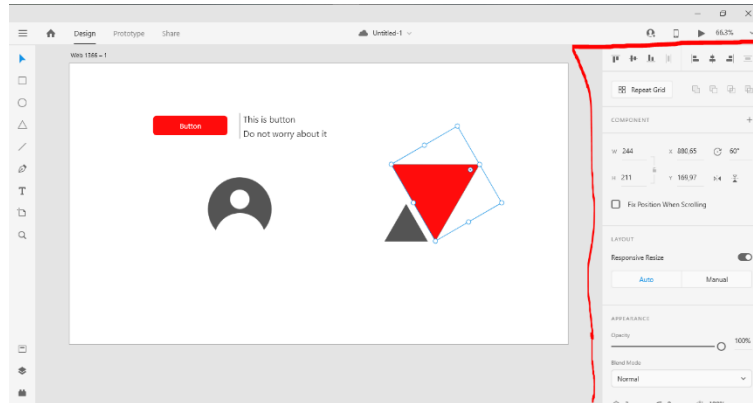
- Saat salah satu dari objek ditekan, maka terdapat fungsi yang bisa membesarkan dan memutar objek.



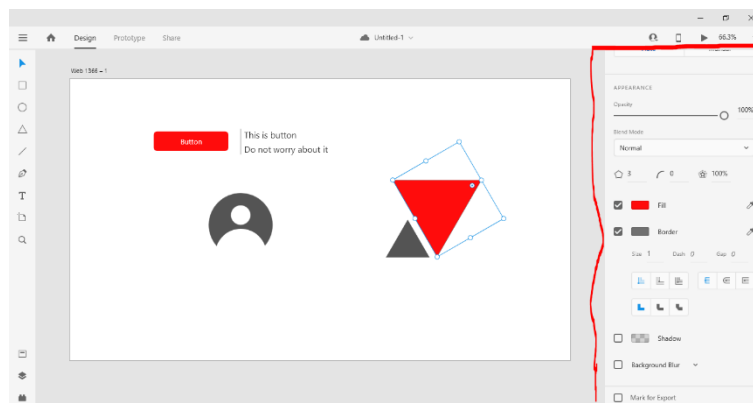
Gambar 2.16 Adobe XD Perbesar Pemutar



6. Disamping kanan, terdapat fitur tambahan terhadap objek yang dipilih yang bisa digunakan untuk pewarnaan, *alignment*, dan masih banyak lagi.

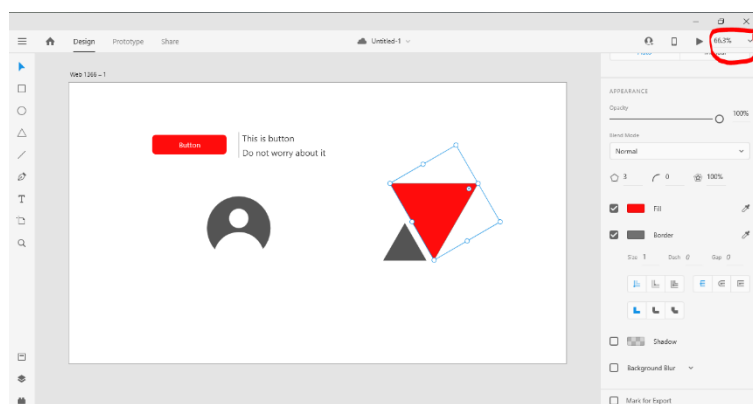


Gambar 2.17 Adobe XD Fitur Tambahan (1)



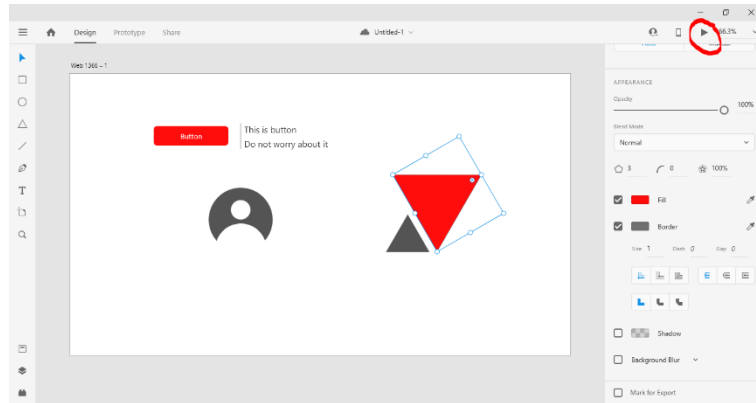
Gambar 2.18 Adobe XD Fitur Tambahan (2)

7. Disamping kanan atas bagian paling kanan, terdapat pembesar dimana pengguna bisa membesarkan dan mengecilkan layar pada *Artboard*.

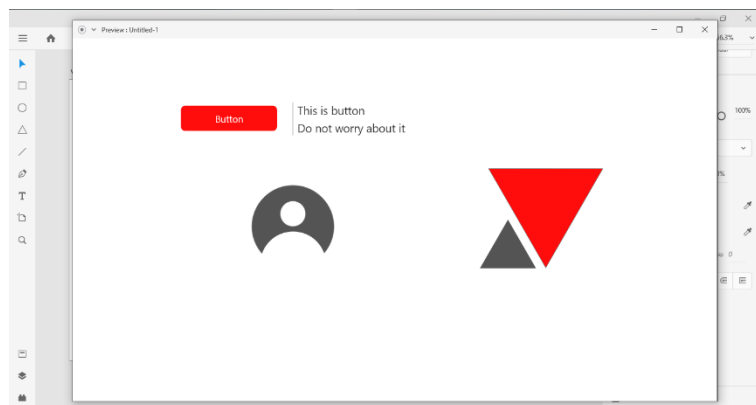


Gambar 2.19 Adobe XD Perbesar Layar

8. Disamping kanan atas bagian kedua dari kanan, terdapat *play mode* dimana pengguna dapat melihat keseluruhan *design* secara nyata dari setiap *artboard*.

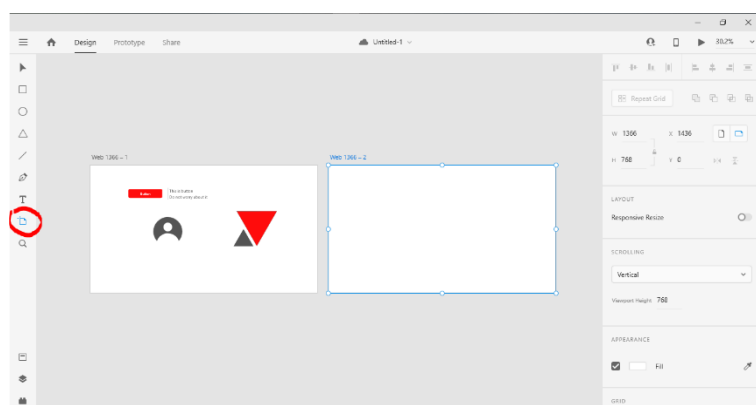


Gambar 2.20 Adobe XD Play Mode Tombol



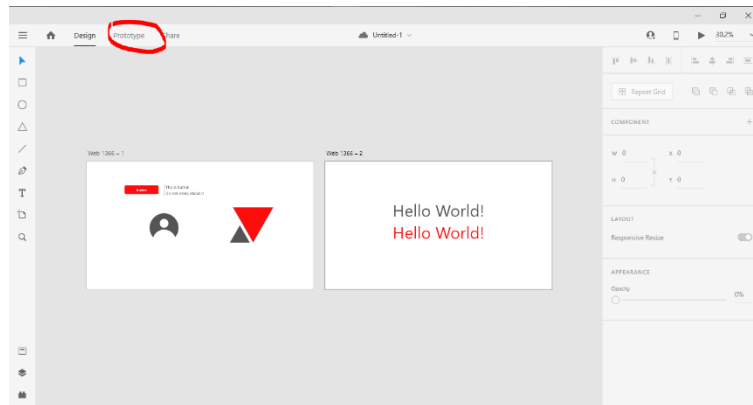
Gambar 2.21 Adobe XD Play Mode

9. Salah satu fitur yang terkenal dan penting di Adobe XD adalah penambahan *artboard*. Saat menggunakannya, pengguna dapat menambah artboard yang berfungsi seperti menambah atau dapat mengganti halaman.

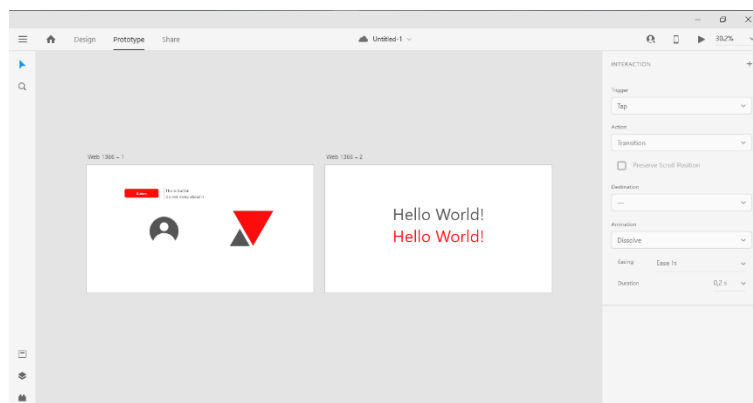


Gambar 2.22 Adobe XD Penambah Artboard

10. Dibagian atas terdapat tab untuk mengganti ke bagian *prototype*. Dapat berfungsi untuk melakukan *prototyping* yang sebenarnya, yaitu mengarahkan halaman menggunakan sebuah objek ketika objek tersebut mendapatkan input dari pengguna berupa tekan objek.

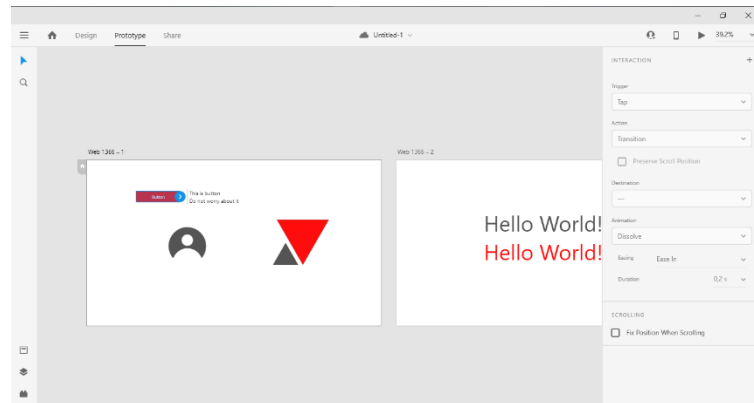


Gambar 2.23 Adobe XD Prototype Tab

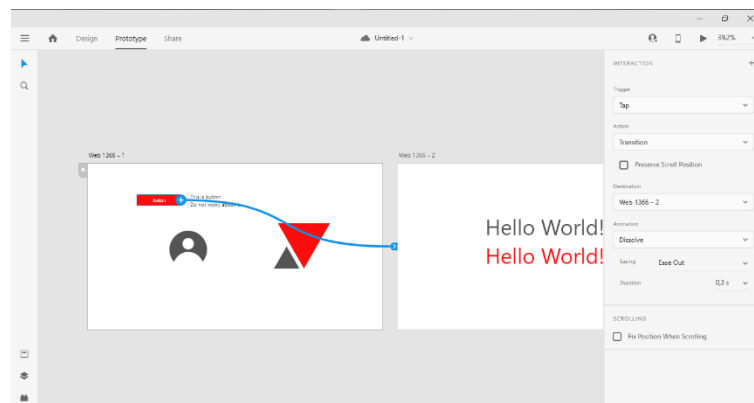


Gambar 2.24 Adobe XD Prototype

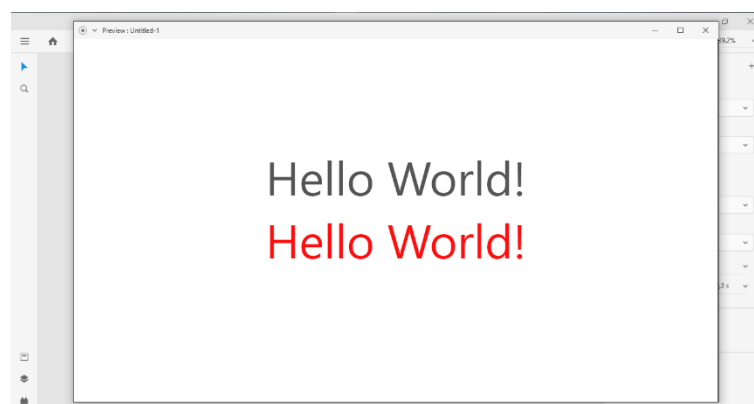
11. Cara menggunakannya adalah dengan menekan suatu objek. Lalu seret dan arahkan ke halaman yang dituju agar saat objek tersebut ditekan dapat berganti ke halaman yang diinginkan. Dan ketika saat *play mode*, objek tersebut bisa digunakan untuk berganti halaman ketika objek tersebut ditekan.



Gambar 2.25 Adobe XD Penggunaan Prototype (1)

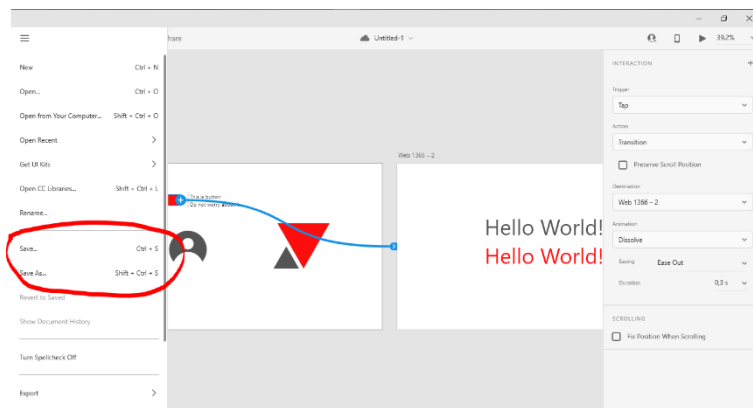


Gambar 2.26 Adobe XD Penggunaan Prototype (2)

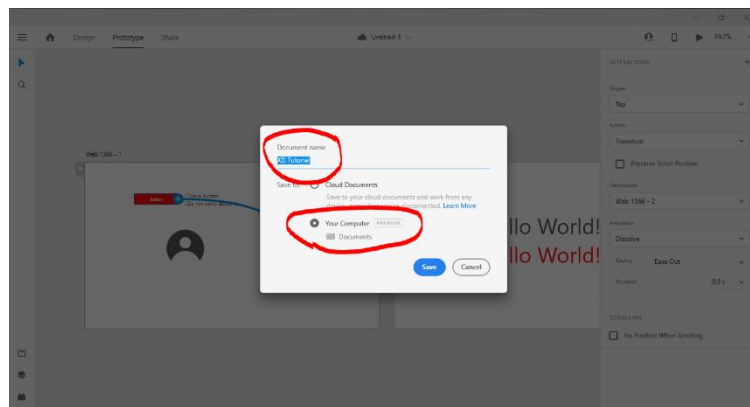


Gambar 2.27 Adobe XD Penggunaan Prototype (3)

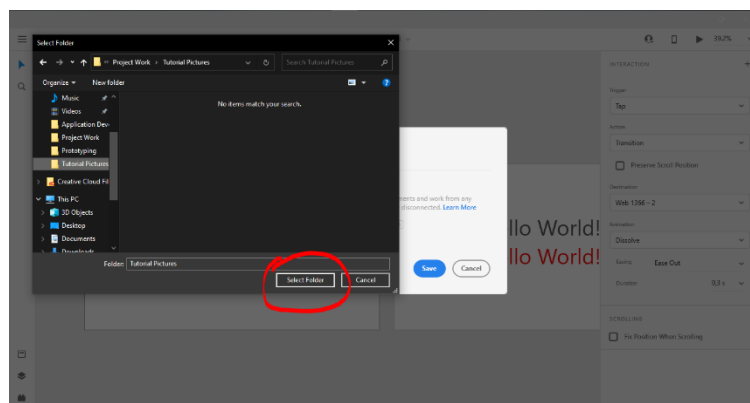
12. Terakhir yang paling penting harus diketahui adalah bagaimana cara menyimpan file tersebut agar tidak kehilangan proses yang sedang dikerjakan. Dibagian kiri paling atas terdapat tiga horizontal baris. Tekan dan akan terlihat tombol untuk menyimpan. Setelah itu, ikuti saja alur menyimpan di beberapa gambar di bawah ini. Dan penamaan file di paling atas berubah menjadi file yang pengguna namakan.



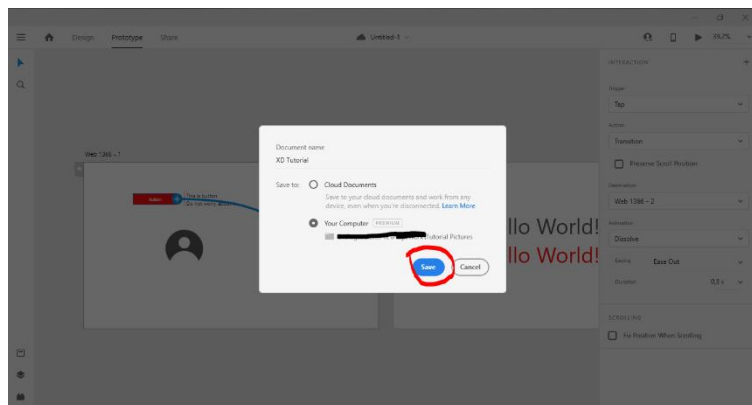
Gambar 2.28 Adobe XD Tombol Simpan



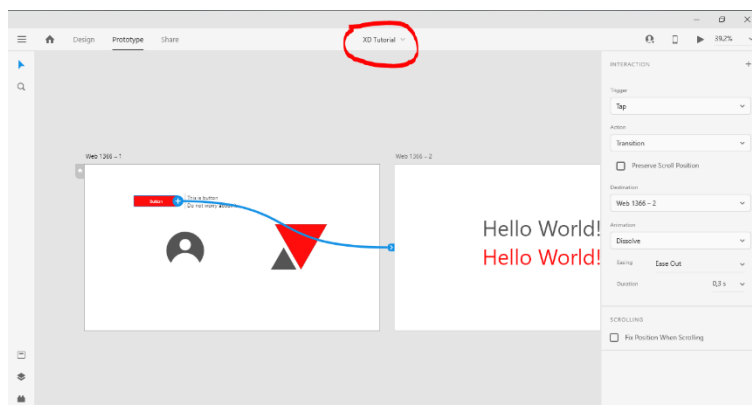
Gambar 2.29 Adobe XD Penamaan File



Gambar 2.30 Adobe XD Pilih Lokasi Folder Simpan



Gambar 2.31 Adobe XD Tekan Tombol Simpan



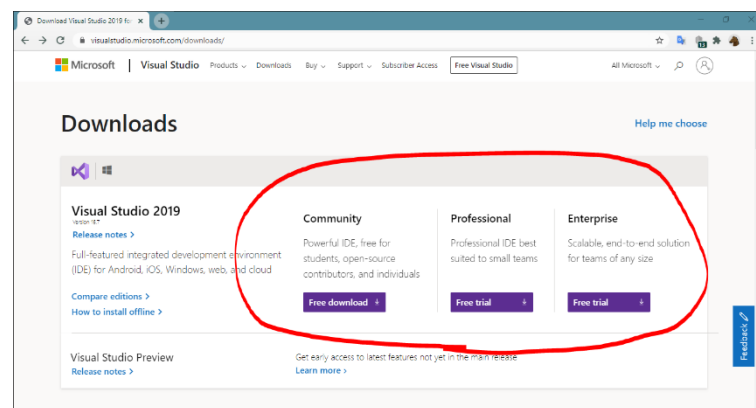
Gambar 2.32 Adobe XD Nama File

## b. APPLICATION DEVELOPMENT

1. Unduh dan install aplikasi Visual Studio 2019 di *official website*. Saat menuju bagian pengunduhan di *official website* Visual Studio, Visual Studio menyediakan beberapa edisi yang berbeda untuk produknya yaitu diantaranya adalah Visual Studio *Community*, Visual Studio Professional dan Visual Studio *Enterprise*. Penulis menyarankan untuk memilih Visual Studio *Community* dikarenakan pembaca dan penulis tidak menggunakannya untuk sebuah keperluan industri ataupun bisnis. Dan sekaligus menghemat pengeluaran dikarenakan ini bersifat gratis.

Berikut link pengunduhan :

<https://visualstudio.microsoft.com/downloads/>



Gambar 2.33 Visual Studio Official Website

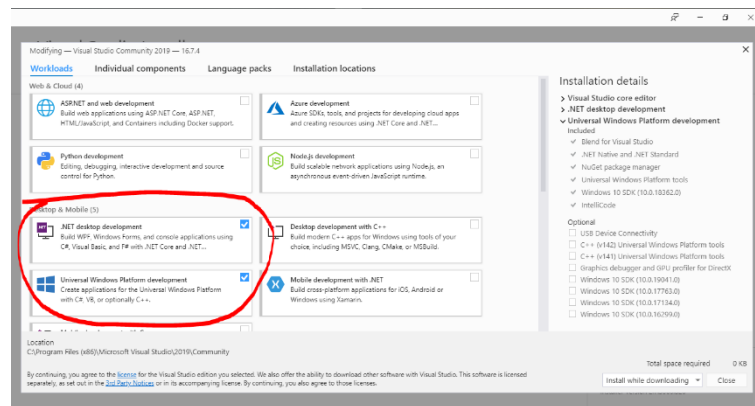
Berikut link jika membutuhkan bantuan dalam pengunduhan :

<https://docs.microsoft.com/en-us/visualstudio/install/install-visual-studio?view=vs-2019>

Berikut link jika memiliki masalah terhadap pengunduhan :

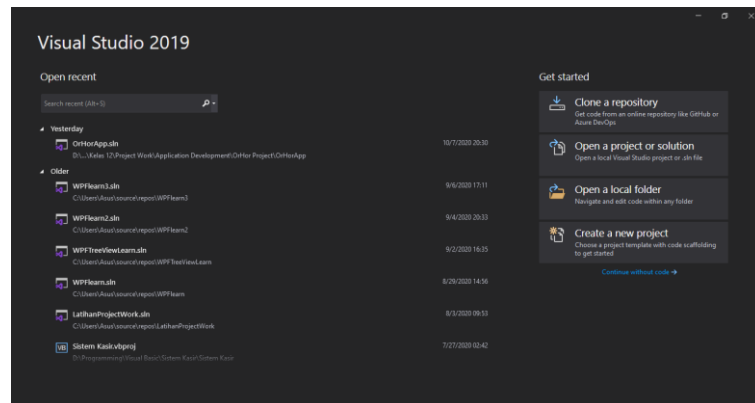
<https://docs.microsoft.com/en-us/visualstudio/install/troubleshooting-installation-issues?view=vs-2019>

2. Saat penginstalan sudah mencapai untuk memilih *workloads*, pastikan internet tersedia dan pilih bagian *.NET Desktop Development* dan *Universal Windows Platform Development* untuk mendapatkan yang dibutuhkan seperti *Windows Presentation Foundation (WPF)* dan *.NET Framework*. Setelah itu, unduh kedua tersebut dan tunggu hingga selesai agar siap untuk digunakan.



Gambar 2.34 Visual Studio Workloads

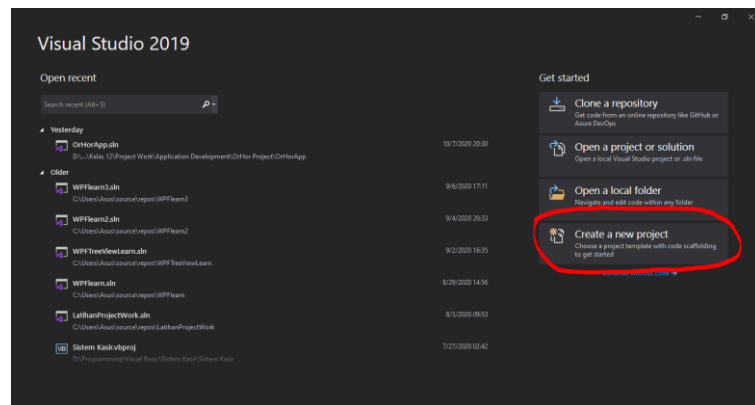
3. Setelah semua pengunduhan telah selesai, buka Visual Studio 2019 dan pengguna akan ditunjukkan kepada penampilan pertama dari Visual Studio.



Gambar 2.35 Visual Studio Tampilan Pertama

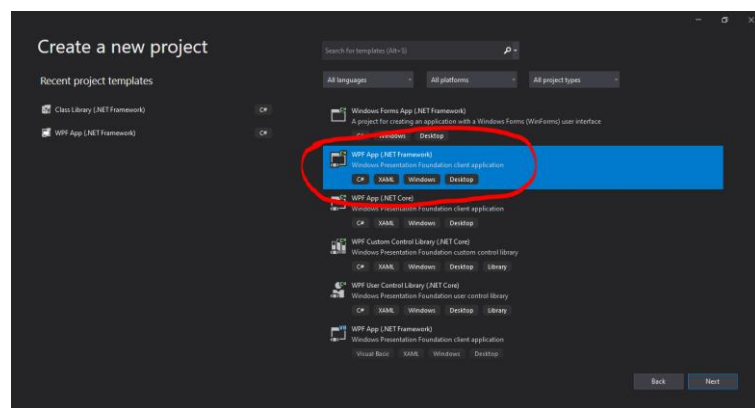


4. Dibagian kanan paling bawah, pilih membuat *project* agar bisa membuat *project* yang diinginkan.



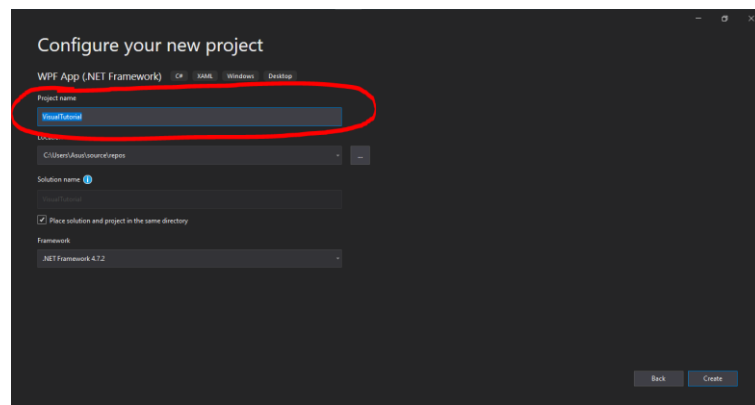
Gambar 2.36 Visual Studio Bikin Project

5. Pada tampilan ini, disediakan banyak sekali *project templates* yang bisa digunakan. Cari yang penggunaannya untuk *Windows Presentation Foundation* (WPF) dengan menggunakan .NET Framework. Pastikan untuk tidak salah memilih antara .NET Framework dan .NET Core karena kedua hal itu sangat berbeda.



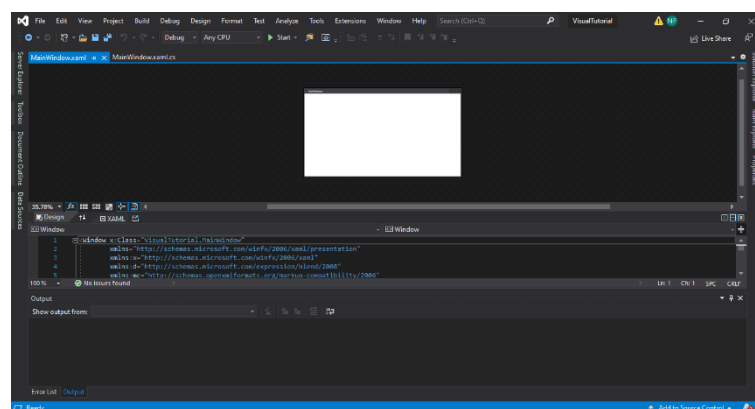
Gambar 2.37 Visual Studio Project Baru

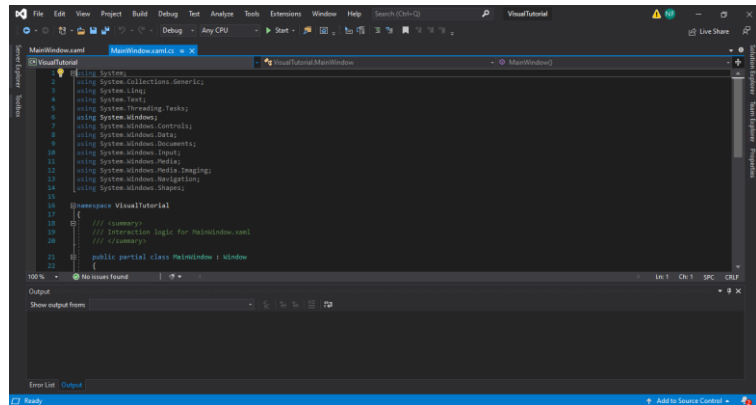
6. Setelah menemukan dan memilih yang dibutuhkan, pengguna akan ditunjukan ke halaman selanjutnya. Disini, pengguna bisa menamakan file, menempatkan lokasi penempatan file dan memilih versi dari framework yang akan digunakan. Pastikan untuk tidak memberikan penamaan file dengan spasi. Walaupun Visual Studio memperbolehkan dengan spasi tapi penulis lebih menyarankan untuk tidak menggunakan spasi dikarenakan akan terdapat kesalahpahaman sebuah sistem dalam penamaan file di pertengahan pembuatan *project* kelak.



Gambar 2.38 Visual Studio Konfigurasi Project

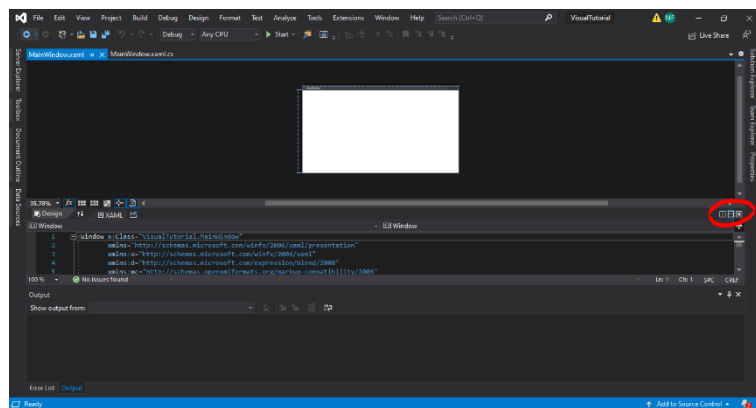
7. Disini, pengguna akan ditunjukan tampilan pertama dari sebuah *project*. Terdapat dua file yang disediakan yaitu *MainWindow.xaml* yang merupakan *Extensible Application Markup Language (XAML) coding language* dan *MainWindow.xaml.cs* yang merupakan *C# programming language*.





Gambar 2.40 Visual Studio MainWindow.xaml.cs C#

8. Dibagian paling kanan pertengahan, terdapat fitur yang membuat *layout* dari file XAML ini menjadi *vertical* ataupun *horizontal*. Pengguna bisa bebas menentukan *layout* yang cocok untuk dipandang.



Gambar 2.41 Visual Studio XAML Layout

9. Dalam tahap ini, penulis akan memberikan petunjuk pengimplementasian programming atau coding dalam project ini sekaligus memberikan sedikit pelajaran dasar dari XAML.

10. Yang paling terkenal dalam XAML coding adalah *grid*. Grid memiliki kegunaan agar pengguna bisa menentukan posisi dari suatu objek secara statik atau responsif.

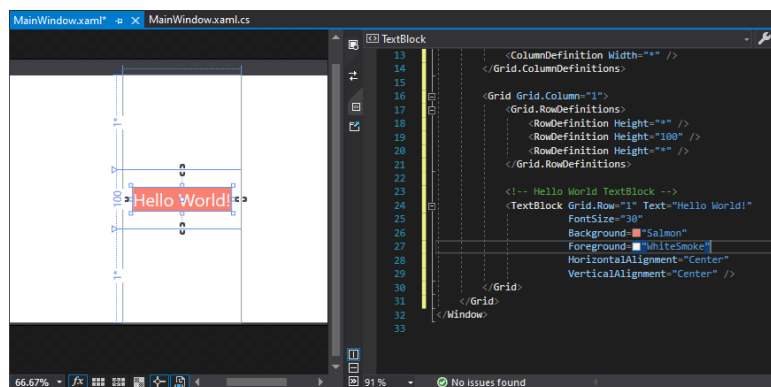
```

1 Window x:Class="VisualTutorial.MainWindow"
2   xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
3   xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
4   xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
5   xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
6   mc:Ignorable="d"
7   Title="MainWindow" Height="450" Width="800">
8
9   <Grid>
10    <Grid.ColumnDefinitions>
11      <ColumnDefinition Width="*" />
12      <ColumnDefinition Width="200" />
13      <ColumnDefinition Width="*" />
14    </Grid.ColumnDefinitions>
15
16    <Grid Grid.Column="1">
17      <Grid.RowDefinitions>
18        <RowDefinition Height="*" />
19        <RowDefinition Height="100" />
20        <RowDefinition Height="*" />
21      </Grid.RowDefinitions>
22    </Grid>
23  </Grid>
24 </Window>

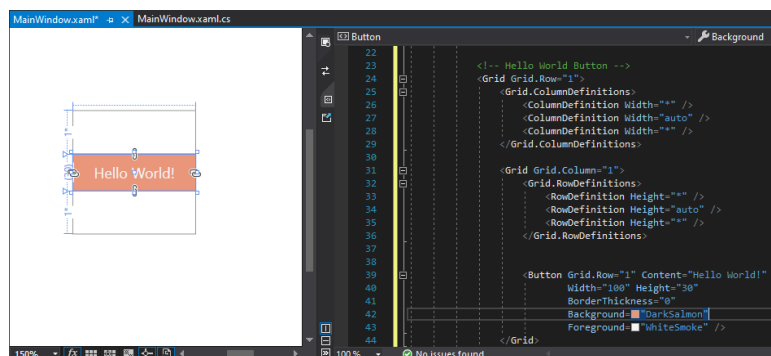
```

Gambar 2.42 Visual Studio Grid XAML

11. Selanjutnya pemberian objek beserta tampilan objek di halaman yang kita gunakan. Objek yang digunakan sangat familiar bagi pengguna di dunia serba teknologi ini seperti TextBlock, Button, TextBox, dan semacamnya.

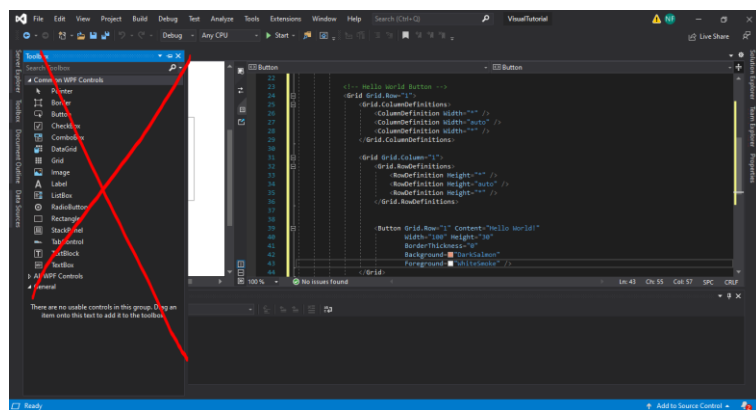


Gambar 2.43 Visual Studio TextBlock XAML



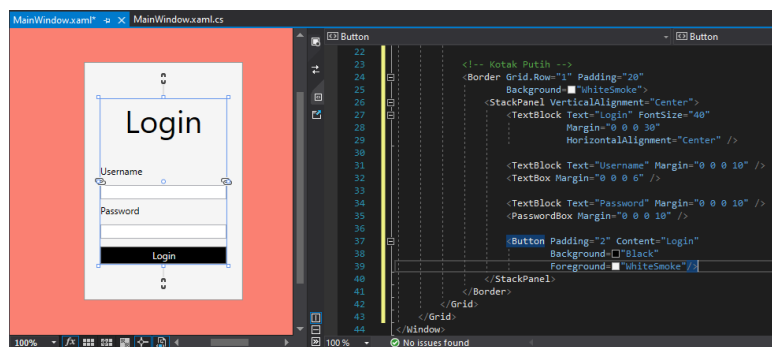
Gambar 2.44 Visual Studio Button XAML

12. Dibagian paling kiri dengan tulisan 90 derajat itu, terdapat *toolbox*. Toolbox ini berguna agar pengguna bisa membuat tampilan hanya dengan seret dan taruh. Namun di dunia XAML coding, developers sangat melarang penggunaan seret dan taruh dikarenakan akan terbentuk sebuah kodongan yang sangat berantakan dan tidak diinginkan. Ditambah tampilan dari *User Interface* (UI) yang dibikin dari seret dan taruh ini, akan membuat tampilan tidak rapih dan tidak bersih.



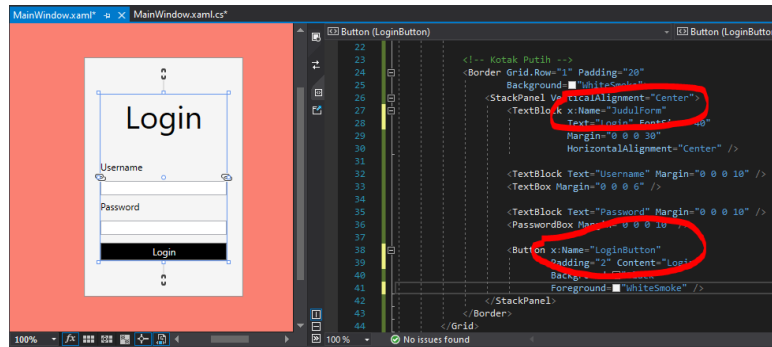
Gambar 2.45 Visual Studio Toolbox

13. Selanjutnya, terdapat fitur *StackPanel* yang membuat seluruh objek yang terdapat didalamnya akan memiliki ukuran yang sama dan pasti. Disini, penulis membuat form login dengan menggunakan *StackPanel*.



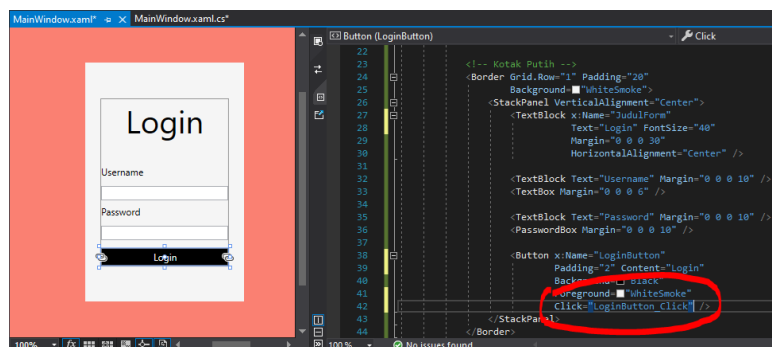
Gambar 2.46 Visual Studio StackPanel XAML

14. Selanjutnya, terdapat fitur dari WPF yang membuat pengguna bisa terhubung dengan C# atau bagian dalam yaitu `x:Name`. Fitur ini berguna untuk menamakan sebuah objek sehingga bisa digunakan di file C#.



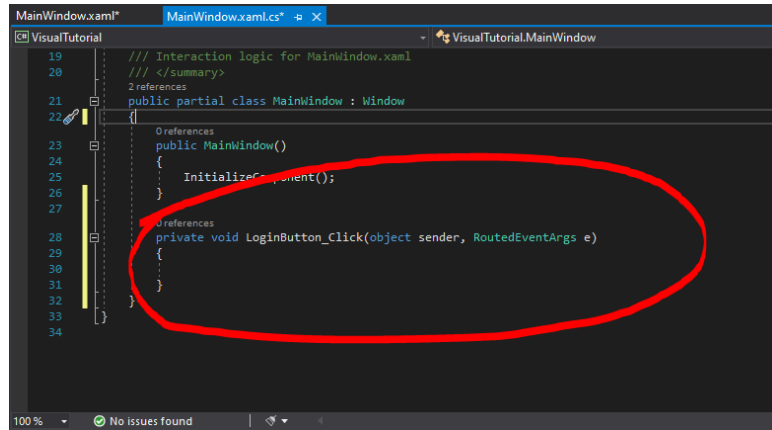
Gambar 2.47 Visual Studio Nama Objek XAML

15. Dalam WPF terdapat pembagian objek. Salah satunya adalah WPF control. WPF control memiliki kegunaan untuk melakukan aksi suatu objek yang didapat dari input pengguna sehingga bisa menghubungkan ke C# atau bagian dalam untuk dimanipulasi sebuah nilai dari objek. Salah satu dari WPF control adalah Button. Disini, penulis memberikan instruksi kepada button agar saat button tersebut mendapat input dari pengguna, akan langsung dihubungkan ke sebuah *method* yang ada di C#.



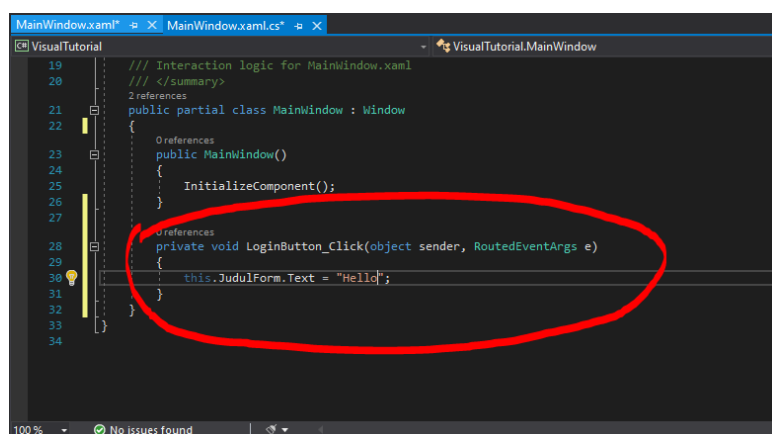
Gambar 2.48 Visual Studio Aksi Objek XAML

16. Dari hasil penamaan yang dilakukan oleh penulis di poin 14, membuat penulis bisa memanipulasi nilai dari objek tersebut saat *method* tersebut jalan di C#.



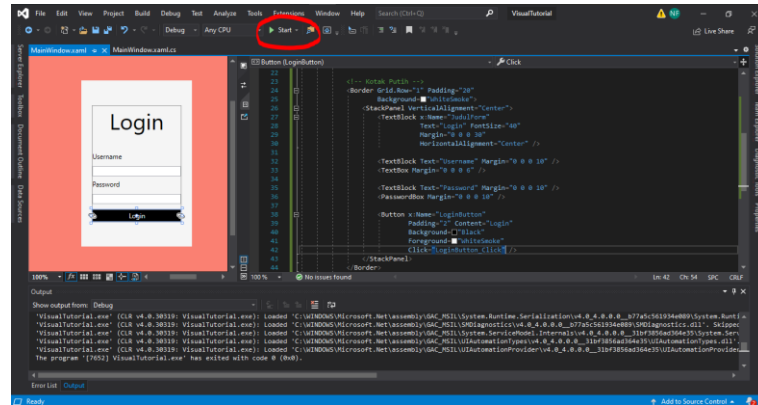
Gambar 2.49 Visual Studio LoginButton Method C#

17. Terdapat fitur yang sangat terkenal di dunia Integrated Development Environment (IDE) ini yaitu *debugging* dan *execute* atau kadang disebut sebagai *start* atau *run*. Saat menggunakan fitur itu, maka komputer secara otomatis akan menerjemahkan *coding* dan *programming language* menjadi bahasa komputer sehingga setelah dibentuk menjadi program oleh komputer, komputer bisa mengeksekusi program yang telah dibuat.

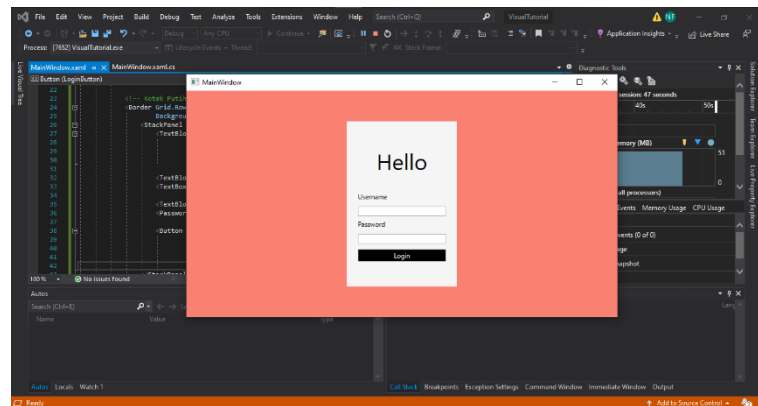


Gambar 2.50 Visual Studio Aksi Method C#

18. Dan akhirnya disini akan diperlihatkan hasil dari yang telah dikerjakan. Saat pengguna menekan tombol maka akan terjadi suatu perubahan.



Gambar 2.51 Visual Studio Debugging dan Execute



Gambar 2.52 Visual Studio Hasil



*Code Snippet* ( Klik dua kali untuk melihat keseluruhan *code* ) :

```
<!-- MainWindow.xaml -->

<Window x:Class="VisualTutorial.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
    xmlns:local="clr-namespace:VisualTutorial"
    mc:Ignorable="d"
    Title="MainWindow" Height="450" Width="800">
    <Grid Background="Salmon">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*" />
            <ColumnDefinition Width="200" />
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>

        <Grid Grid.Column="1">
            <Grid.RowDefinitions>
                <RowDefinition Height="*" />
                <RowDefinition Height="300" />
                <RowDefinition Height="*" />
            </Grid.RowDefinitions>

            <!-- Kotak Putih -->
            <Border Grid.Row="1" Padding="20"
                Background="WhiteSmoke">
                <StackPanel VerticalAlignment="Center">
                    <TextBlock x:Name="JudulForm"
```

```
// MainWindow.xaml.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

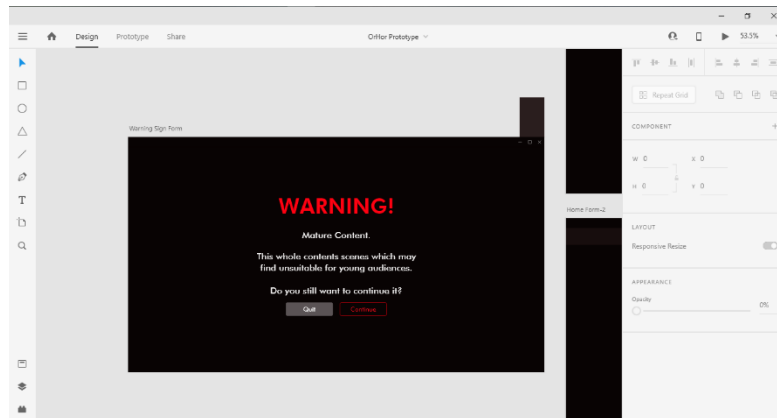
namespace VisualTutorial
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();

            private void LoginButton_Click(object sender,
RoutedEventArgs e)
```

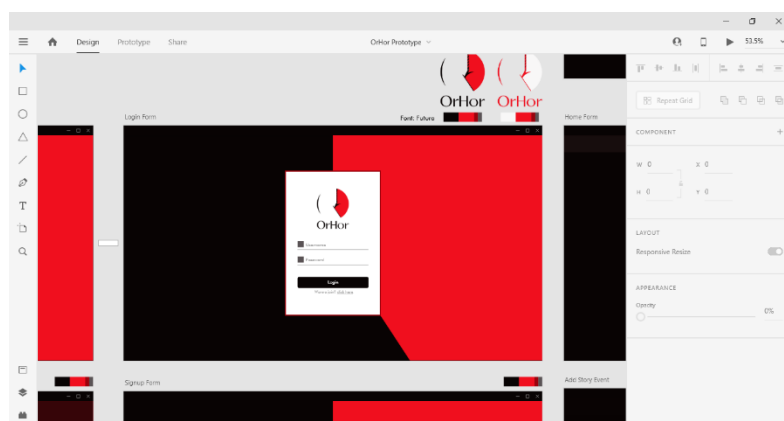
### 2.3.3 HASIL

Sebelum menunjukkan hasilnya, yang ditampilkan hanya sebagian dari yang seharusnya karena *project* masih dalam proses 30%.

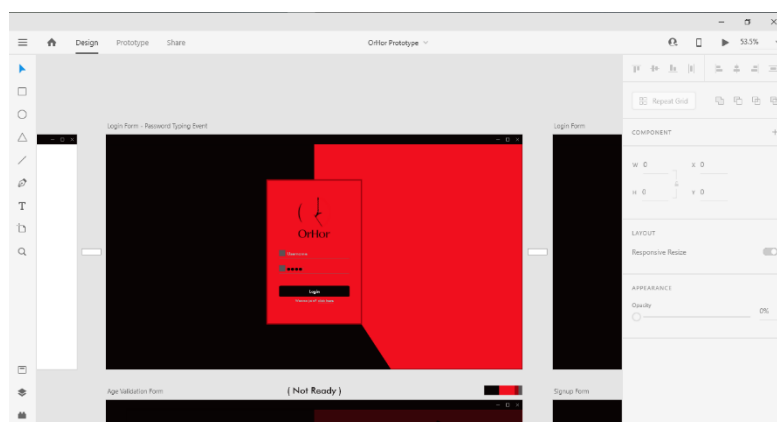
#### a. PROTOTYPE



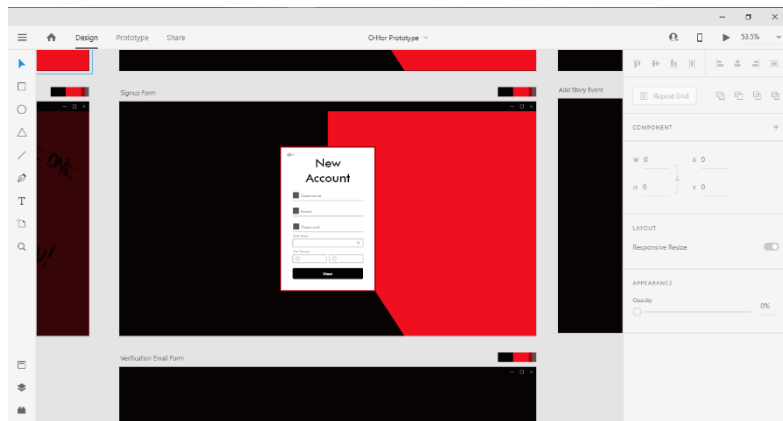
Gambar 2.53 Halaman Peringatan



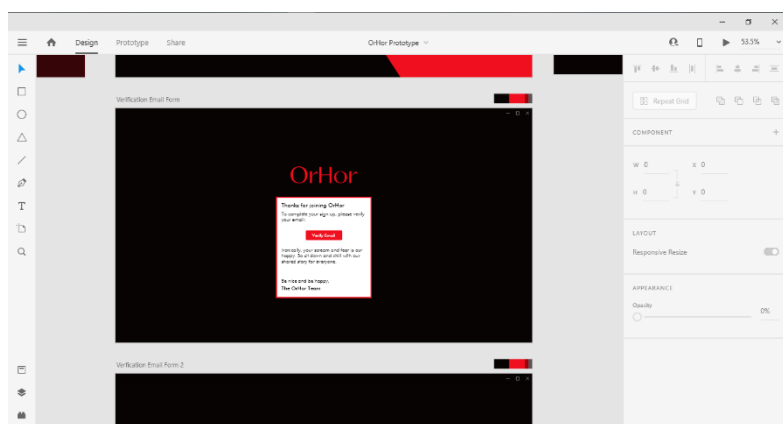
Gambar 2.54 Halaman Login



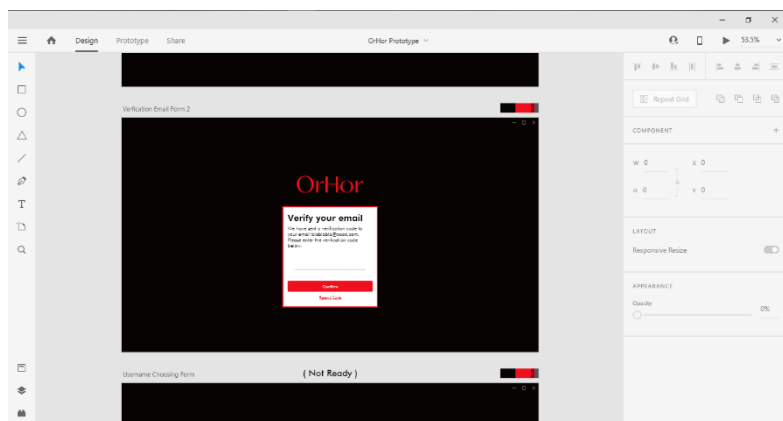
Gambar 2.55 Halaman Login (Password)



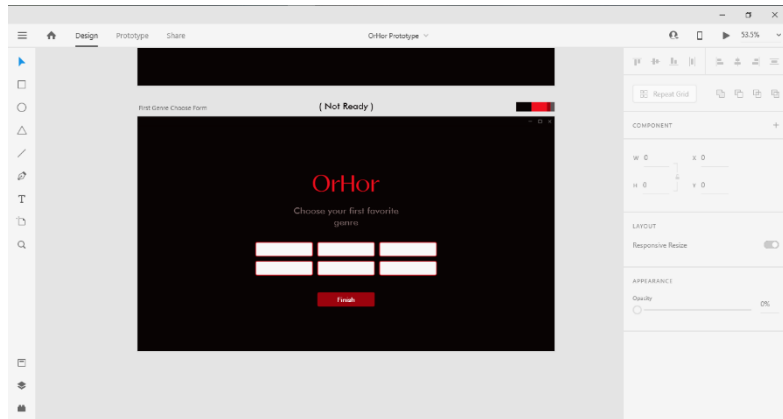
Gambar 2.56 Halaman Register



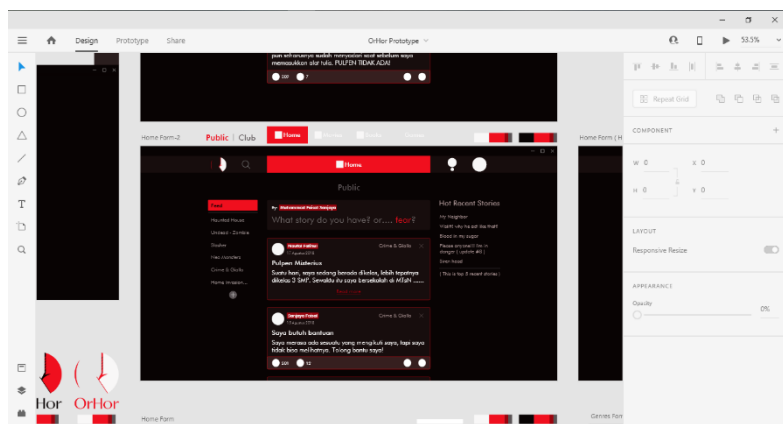
Gambar 2.57 Halaman Verifikasi Email (1)



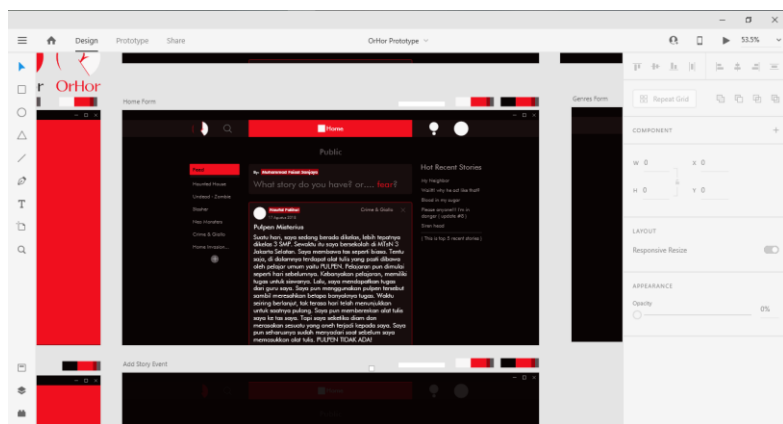
Gambar 2.58 Halaman Verifikasi Email (2)



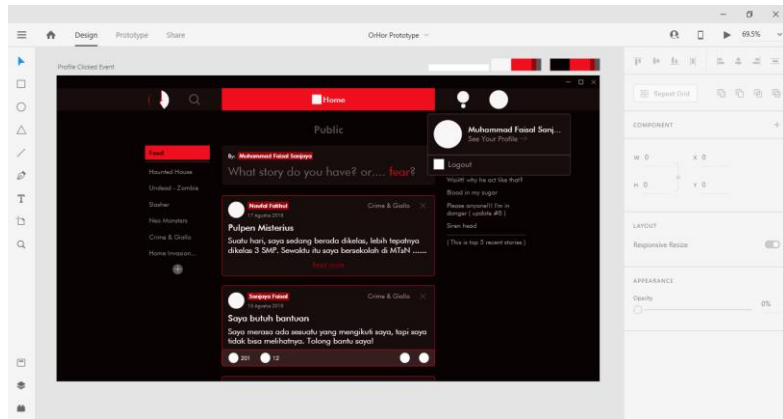
Gambar 2.59 Halaman Pilih Genre



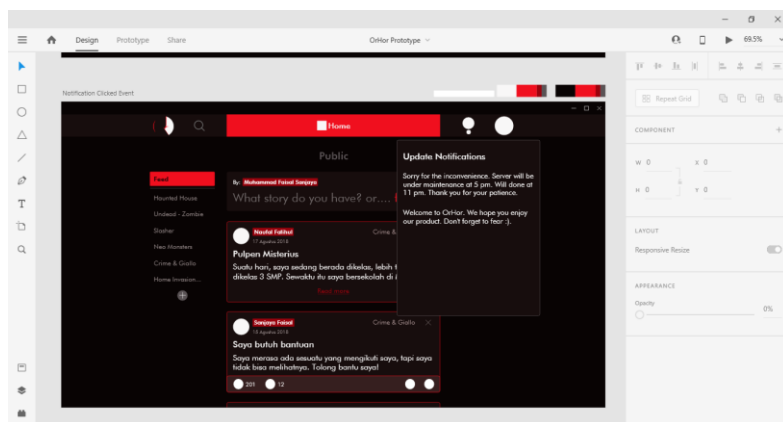
Gambar 2.60 Halaman Home



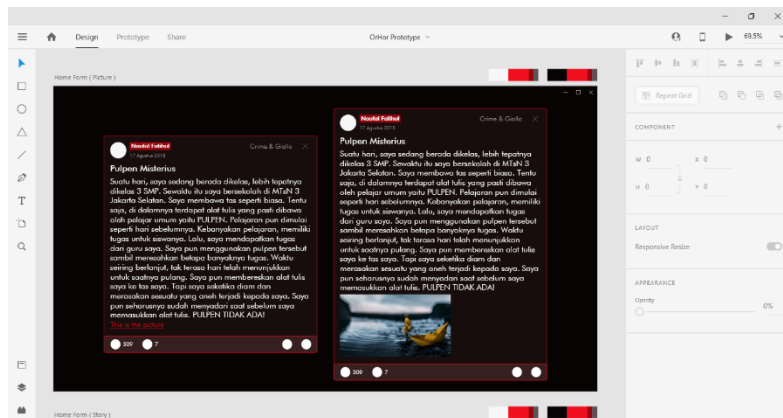
Gambar 2.61 Halaman Home (Meluaskan Cerita)



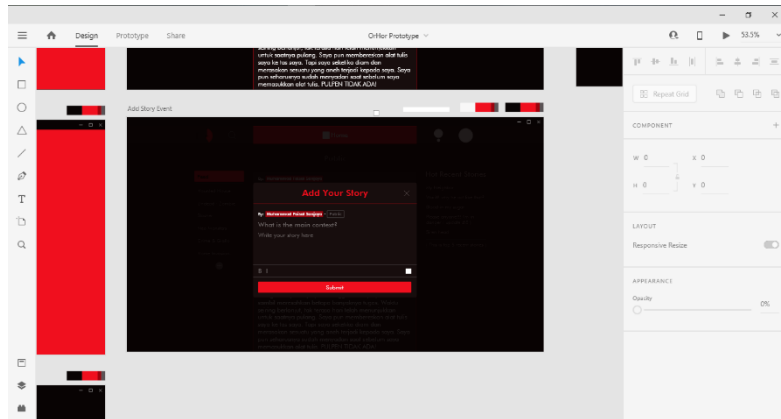
Gambar 2.62 Halaman Home (Profil Popup)



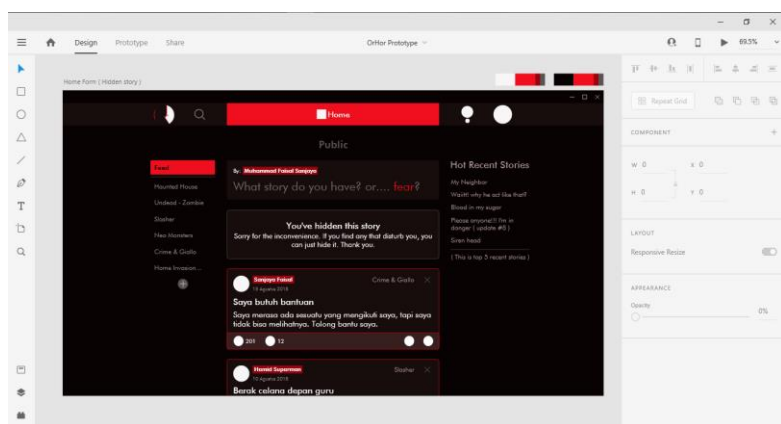
Gambar 2.63 Halaman Home (Notifikasi Popup)



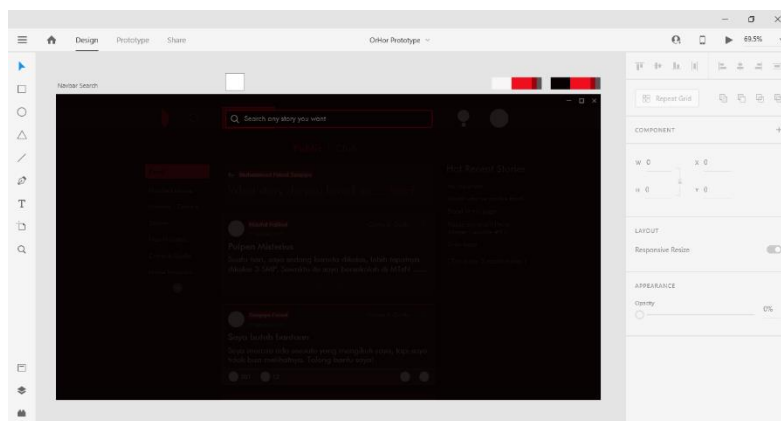
Gambar 2.64 Kerangka Cerita



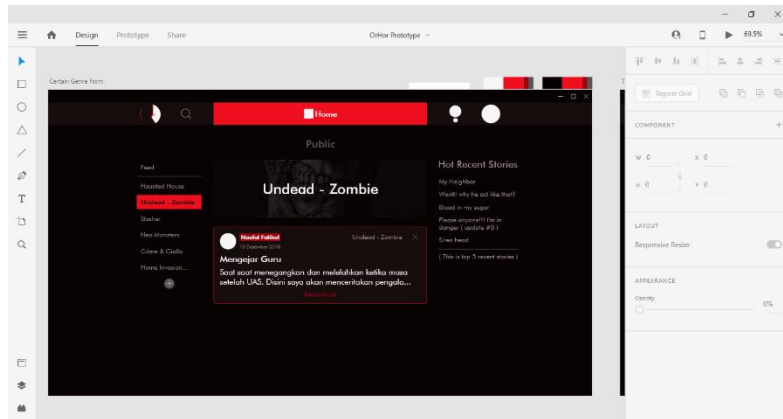
Gambar 2.65 Halaman Home (Tambah Cerita)



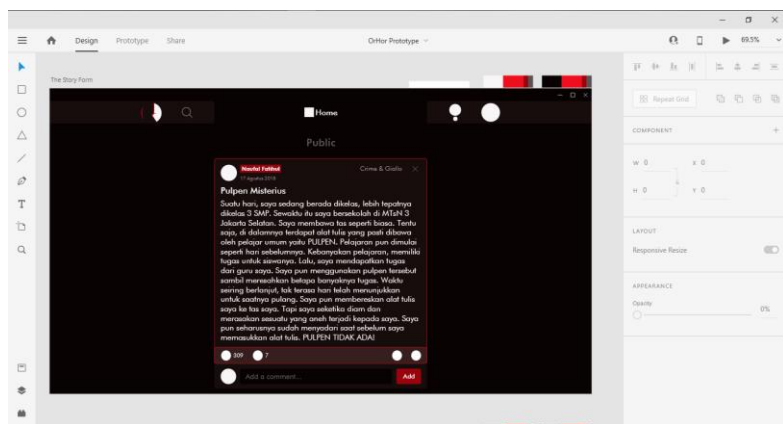
Gambar 2.66 Halaman Home (Sembunyikan Cerita)



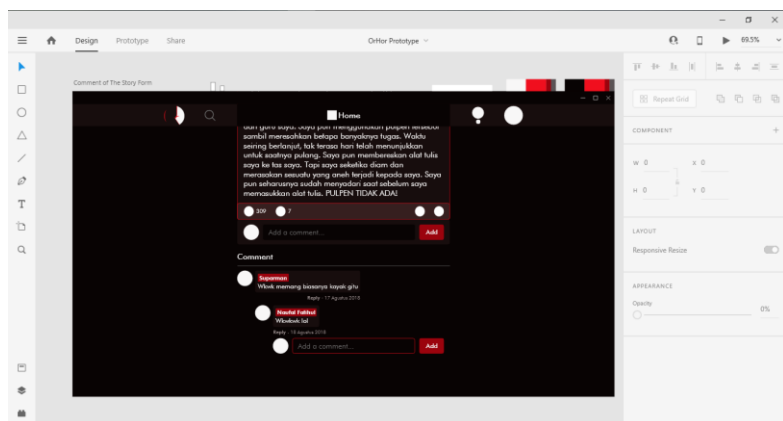
Gambar 2.67 Halaman Home (Cari Popup)



Gambar 2.68 Halaman Home (Pilih Genre)

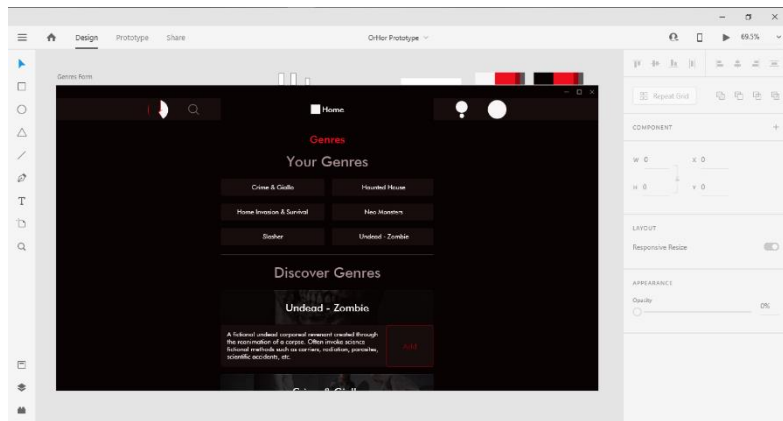


Gambar 2.69 Halaman Cerita

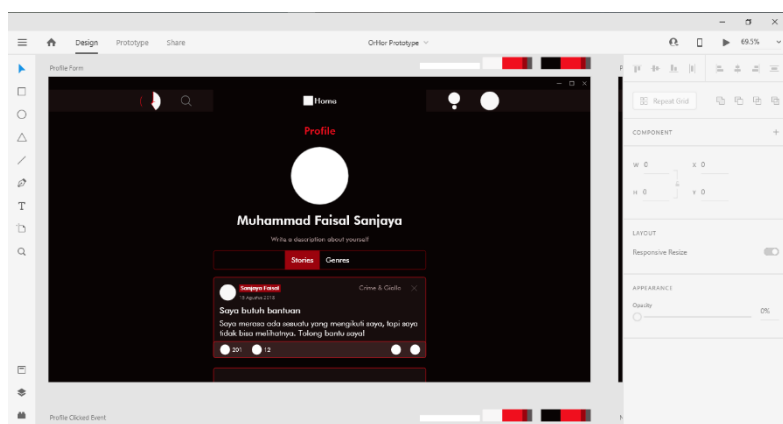


Gambar 2.70 Halaman Cerita (Komentar)

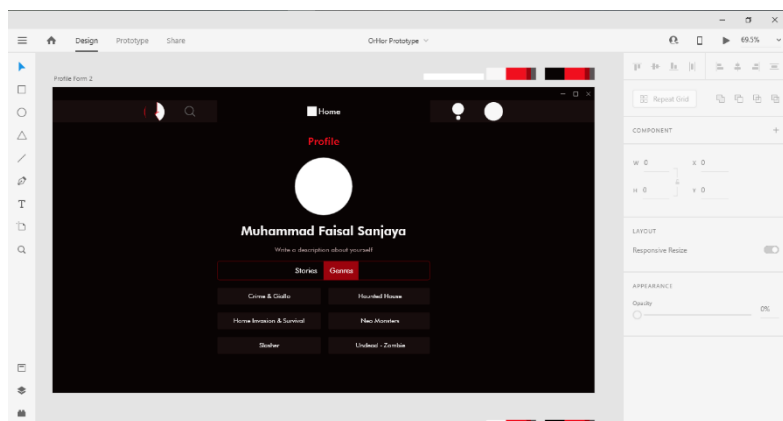




Gambar 2.71 Halaman Pilih Genre

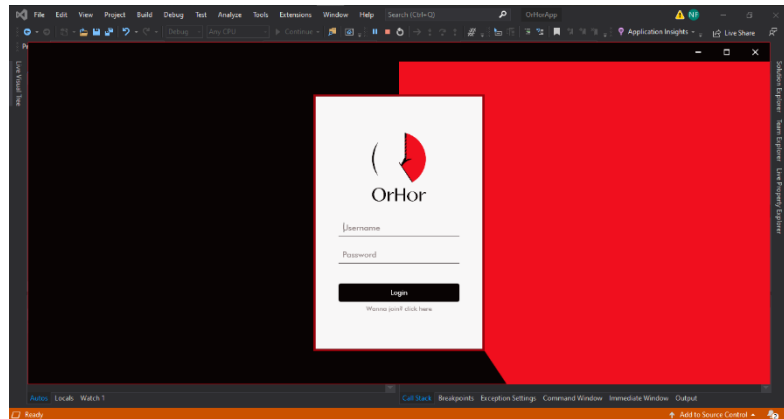


Gambar 2.72 Halaman Profil (Cerita)

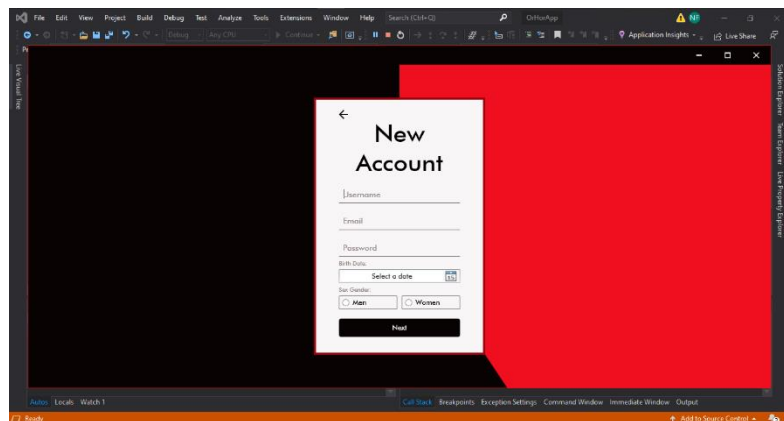


Gambar 2.73 Halaman Profil (Genre)

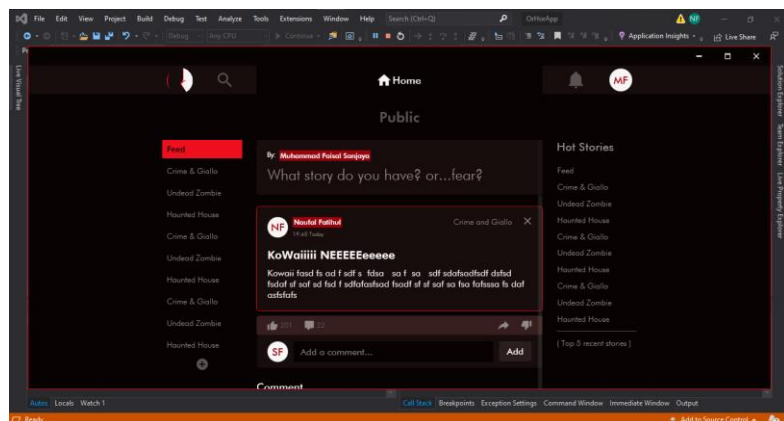
## b. APPLICATION



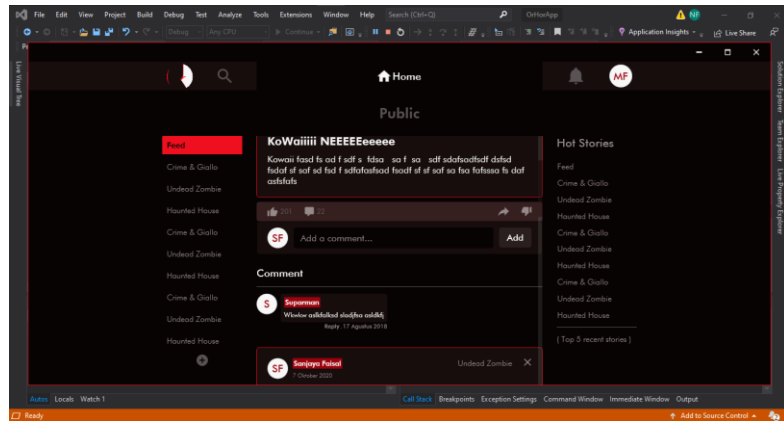
Gambar 2.74 Halaman Login Aplikasi



Gambar 2.75 Halaman Register Aplikasi



Gambar 2.76 Halaman Home Aplikasi (1)



Gambar 2.77 Halaman Home Aplikasi (2)