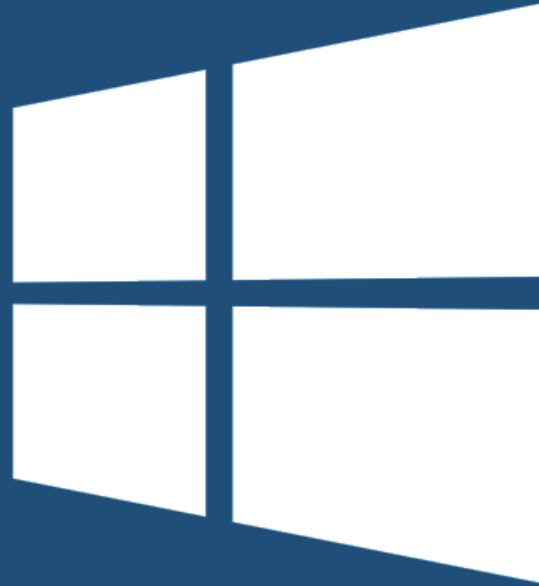




BELAJAR MEMBUAT APP WINDOWS PHONE 8.1

I Putu Yoga Permana

Microsoft Student Partner

DAFTAR ISI

Permulaan.....	3
Spesifikasi.....	3
Langkah Instalasi.....	3
Membuat Project.....	8
Menambah Referensi & Assets.....	9
Membuat Model.....	10
Rancang Halaman.....	12
Mengambil Data dari Berkas.....	14
Berpindah Halaman.....	18
Menu AppBar.....	22
Publish ke Store.....	24

Permulaan

Pada modul ini kita akan mencoba bagaimana membuat sebuah app sederhana tentang Puteri Indonesia. Nantinya terdapat daftar Puteri Indonesia beserta foto dan biografinya.

App yang kita coba buat ini akan berbasis WinRT bukan Silverlight, tentunya ada beberapa perbedaan dengan yang versi Silverlight seperti perpindahan halaman, Application Bar dll.

Sebelum memulai membuat app, **pastikan sudah mengunduh berkas** yang dibutuhkan seperti data json dan foto-foto yang akan dipakai. Semuanya bisa di unduh di http://winpoin.com/?attachment_id=29414.

Pastikan kamu punya installer Visual Studio 2013 Community Update 4, karena di dalamnya sudah tersedia Windows Phone SDK 8 beserta emulator image dari Windows Phone 8.1.

Berkas Installer atau ISO Visual Studio Community 2013 dapat kamu temukan di [halaman ini](#). Untuk ISO sendiri ukurannya sangat besar yaitu **7GB** jadi pastikan koneksi internetmu cukup memadai.

Catatan: Jika pilihan emulator Windows Phone 8.1 tidak ada, kamu bisa mengunduh iso SDK 8.1 tambahan [disini](#).

Spesifikasi

Namun sebelum melanjutkan topik ini, kamu perlu memastikan komputermu memenuhi kriteria spesifikasi sebagai berikut yang nantinya digunakan untuk menjalankan emulator Windows Phone 8.1 :

- RAM 4GB atau lebih (untuk emulator)
- Ruang kosong di hardisk 15GB atau lebih
- BIOS harus mendukung fitur : Hardware-assisted virtualization, Second Level Address Translation (SLAT), Hardware-based Data Execution Prevention (DEP).
- Sistem Operasi: Windows 8.1 Pro 64 bit atau lebih tinggi.

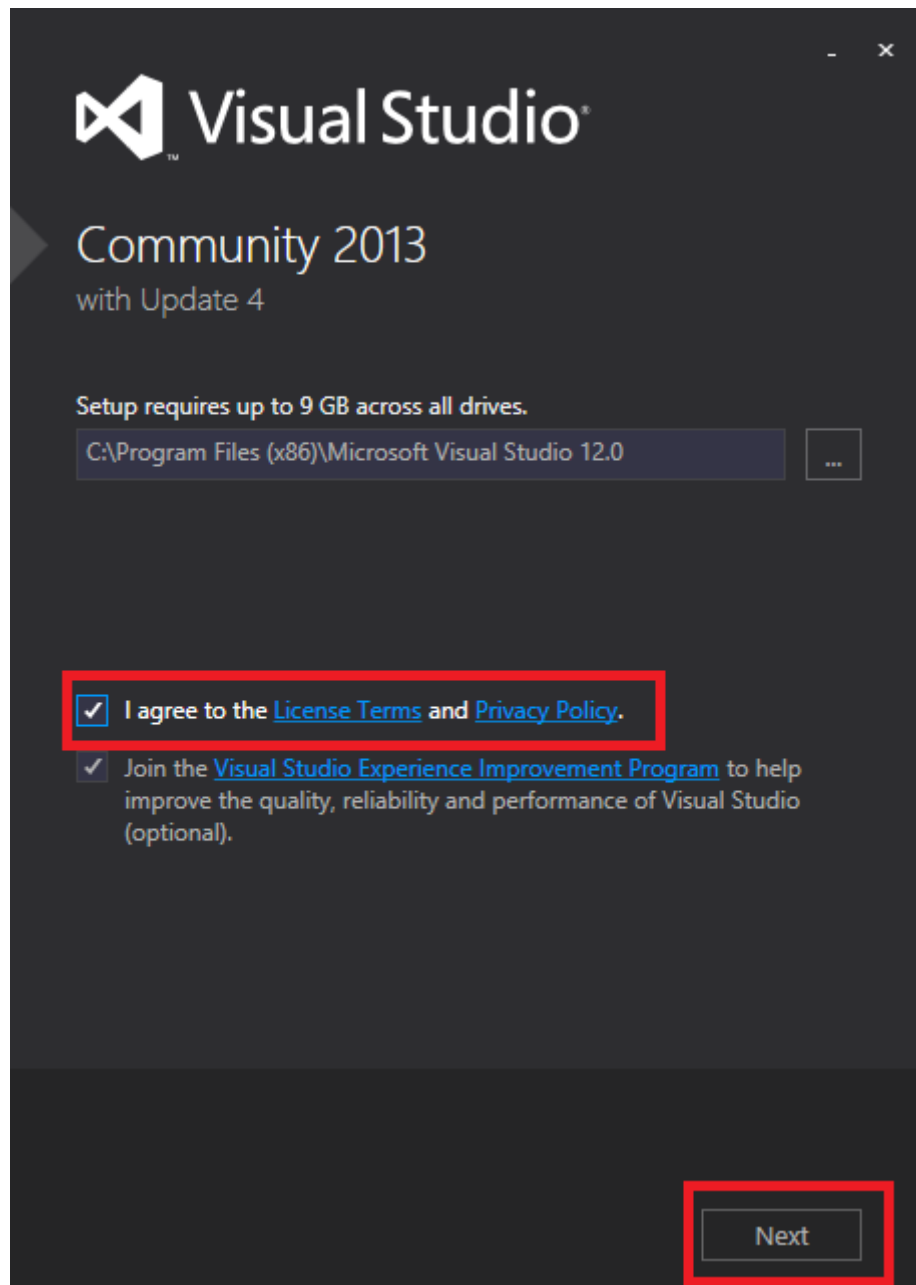
Ingat: Kamu perlu mengaktifkan Hyper-v di BIOS sebelum kamu bisa menjalankan emulator.

Langkah Instalasi

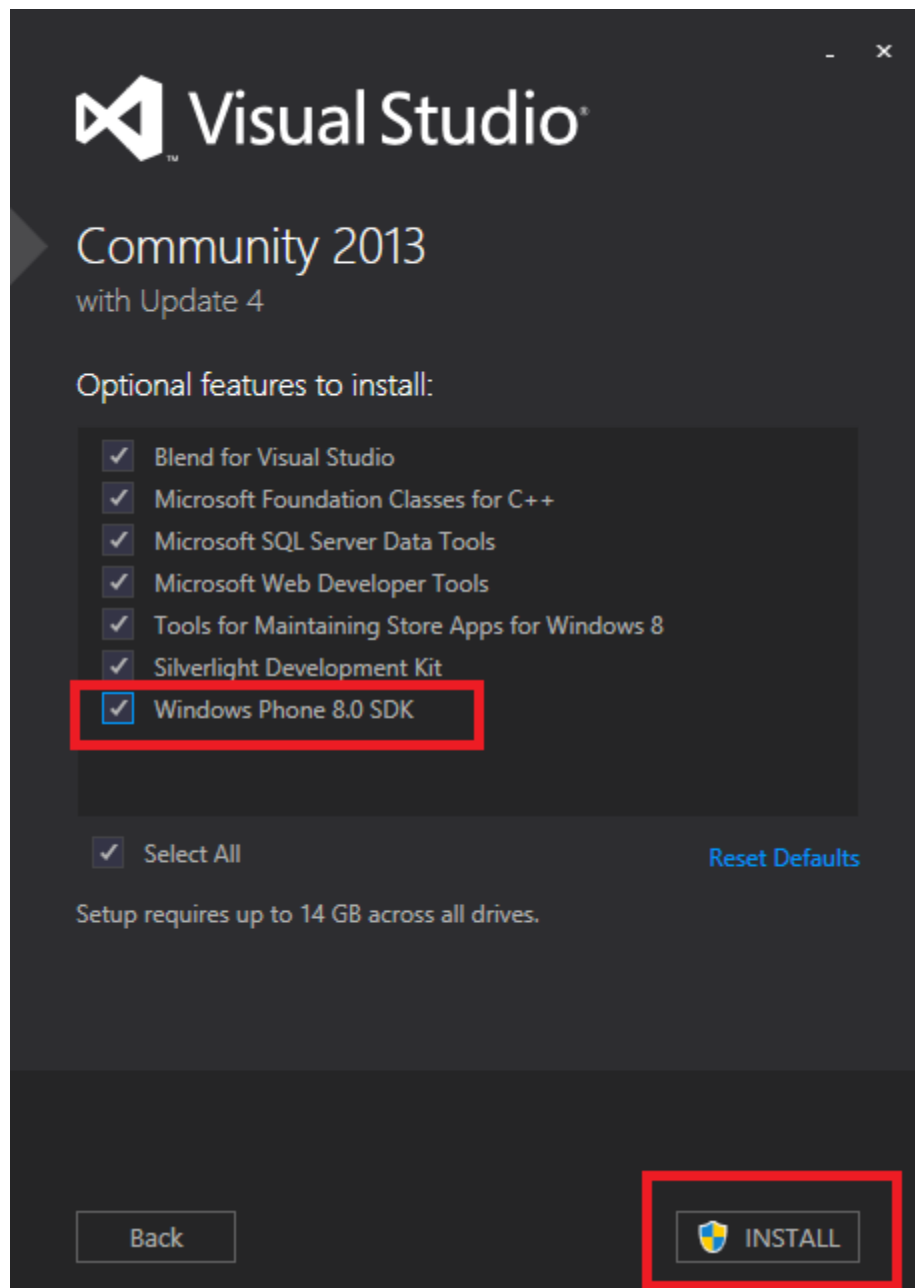
Pada bagian ini akan berfokus pada bagaimana caranya kamu menginstall dan mempersiapkan lingkungan pengembangan dalam komputermu. Dengan asumsi

kamu telah memiliki ISO Visual Studio 2013 Community Edition, berikut langkah-langkah instalasinya:

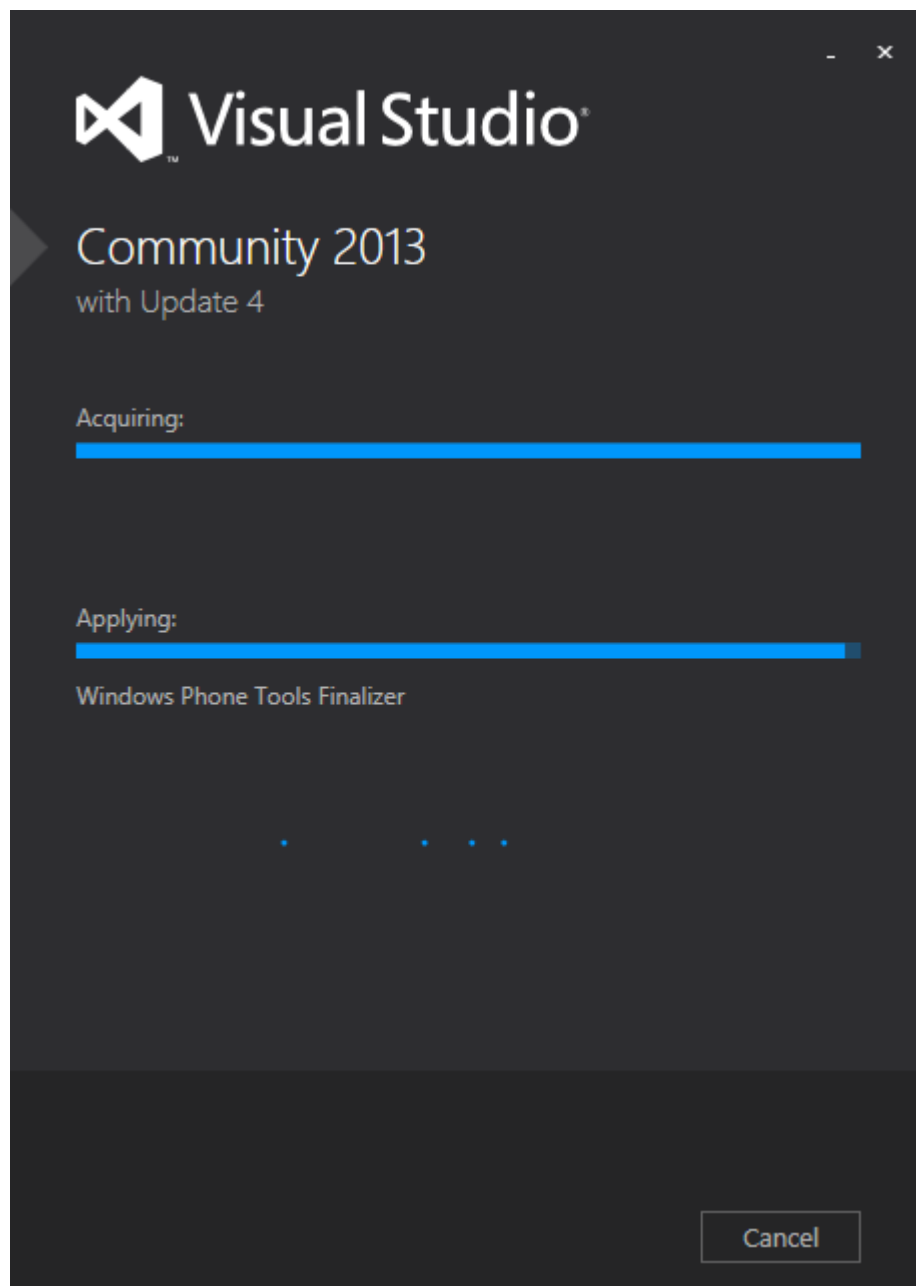
1. Mount berkas ISO Visual Studio 2013 Community Edition, kemudian jalankan berkas vs_community.exe yang ada di dalam DVD Drive. **Centang checkbox** “I agree to the License Terms and Privacy Policy”, kemudian klik tombol **Next**.



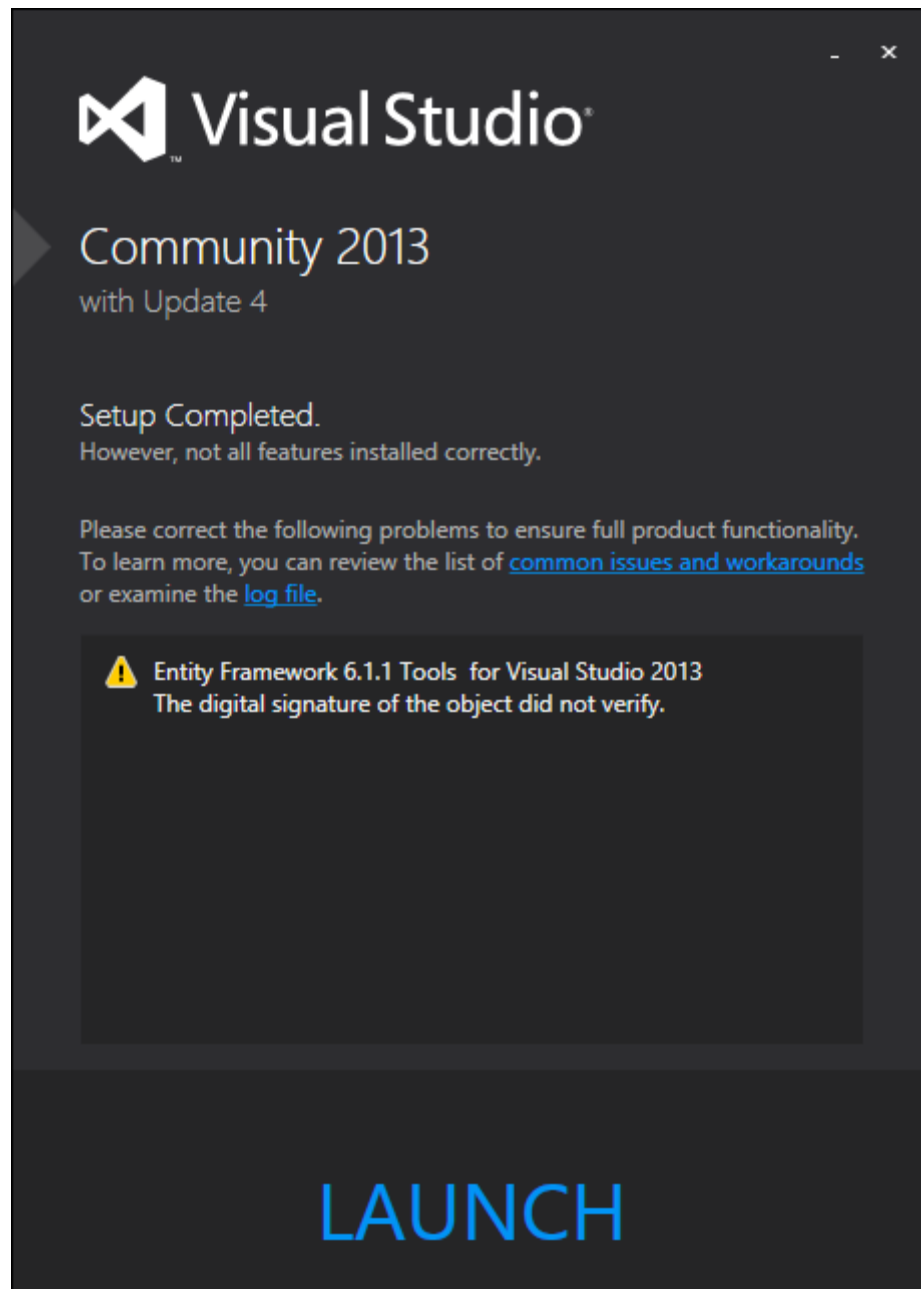
2. Selanjutnya kamu akan dihadapkan pilihan fitur yang akan diinstall, disini saya **mencentang semua fitur** yang ada termasuk di dalamnya Windows Phone 8.0 SDK. Kemudian klik tombol **install**.



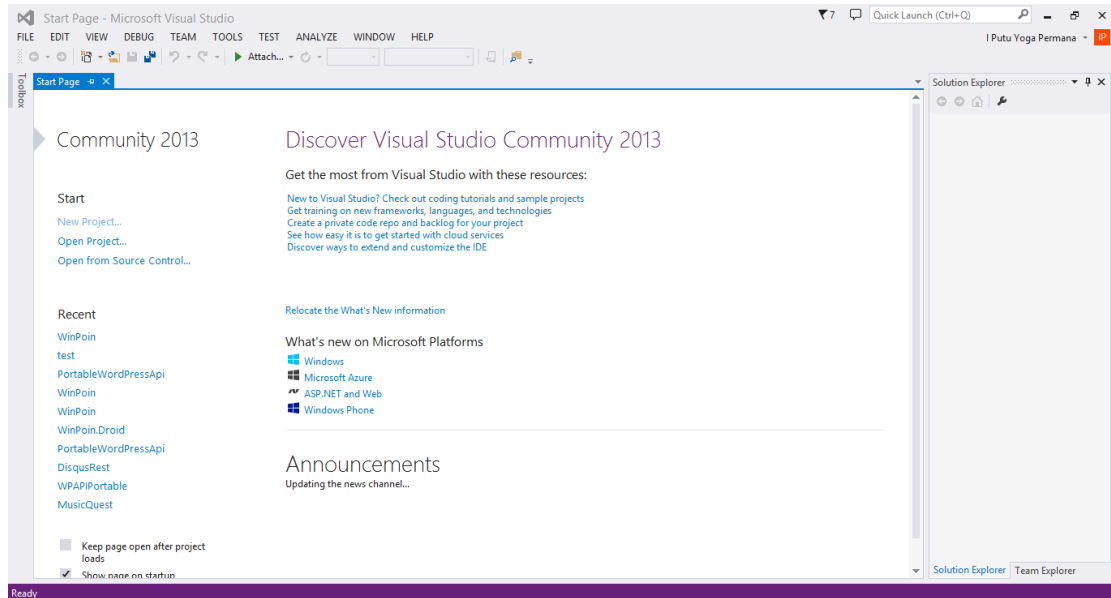
3. Sampai disini akan membutuhkan waktu yang sangat lama, terhitung waktu yang saya butuhkan untuk instalasi ini sekitar 1-2 jam.



4. Selanjutnya akan muncul *window setup completed*. Klik tombol **Launch** untuk membuka Visual Studio 2013 Community Edition.

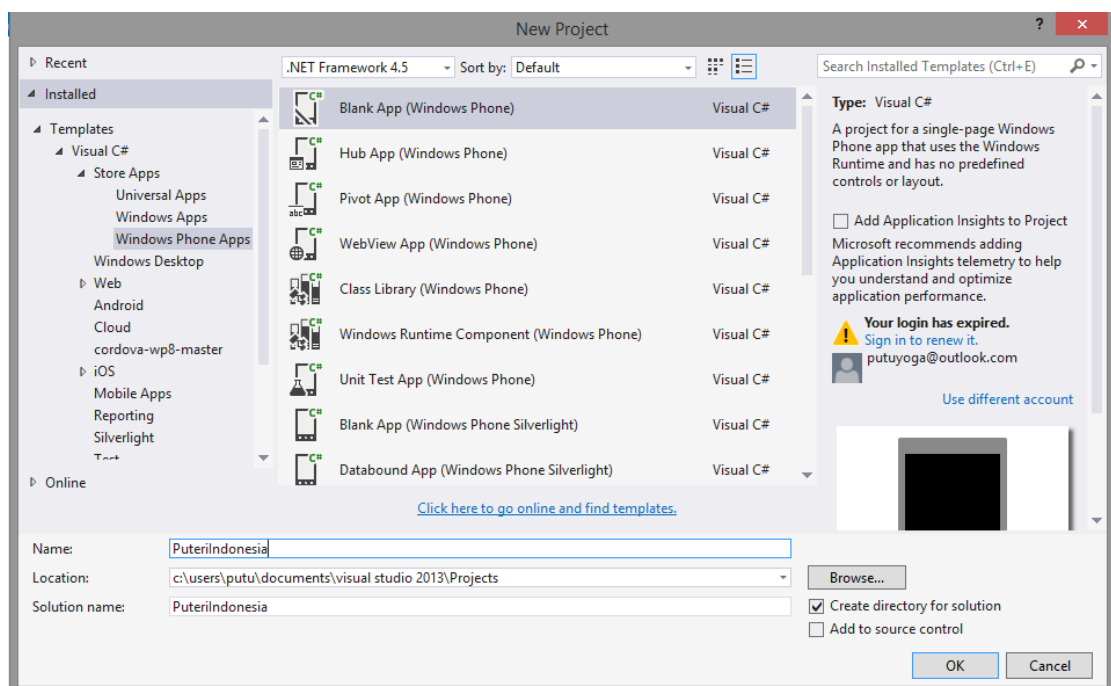


5. Selamat ! sekarang kamu sudah satu langkah lebih dekat untuk mempunyai app windows phone 8.1 buatanmu sendiri.



Membuat Project

Jika semua sudah ada, sekarang langsung dimulai saja. Pertama-tama buat new project di Visual Studio. Caranya dari menu File → New → New Project atau dengan shortcut Ctrl + Shift + N.



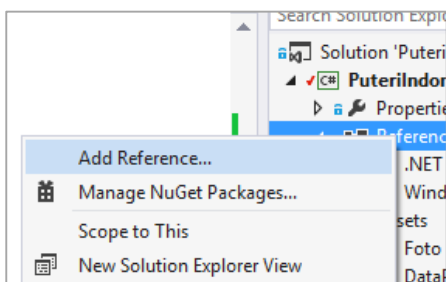
Pilih dari Templates → Store Apps → Windows Phone Apps → Blank App (Windows Phone), beri nama PuteriIndonesia kemudian klik tombol OK.

Menambah Referensi & Assets

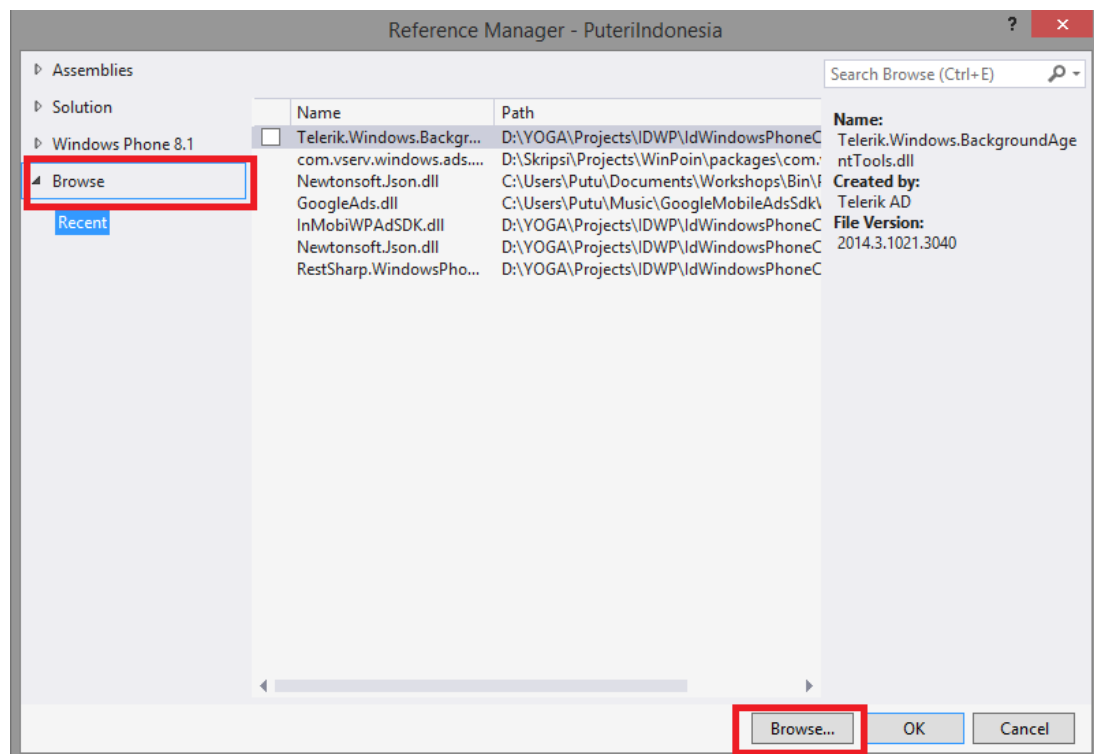
Jika sudah mengunduh berkas yang akan diperlukan, sekarang kita perlu menambahkannya di project ini.

Pada project ini, pustaka yang akan digunakan yaitu JSON.Net untuk Deserialize string biasa menjadi objek yang bisa kita gunakan.

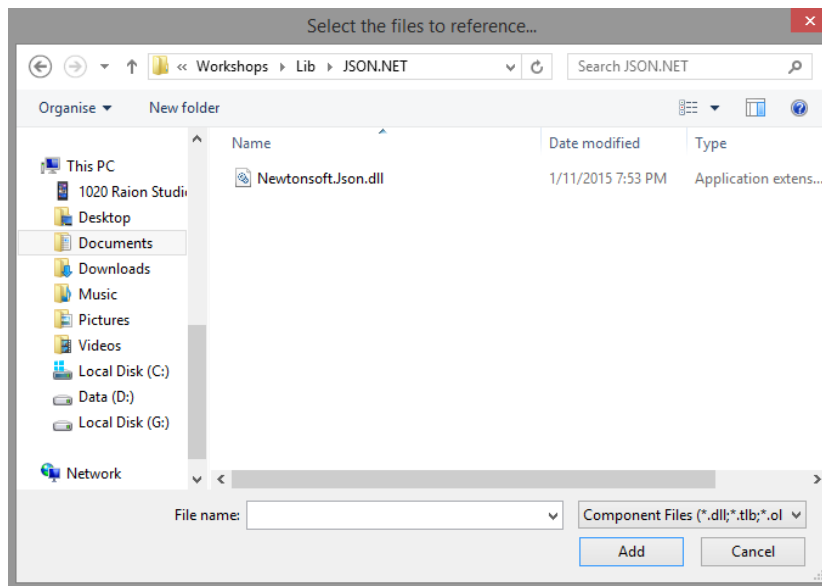
Untuk cara menambahkan referensi pustaka ke project ini, klik kanan Reference → Add Reference.



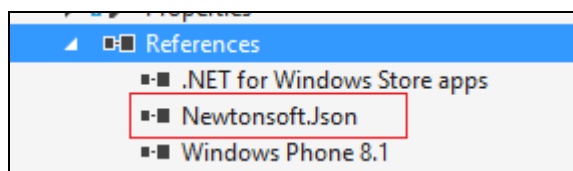
Selanjutnya pilih tombol Browse.



Tentukan lokasi pustaka dll, kemudian pilih tombol Add.



Jika sudah Klik OK, lihat kembali daftar References jika berhasil menambahkan pustaka akan muncul nama “Newtonsoft.Json”.

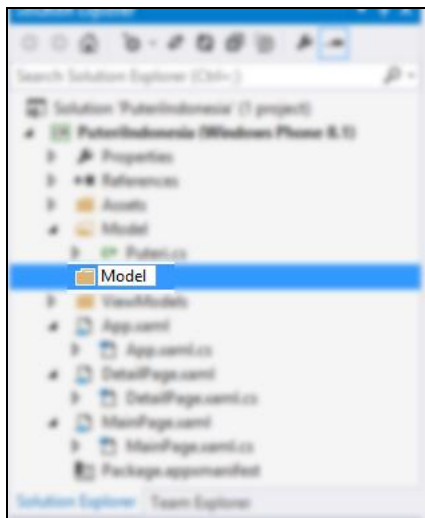


Selain itu juga ada foto para puteri Indonesia dan berkas json data list sang Puteri yang harus ditambahkan ke folder Assets.

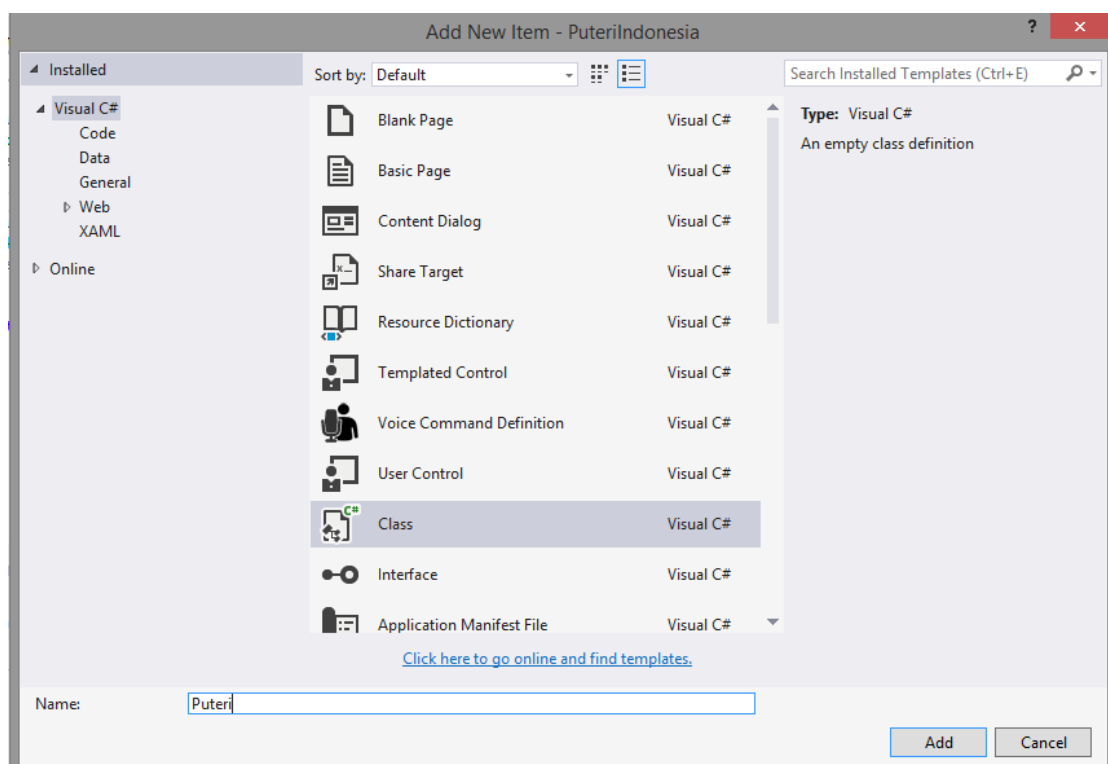
Caranya sederhana, cukup drag berkas-berkas tersebut ke folder Assets via Solution Explorer yang ada di Visual Studio. Terakhir jangan lupa pada berkas DataPuteri.json set **Build Action-nya ke Content** dengan cara klik kanan berkas tersebut dan pilih Properties.

Membuat Model

Sekarang kita harus membuat class model **Puteri** untuk menampung data Puteri. Klik kanan Project di Solution Explorer pilih menu Add kemudian Add New Folder.



Selanjutnya klik kanan di folder Model, pilih menu Add lalu Class. Beri nama Puteri kemudian klik tombol Add.



Buka berkas Puteri.cs, lalu ganti class Puteri dengan kode berikut mulai dari baris nomor 9:

Model/Puteri.cs

```

9  public class Puteri
10 {
11     public int Id { get; set; }
12     public string Nama { get; set; }
13     public int Tahun { get; set; }
14     public string Asal { get; set; }
15     public string Biografi { get; set; }
16 }

```

Rancang Halaman

Dalam MainPage.xaml, buat 2 definisi baris dalam elemen Grid, seperti kode berikut yang ada di bawah:

MainPage.xaml

```

11 <Grid>
12     <Grid.RowDefinitions>
13         <RowDefinition Height="Auto" />
14         <RowDefinition Height="*" />
15     </Grid.RowDefinitions>
16 </Grid>

```

Sekarang kita mempunyai dua definisi baris yang pertama dengan tinggi Auto untuk judul, dan kedua * akan memenuhi sisa ruang untuk daftar puteri Indonesia nantinya.

Sekarang tambahkan elemen `TextBlock` beserta propertinya setelah tag penutup `Grid.RowDefinitions`:

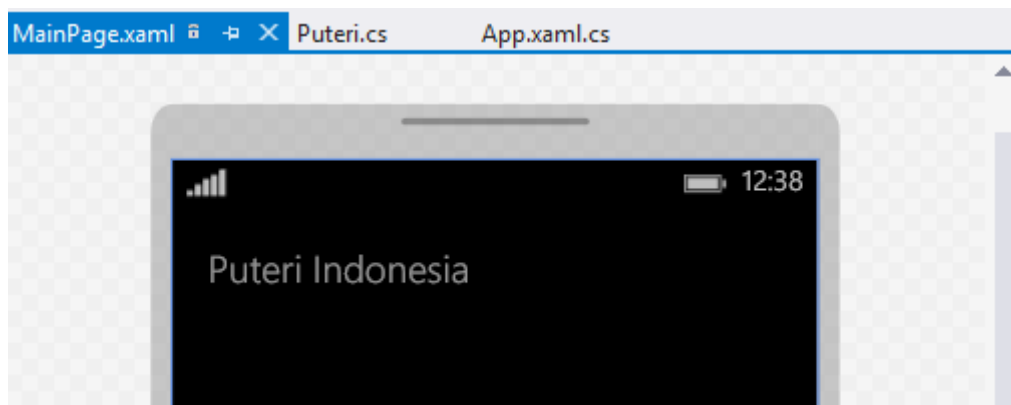
MainPage.xaml

```

11 <Grid>
12     <Grid.RowDefinitions>
13         <RowDefinition Height="Auto" />
14         <RowDefinition Height="*" />
15     </Grid.RowDefinitions>
16     <TextBlock Text="Puteri Indonesia" Grid.Row="0" Style="{StaticResource
GroupHeaderTextBlockStyle}" Margin="24" />
17 </Grid>

```

Pada kode diatas kita mengatur properti marginnya sebesar 24 (horizontal dan vertikal) dan style menggunakan resource statis `GroupHeaderTextBlockStyle`. Nantinya di window XAML Designer akan nampak seperti tampilan berikut:



Selanjutnya kita perlu menambahkan **ListView** beserta template itemnya dibawah elemen **TextBlock** judul, dengan kode berikut:

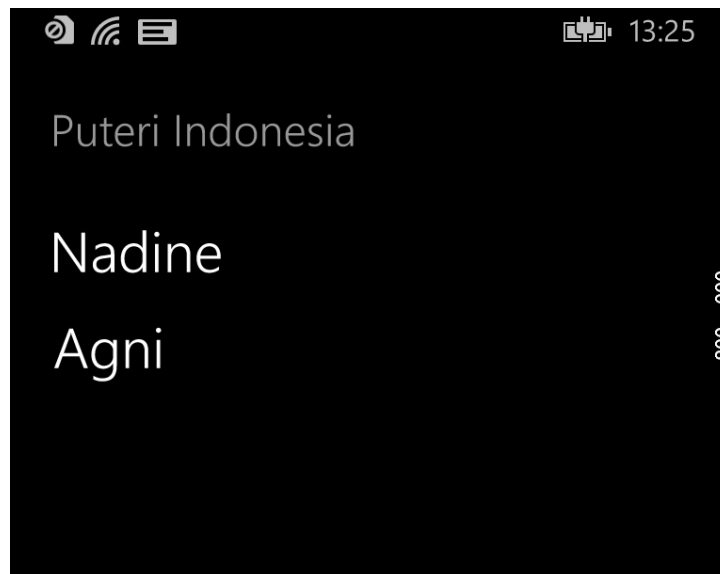
MainPage.xaml	
17	<ListView x:Name="ListPuteri" Grid.Row="1">
18	<ListView.ItemTemplate>
19	<DataTemplate>
20	<TextBlock Text="{Binding Nama}" Margin="24,6" Style="{StaticResource
	ListViewItemTextBlockStyle}" />
21	</DataTemplate>
22	</ListView.ItemTemplate>
23	</ListView>

Maksud dari kode diatas kita menambahkan elemen **ListView** ke dalam **Grid** yang diletakkan di baris ke 2 (indeks 1).

Kemudian buka berkas MainPage.xaml.cs, kita coba menampilkan data ke dalam ListView yang telah kita buat sebelumnya. Tambahkan kode berikut ke dalam konstruktor kelas MainPage.

MainPage.xaml.cs	
32	List<Puteri> listItem = new List<Puteri>();
33	listItem.Add(new Puteri())
34	{
35	Nama = "Nadine"
36	});
37	listItem.Add(new Puteri())
38	{
39	Nama = "Agni"
40	});
41	ListPuteri.ItemsSource = listItem;

Kemudian coba jalankan project di emulator atau device, nantinya akan terlihat tampilan seperti dibawah ini :



Jika tampilan sudah sama dengan yang diatas, berarti kamu berhasil menampilkan data yang ada di objek `listItem` ke dalam elemen `ListView`.

Mengambil Data dari Berkas

Oke, selanjutnya kita akan membuat berkas kelas baru bernama `MainViewModel` yang berisi List data Puteri Indonesia beserta method untuk memuat data dari berkas JSON.

Pertama-tama buat folder `ViewModels` di dalam project. Kemudian tambahkan `MainViewModel.cs` ke dalamnya. Selanjutnya masukan kode berikut kedalam berkas `.cs` tersebut :

```

ViewModels/MainViewModel.cs

13 public class MainViewModel : INotifyPropertyChanged
14 {
15     private List<Puteri> _listPuteri;
16     public List<Puteri> ListPuteri
17     {
18         get { return _listPuteri; }
19         set
20         {
21             if(_listPuteri != value)
22             {
23                 _listPuteri = value;
24                 NotifyPropertyChanged("ListPuteri");
25             }
26         }
27     }
28
29     public async void GetData()
30     {
31         Uri dataUri = new Uri("ms-appx:///Assets/DataPuteri.json");
32         StorageFile file = await StorageFile.GetFileFromApplicationUriAsync(dataUri);
33         string jsonText = await FileIO.ReadTextAsync(file);
34         ListPuteri = JsonConvert.DeserializeObject<List<Puteri>>(jsonText);

```

```

35     }
36
37
38     public event PropertyChangedEventHandler PropertyChanged;
39     public void NotifyPropertyChanged(string propertyName)
40     {
41         if(PropertyChanged != null)
42         {
43             PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
44         }
45     }
46 }

```

Pada kelas MainViewModel di atas, mengimplementasikan interface **INotifyPropertyChanged** yaitu event **PropertyChanged**. Kemudian method **NotifyPropertyChanged** yang digunakan untuk memberitahu properti tertentu berubah.

Ada juga method **GetData** yang digunakan untuk mengambil string json dari berkas DataPuteri.json yang selanjutnya di deserialize dari string menjadi objek.

Kita perlu objek dari MainViewModel ini dapat diakses dimana saja, oleh karena itu kita menambahkan properti statis berikut ke App.xaml.cs :

App.xaml.cs

```

31 public static MainViewModel ViewModel { get; private set; }

```

Dan jangan lupa, instanisasi properti ViewModel di kontruktor kelas **App** seperti berikut:

App.xaml.cs

```

39 public App()
40 {
41     this.InitializeComponent();
42     this.Suspending += this.OnSuspending;
43
44     ViewModel = new MainViewModel();
45 }

```

Sekarang ViewModel bisa diakses dimana saja, kembali ke konstruktor MainPage.xaml.cs hapus kode **List** yang telah kita buat sebelumnya, dan tambahkan 2 baris kode seperti yang terlihat dibawah:

App.xaml.cs

```

27 public MainPage()
28 {
29     this.InitializeComponent();
30
31     this.NavigationCacheMode = NavigationCacheMode.Required;
32     this.DataContext = App.ViewModel;
33     App.ViewModel.GetData();
34 }

```

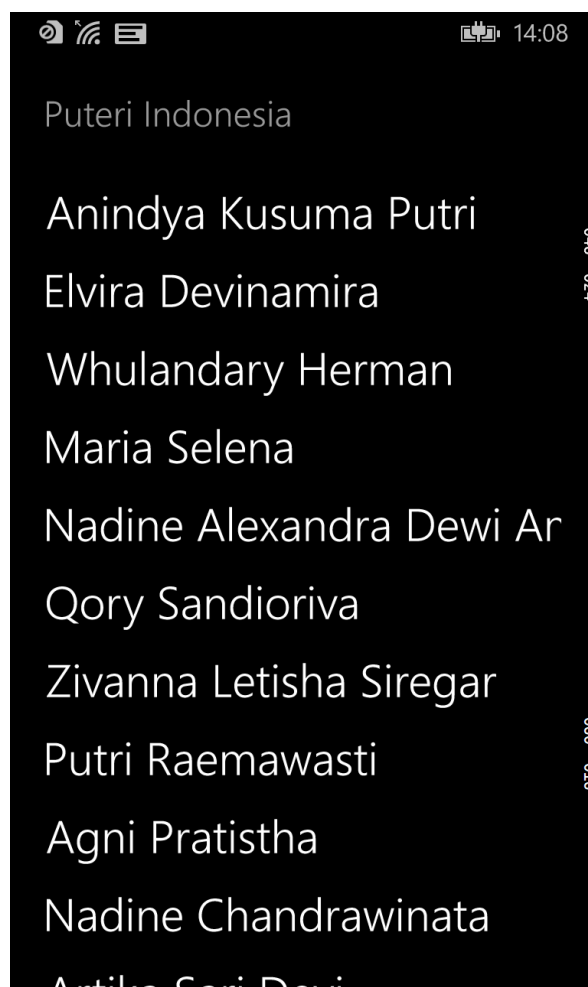
Pada kode di atas, kita mengatur data konteks dari `MainPage` untuk mengacu ke objek `ViewModel` tersebut.

Lalu method `GetData` dipanggil untuk mengambil data json dari berkas dan mengisi properti `ListPuteri` dari `ViewModel`.

Pada `MainPage.xaml`, selanjutnya kita perlu menambah binding kedalam properti `ItemsSource` pada elemen `ListView` seperti berikut :

MainPage.xaml	
17	<code><ListView x:Name="ListPuteri" Grid.Row="1" ItemsSource="{Binding ListPuteri}"></code>

Coba jalankan project dan hasilnya yang tampak akan seperti ini :



Karena pada WinRT tidak mendukung `StringFormat` di `Binding`, kita mempunyai opsi untuk menambahkan `Converters` atau menambahkan properti baru di kelas `Puteri`. Disini saya memilih opsi kedua, yaitu menambahkan properti Foto untuk kelas `Puteri`, berikut kodenya:

Model/Puteri.cs

```

16 public string Foto
17 {
18     get { return "/Assets/Foto/" + Id + ".jpg"; }
19 }

```

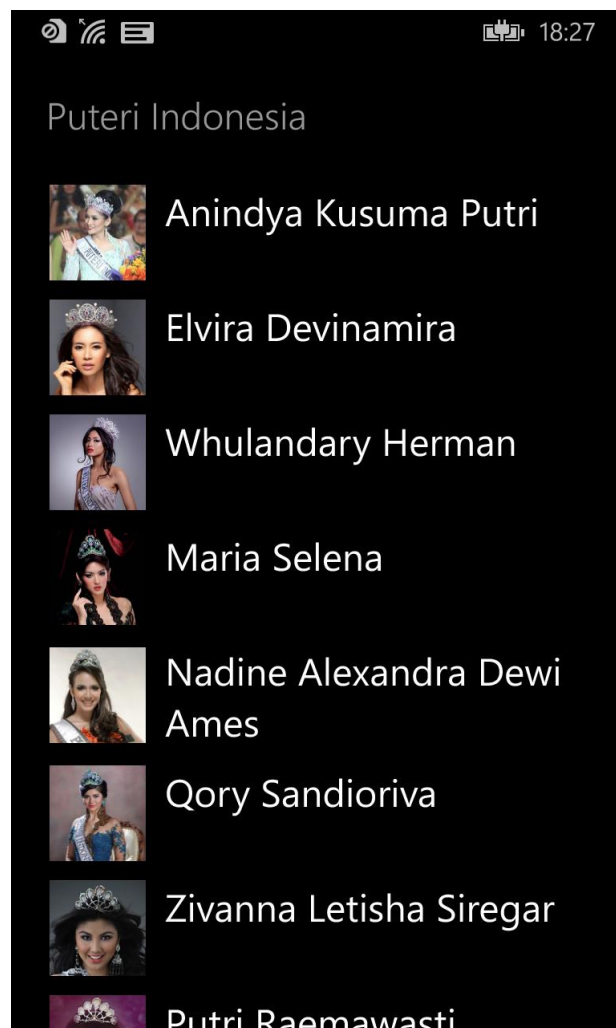
Pada tampilan diatas masih berupa teks tanpa gambar, sekarang untuk menambahkan gambar serta mengecilkan teks, kita perlu memodifikasi item template dari **ListView** :

MainPage.xaml

```

18 <ListView.ItemTemplate>
19     <DataTemplate>
20         <Grid Margin="24,6">
21             <Grid.ColumnDefinitions>
22                 <ColumnDefinition Width="60" />
23                 <ColumnDefinition Width="*" />
24             </Grid.ColumnDefinitions>
25             <Image Source="{Binding Foto}" Grid.Column="0" Height="60"
26 Stretch="UniformToFill"/>
27             <TextBlock Text="{Binding Nama}" Grid.Column="1" FontSize="24"
28 Margin="12,0,0,0" TextWrapping="WrapWholeWords" />
29         </Grid>
30     </DataTemplate>
31 </ListView.ItemTemplate>

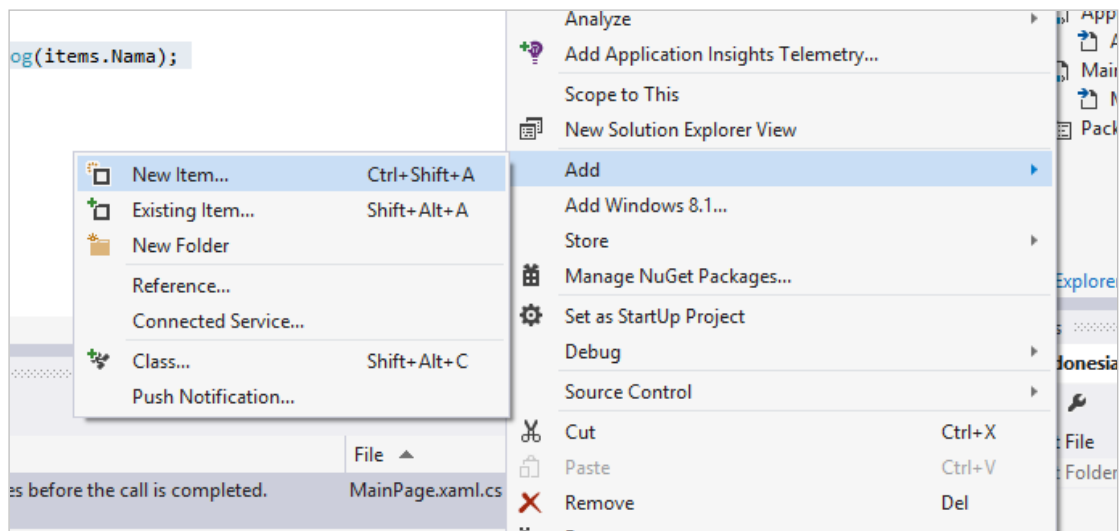
```



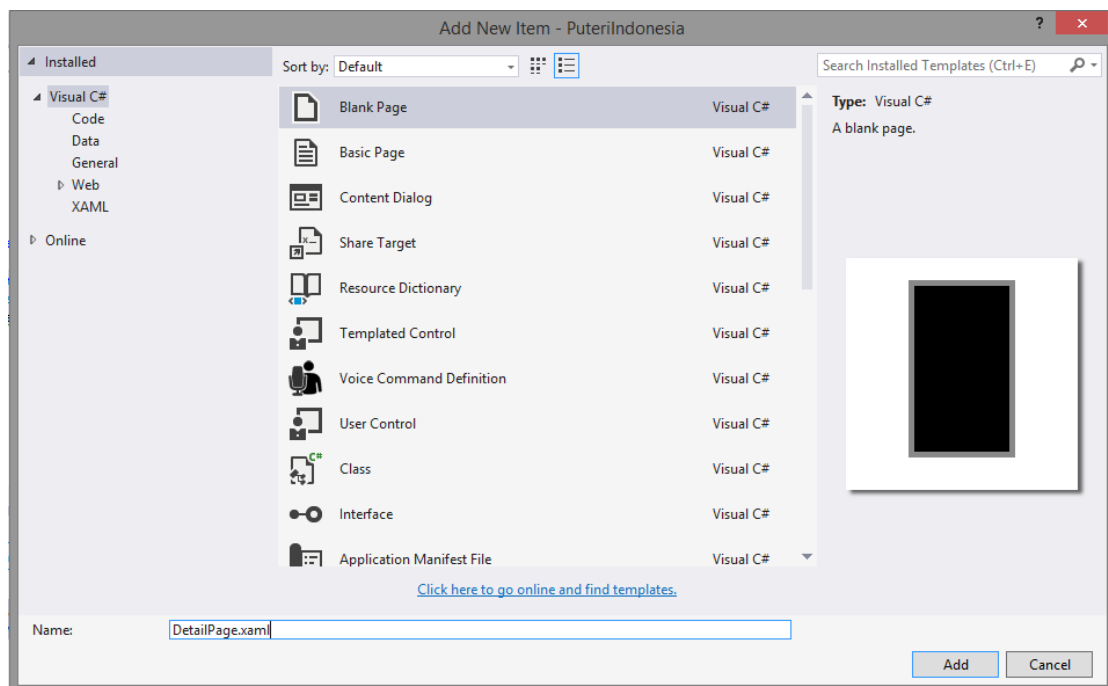
Berpindah Halaman

Sekarang kita akan membuat pengguna ketika memilih/tap item yang ada di list, akan berpindah ke halaman baru yang menampilkan informasi yang lebih lengkap.

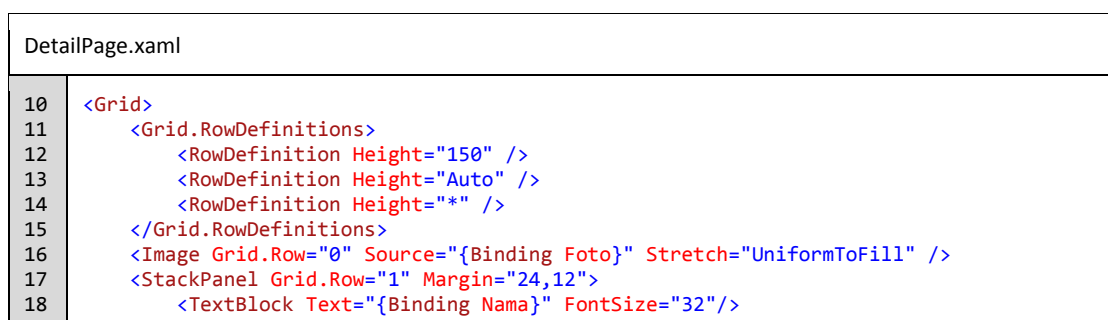
Sekarang kita membuat halaman baru bernama `DetailPage.xaml`, caranya klik kanan project kemudian pilih menu Add New Item.



Selanjutnya pilih Blank Page, beri nama **DetailPage**, dan Klik Tombol Add.



Halaman baru DetailPage.xaml akan ditambahkan ke root folder project. Buka DetailPage.xaml, dan tambahkan kode berikut :



```

19         <TextBlock Text="{Binding Asal}" FontSize="18" Foreground="Gray"/>
20         <TextBlock Text="{Binding Tahun}" FontSize="18" Foreground="Gray" />
21     </StackPanel>
22     <ScrollView Grid.Row="2" Margin="24,12">
23         <TextBlock Text="{Binding Biografi}" FontSize="20"
24         TextWrapping="WrapWholeWords"/>
25     </ScrollView>
26 </Grid>

```

Pada kode diatas, kita menambahkan gambar cover di baris pertama, **StackPanel** yang berisi informasi umum di baris kedua, dan kemudian **ScrollView** yang berisi teks biografi di baris ketiga.

Kemudian kita perlu menambahkan properti berikut sebagai referensi objek Puteri yang dipilih dari halaman sebelumnya, ke dalam kelas **DetailPage** seperti berikut :

DetailPage.xaml.cs

```

26 Puteri _selected;

```

Selanjutnya kita perlu menangani informasi yang akan diterima dari **MainPage.xaml.cs**, dengan menambahkan kode berikut di **OnNavigatedTo** :

DetailPage.xaml.cs

```

38 protected override void OnNavigatedTo(NavigationEventArgs e)
39 {
40     int id = Convert.ToInt32(e.Parameter);
41     _selected = App.ViewModel.ListPuteri.FirstOrDefault(p => p.Id == id);
42     this.DataContext = _selected;
43 }

```

Tidak seperti pada versi 8, pada Windows Phone 8.1 WinRT ini, kita perlu menangani tombol back secara manual, tambahkan kode berikut di dalam konstruktor kelas **DetailPage** :

DetailPage.xaml.cs

```

32 HardwareButtons.BackPressed += HardwareButtons_BackPressed;

```

Selanjutnya menambahkan delegate method event handler yang berkaitan, untuk bisa back ke halaman sebelumnya :

DetailPage.xaml.cs

```

35 void HardwareButtons_BackPressed(object sender, BackPressedEventArgs e)
36 {
37     Frame frame = Window.Current.Content as Frame;
38     if (frame == null)
39     {
40         return;
41     }

```

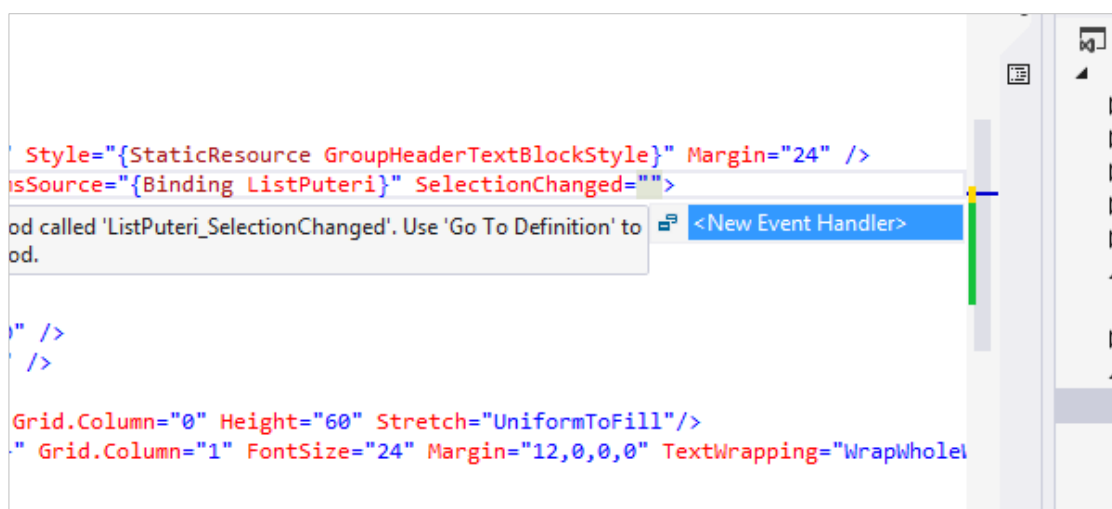
```

42
43     if (frame.CanGoBack)
44     {
45         frame.GoBack();
46         e.Handled = true;
47     }
48 }

```

Kembali ke MainPage.xaml, kita memerlukan subscribe event **SelectionChanged**, sehingga ketika kita melakukan tap item di **ListView** akan memicu event **SelectionChanged**.

Cara subscribe event tersebut, cukup ketikkan **SelectionChanged** hingga muncul opsi New Event Handler, pilih opsi tersebut.



Nanti di berkas MainPage.xaml.cs akan muncul delegate method event handler **ListPuteri_SelectionChanged** yang akan dieksekusi ketika event **SelectionChanged** terjadi. Tambahkan kode berikut di dalam method tersebut :

DetailPage.xaml.cs

```

53 private void ListPuteri_SelectionChanged(object sender, SelectionChangedEventArgs e)
54 {
55     var listview = (ListView)sender;
56     var selected = (Puteri)listview.SelectedItem;
57     if (selected == null) return;
58
59     Frame.Navigate(typeof(DetailPage), selected.Id);
60 }

```

Pada kode diatas, akan didapatkan referensi objek item Puteri terpilih, yang kemudian berpindah halaman ke **DetailPage** dengan parameter Id dari Puteri tersebut.

Sekarang jalankan project, dan lihat hasilnya. Coba pilih salah satu putri yang ada di list yang otomatis akan membawa ke halaman baru seperti tampilan dibawah:



Menu AppBar

Cara menambahkan AppBar pada Windows Phone 8.1 WinRT sedikit berbeda dibandingkan versi Silverlight.

Menariknya kita tidak perlu menambahkan ikon secara manual, karena ikon yang umum dipakai sudah tersedia, tinggal pakai saja.

Berikut kode untuk menambahkan AppBar di XAML.

DetailPage.xaml	
26	<Page.BottomAppBar>
27	<CommandBar IsSticky="True" x:Name="appBar">
28	<CommandBar.PrimaryCommands>
29	<AppBarButton Icon="Find" IsCompact="False" Label="Cari"
30	Click="Cari_Click"/>
31	</CommandBar.PrimaryCommands>
32	<CommandBar.SecondaryCommands>
33	<AppBarButton Label="Bagikan" Icon="ReShare" Click="Share_Click"/>
34	</CommandBar.SecondaryCommands>
35	</CommandBar>
36	</Page.BottomAppBar>

Dan kemudian saya tambahkan method event handler untuk menangani dua tombol appbar diatas :

DetailPage.xaml.cs	
66	<code>private async void Cari_Click(object sender, RoutedEventArgs e)</code>
67	<code>{</code>
68	<code> Uri link = new Uri("http://www.bing.com/search?q=" + _selected.Nama,</code>
69	<code>UriKind.Absolute);</code>
70	<code> await Launcher.LaunchUriAsync(link);</code>
71	<code>}</code>
72	
73	<code>private void Share_Click(object sender, RoutedEventArgs e)</code>
74	<code>{</code>
75	<code> Windows.ApplicationModel.DataTransfer.DataTransferManager.ShowShareUI();</code>
76	<code>}</code>

Jangan lupa, untuk subscribe event ketika DataRequested diminta pada method OnNavigatedTo, dengan kode berikut :

DetailPage.xaml.cs	
65	<code>DataTransferManager dtManager = DataTransferManager.GetForCurrentView();</code>
66	<code>dtManager.DataRequested += dtManager_DataRequested;</code>

Selanjutnya menambahkan data yang akan di share di dalam delegate method event handler **DataRequested** :

DetailPage.xaml.cs	
80	<code>private void dtManager_DataRequested(DataTransferManager sender,</code>
81	<code>DataRequestedEventArgs args)</code>
82	<code>{</code>
83	<code> args.Request.Data.Properties.Title = _selected.Nama;</code>
84	<code> args.Request.Data.Properties.Description = _selected.Asal;</code>
85	<code> args.Request.Data.SetWebLink(new Uri("http://putuyoga.com"));</code>
86	<code>}</code>

Coba jalankan app, dan lihat hasilnya. Tap salah satu puteri, kemudian lihat **AppBar** yang ada di bawah, tap tombol menu yang ada disitu.

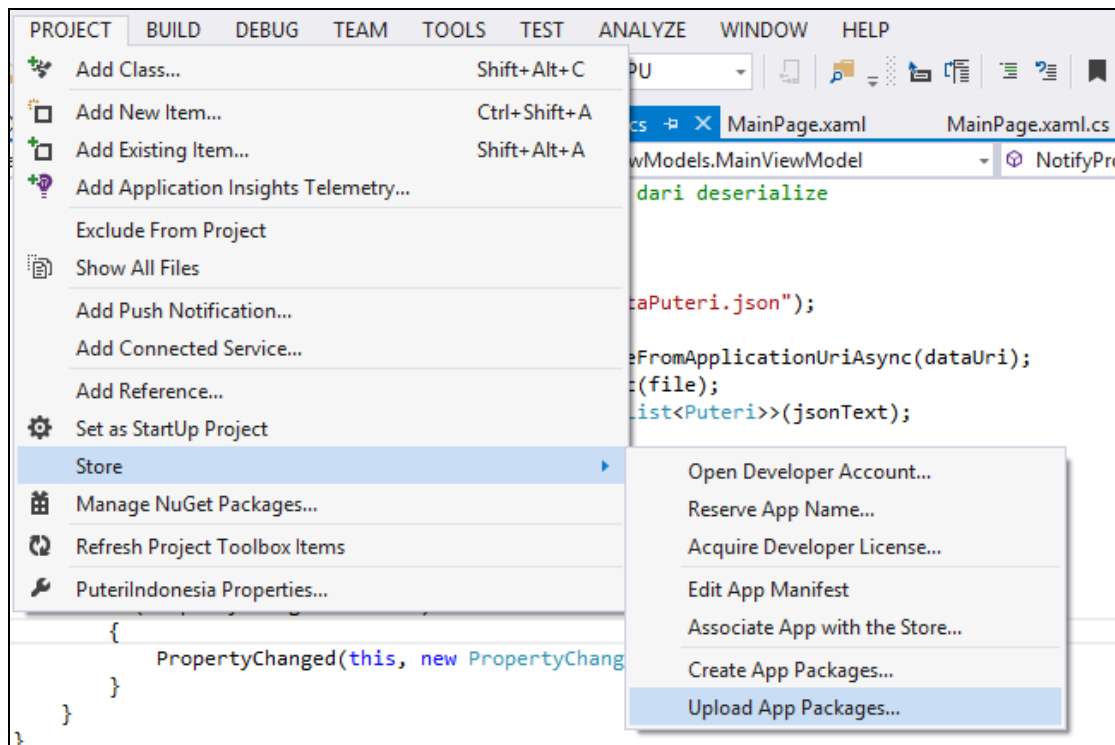
Dibawah merupakan tampilan **AppBar** yang berhasil diimplementasikan.



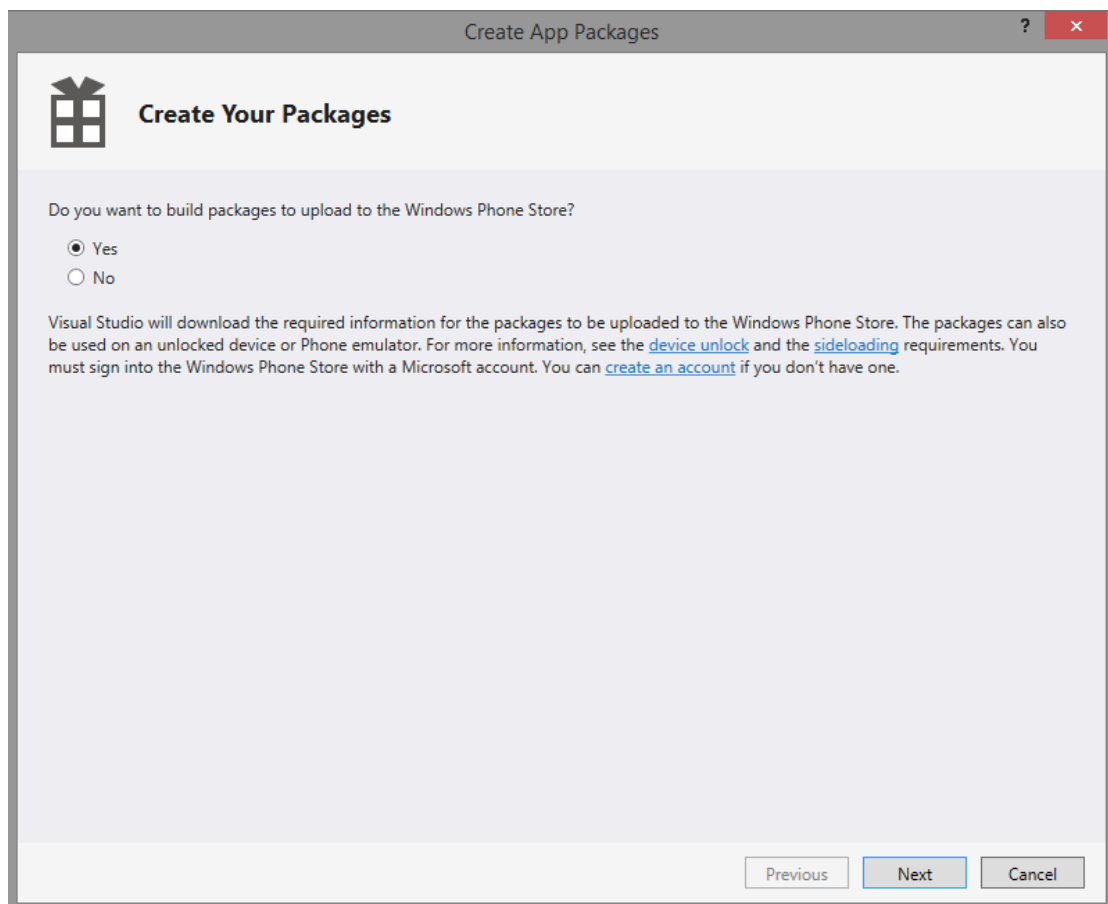
Publish ke Store

Sebelum bisa publish ke store, pastikan kamu sudah punya akun store, jika belum punya akun dan kebetulan masih mahasiswa, bisa mengikuti artikel disini untuk mendaftarkan akun dreamspark dengan benefit akun developer gratis.

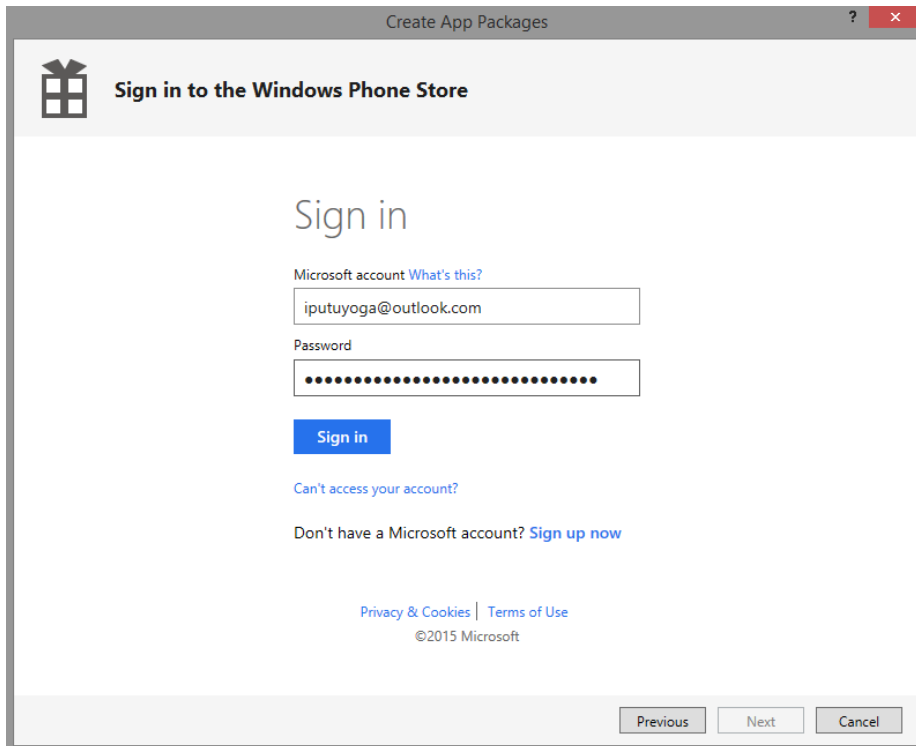
Pertama-tama kita harus buat packages yang dibutuhkan untuk di upload. Buka menu Project → Store → Create Package.



Kemudian akan muncul window Create Your Package, pada pertanyaan ingin mengunggah package ke store, pilih opsi Yes dan klik tombol Next.

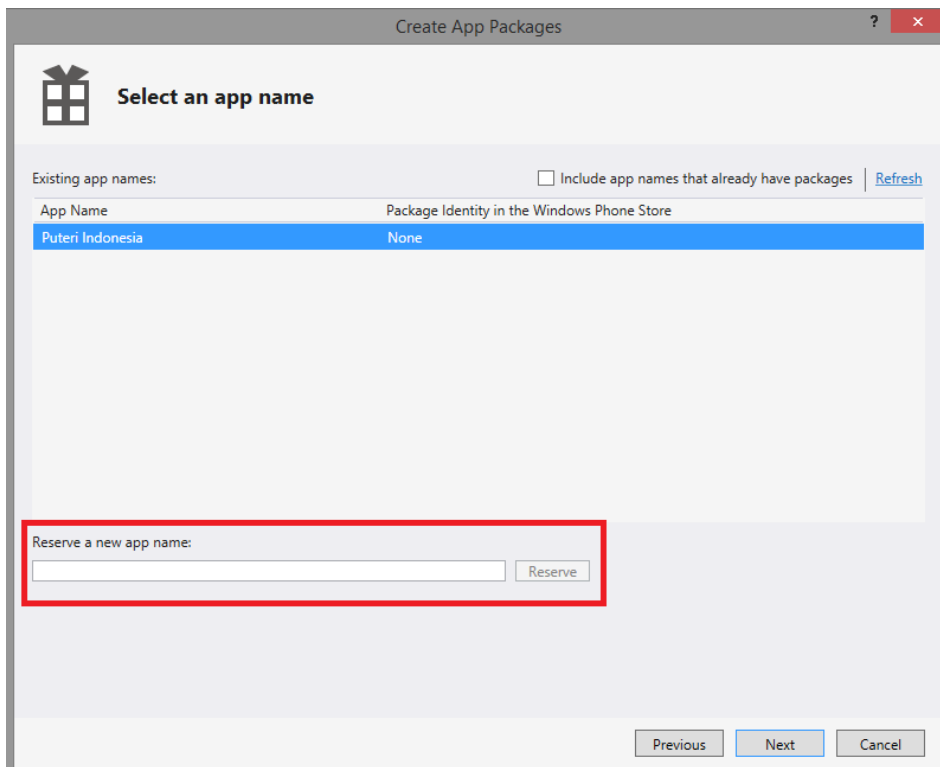


Kemudian login ke akun Microsoft yang berasosiasi dengan akun Windows developer-mu.



The screenshot shows a window titled "Create App Packages" with a sub-header "Sign in to the Windows Phone Store". It features a "Sign in" section with a Microsoft account login form. The email field contains "iputuyoga@outlook.com" and the password field is masked with dots. A blue "Sign in" button is present. Below the login fields are links for "Can't access your account?" and "Don't have a Microsoft account? Sign up now". At the bottom, there are links for "Privacy & Cookies" and "Terms of Use", and a copyright notice "©2015 Microsoft". Navigation buttons "Previous", "Next", and "Cancel" are at the bottom right.

Kemudian kita harus memilih App Name yang akan kita gunakan. Jika belum ada nama yang di reservasi, kamu harus mereservasi sebuah nama terlebih dahulu. Jika sudah klik Next



The screenshot shows a window titled "Create App Packages" with a sub-header "Select an app name". It displays a table of "Existing app names" with columns "App Name" and "Package Identity in the Windows Phone Store". The table has one row: "Puteri Indonesia" with "None" as the package identity. A checkbox "Include app names that already have packages" is unchecked, and a "Refresh" link is available. Below the table, there is a section "Reserve a new app name:" with a text input field and a "Reserve" button. This section is highlighted with a red rectangle. Navigation buttons "Previous", "Next", and "Cancel" are at the bottom right.

Selanjutnya atur konfigurasi dari package, seperti lokasi output package, nomor versi dan lain-lain.

Create App Packages

Select and Configure Packages

Output location:
 C:\Users\Putu\documents\visual studio 2013\Projects\PuterIndonesia\PuterIndonesia\AppPackages\

Version:
 1 . 1 . 0 . 0

☒ Automatically increment

Generate app bundle:
 If needed

[What does an app bundle mean?](#)

Select the packages to create and the solution configuration mappings:

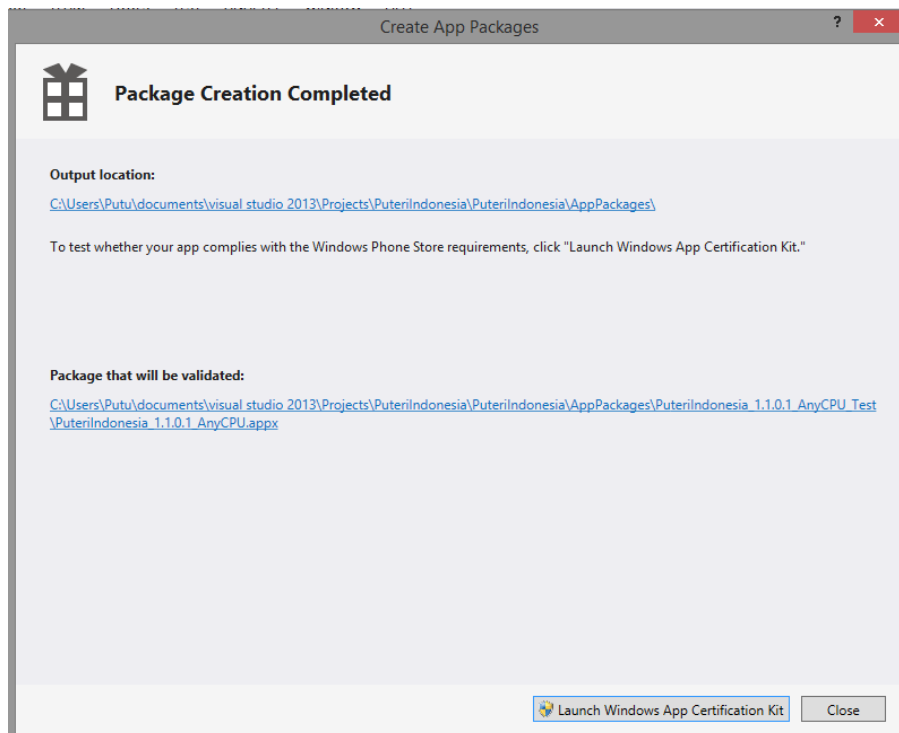
Architecture	Solution Configuration
☒ Neutral	Release (Any CPU)
☐ x86	Release (x86)
☐ ARM	Release (ARM)

☒ Include public symbol files, if any, to enable crash analysis for the app

Previous Create Cancel

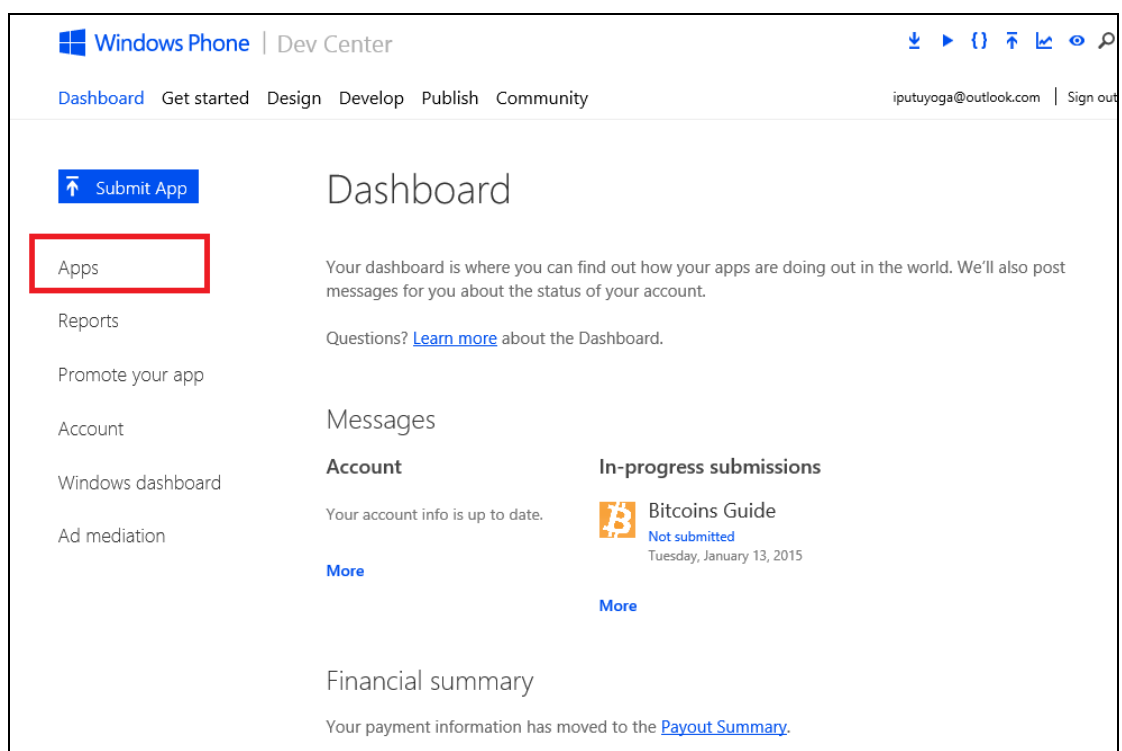
The option to create x86 package is disabled because it's not a valid package that can be published to the Windows Phone Store.

Tunggu sampai proses build dan pembuatan package selesai, nantinya akan muncul window seperti dibawah ini.

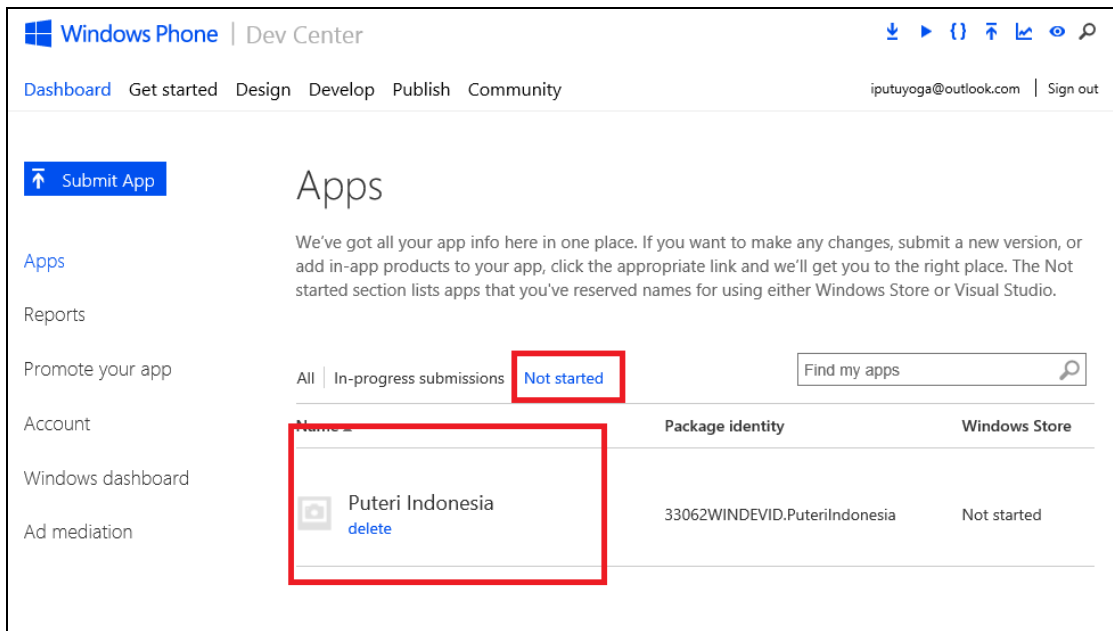


Selanjutnya login ke akun developer di <http://dev.windows.com> dan kemudian buka windows phone store dashboard <http://dev.windowsphone.com/dashboard>.

Klik menu Apps yang ada di sisi kiri, untuk membuka daftar apps.



Klik tab Not Started, dan pilih nama Apps yang telah kamu reservasi sebelumnya via Visual Studio.



Windows Phone | Dev Center

Dashboard Get started Design Develop Publish Community iputuyoga@outlook.com | Sign out

[Submit App](#)

Apps

We've got all your app info here in one place. If you want to make any changes, submit a new version, or add in-app products to your app, click the appropriate link and we'll get you to the right place. The Not started section lists apps that you've reserved names for using either Windows Store or Visual Studio.

[All](#) | [In-progress submissions](#) | **[Not started](#)**

Find my apps

Name	Package identity	Windows Store
 Puteri Indonesia delete	33062WINDEVID.PuteriIndonesia	Not started

Kamu akan dialihkan ke halaman submission, dimana pertama-tama akan diminta mengisi informasi kategori apps, harga dll seperti yang terlihat pada gambar dibawah ini, jika sudah klik tombol Save.

Name*

Names reserved for this app

Puteri Indonesia

You can reserve another name for this app to use in another language or to change its name. [Reserve app name](#)

Package identity name

33062WINDEVID.PuteriIndonesia

Windows publisher name

CN=0ECA11FD-D137-4297-BB0A-C261D4BA30F4

App category

The Category and Subcategory determine where the app will be listed in the Store. [Learn more](#).

Category*

<select one>

Subcategory

Pricing

Base price*

Free or paid? If paid, how much? [Learn more](#) about how this affects pricing in different countries/regions and how the Commerce Expansion Adjustment may affect your proceeds in certain countries/regions.

0.00

IDR

If you'd like to change the pricing and make this a paid app, you'll need to provide your [tax and bank info](#).

☐ Offer free trials of this app. Before you select this option, make sure you've implemented a trial experience in your app. [Learn more](#).

Market distribution

For certain markets, you'll need to provide a game rating to distribute your game. If you don't provide the appropriate info, we can't distribute the game in those markets. [Learn more](#).

☒ Distribute to all available markets at the base price tier

☐ Distribute to all markets except China. [Learn more](#).

☐ Continue distributing to current markets

More options ▼

Save

Kemudian klik “Upload and describe your package(s)” untuk mengunggah package yang telah dibuat sebelumnya.

Required

✓

App info

Give your app a name, price it, and enter other relevant info

2

Upload and describe your package(s)

For each package in your app, this is where you'll enter descriptions and upload screenshots that will showcase your app in the Store.

Selanjutnya kamu diminta untuk mengunggah package app yang ingin di publish di store. Klik add new kemudian pilih package yang akan di unggah.

Pada kasus ini berkas yang diunggah bernama PuteriIndonesia_1.1.0.1_AnyCPU.appxupload

Packages

This is an important page, because in addition to uploading your package, you're also creating your customer's first impression of your app. The info you provide will be part of the Store's listing of your app. If you're updating an existing app, this page will also include packages that you've already uploaded. All these packages will be available in the Store after you've published your submission.

Add new

Save

Proses berikutnya package akan diunggah dan jika telah selesai kita diminta untuk mengisi deskripsi dari app, kata kunci, ikon app berukuran 300x300, screenshot dari app dengan resolusi minimum WXGA.

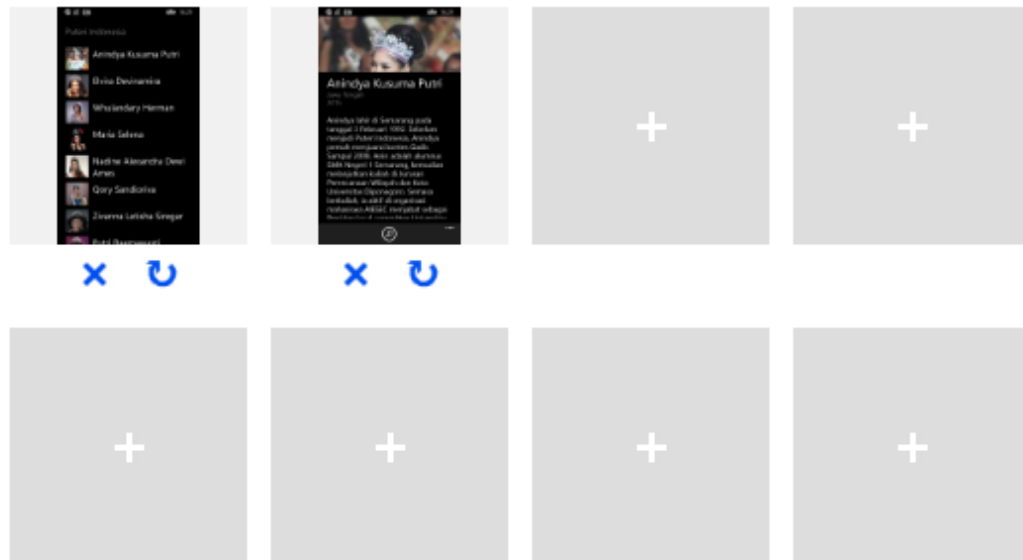
Screenshots

☒ Automatically create lower resolution screenshots from WXGA

WXGA | 720p | WVGA

* We'll use these screenshots to showcase your app.

For WXGA they must be 768 x 1280 px or 1280 x 768 px. [Learn more.](#)



Jika semua sudah diisi, tekan tombol save. Tahap kedua sudah selesai, kemudian tekan tombol submit & review.

Nantinya akan muncul tampilan perubah-perubahan apa saja yang ada yang harus kamu review. Jika dirasa info yang tertera sudah benar, tekan tombol Submit. Namun jika kamu merasa ada kesalahan, tekan tombol Go Back and Edit.

App

☒ Only show changes I made

Field	Published info	New info	
Package identity name	-	33062WINDEVID.PuteriIndonesia	
Category	-	entertainment	Edit
Subcategory	-	none	Edit
Publish as hidden	-	No	Edit
Allow trial downloads	-	No	Edit
Base price	-	0.00 IDR	Edit
Markets	-	191 total	Edit

Packages

Package name	Version	OS	Resolution	Language
 PuteriIndonesia_1.1.0.1_AnyCPU.appx	1.1.0.1	8.1	WVGA, 720P, WXGA	English

[Submit](#)
[Go back and edit](#)

Klik tombol Submit, dan app kamu akan menjalani proses serifikasi yang memakan waktu sampai dengan 5 hari, sebelum bisa publish di Store.

App submission successful!

Puteri Indonesia

If your submission needs to be tested, it will take us up to 5 business days to make sure it meets your certification requirements. If you want to publish it faster, you can either publish it yourself or it will go live automatically if you chose to automatically publish once it's been processed or if we have additional questions.

Would you like to...

[Submit another app](#)
[Go to the Lifecycle page](#)
[Go to the Dashboard](#)

Akhir Kata

Oke mungkin sampai disini saja pembahasan kali ini. Berhubung saya manusia biasa pasti ada kesalahan baik itu kata maupun informasi, jadi sekiranya mohon dimaklumi.

Selain itu juga masih banyak bagian yang bisa dikembangkan pada modul ini. Kebetulan belum punya akun developer ? dan kamu mahasiswa ? coba cek artikel berikut :

1. <http://winpoin.com/cara-daftar-akun-dreamspark>
2. <http://winpoin.com/cara-daftar-akun-developer-windows-via-dreamspark>

WinPoin

Kedepannya saya akan mempublikasikan artikel-artikel tentang tutorial dan tips pengembangan app di Windows atau Windows Phone di situs <http://winpoin.com>.

Source code lengkap bisa didapatkan di :

<https://github.com/putuyoga/PuteriIndonesiaWP8.1>

Tentang Penulis

I Putu Yoga Permana merupakan salah satu Microsoft Student Partner Indonesia sekaligus developer windows phone apps sejak akhir 2012.

Selain mengembangkan apps, juga suka menulis artikel gadget dari jaman smp. Dia juga beberapa kali mengisi workshop pengembangan aplikasi windows phone di beberapa kampus di jawa timur.



Jika kamu penasaran, apa saja app-nya dia, akses <http://bit.ly/1PiEDWI>.

Informasi Kontak

Jika ada pertanyaan, saran, maupun kritik bisa menghubungi via

- Email : putuyoga@outlook.com
- Facebook : <http://facebook.com/putu.yoga.permana>
- Twitter : <http://twitter.com/putuyoga>
- Website : <http://putuyoga.com>