

SKRIPSI

**PENGEMBANGAN APLIKASI PENCARI
RUTE KENDARAAN UMUM
UNTUK WINDOWS PHONE**



YOHAN

NPM: 2011730048

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2014**

UNDERGRADUATE THESIS

**DEVELOPMENT APPLICATION PUBLIC TRANSPORT
ROUTE SEARCH FOR WINDOWS PHONE**



YOHAN

NPM: 2011730048

**DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2014**

LEMBAR PENGESAHAN

PENGEMBANGAN APLIKASI PENCARI RUTE KENDARAAN UMUM UNTUK WINDOWS PHONE

YOHAN

NPM: 2011730048

Bandung, 1 Juli 2014

Menyetujui,

Pembimbing Tunggal

Pascal Alfadian, M.Com.

Ketua Tim Penguji

Anggota Tim Penguji

Thomas Anung Basuki, Ph.D.

Dr. rer. nat. Cecilia Esti Nugraheni

Mengetahui,

Ketua Program Studi

Thomas Anung Basuki, Ph.D.

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PENGEMBANGAN APLIKASI PENCARI RUTE KENDARAAN UMUM UNTUK WINDOWS PHONE

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal 1 Juli 2014

Meterai

Yohan
NPM: 2011730048

ABSTRAK

Sedang Dalam Pembuatan.

Kata-kata kunci: Rute, Kendaraan Umum, Windows Phone

ABSTRACT

Under Construction.

Keywords: Route, Public Transport, Windows Phone

Dipersembahkan untuk diri sendiri

KATA PENGANTAR

Puji Syukur penulis kepada Tuhan yang telah memberikan rahmatnya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Pencari Rute Kendaraan Untuk Windows Phone"

Bandung, Juli 2014

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xviii
DAFTAR TABEL	xix
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Ruang Lingkup Masalah	2
1.6 Metode Penelitian	2
1.7 Teknik Pengumpulan Data	3
1.8 Sistematika Penulisan	3
2 DASAR TEORI	5
2.1 Windows Phone	5
2.1.1 Phone Control	5
2.1.2 Navigation	8
2.1.3 Lifecycle Windows Phone	10
2.1.4 Peta di Windows Phone	11
2.1.5 Memanfaatkan Sumber Data	16
2.2 Kiri API	18
2.2.1 Pengantar Kiri API	18
2.2.2 Routing Web Service	18
2.2.3 Web Service Pencarian Lokasi	19
2.2.4 Web Service Menemukan Transportasi Terdekat	20
DAFTAR REFERENSI	23

DAFTAR GAMBAR

2.1	Kelas Diagram Button	6
2.2	TextBox	7
2.3	Memilih dari ListPicker	7
2.4	Hirarki Navigasi	9
2.5	Keluaran Toolkit Pushpin pada Peta	12

DAFTAR TABEL

BAB 1

PENDAHULUAN

Pada Bab satu akan dibahas pendahuluan dari penelitian yang dilakukan. Bab satu terbagi dalam delapan *sub bab*, yaitu *latar belakang*, *rumusan masalah*, *tujuan*, *batasan masalah*, *ruang lingkup masalah*, *metode penelitian*, *teknik pengumpulan data*, dan *sistematika penulisan*.

1.1 Latar Belakang

Transportasi menjadi bagian yang penting bagi manusia di saat penelitian ini dilakukan. Ada dua jenis transportasi bagi seseorang yaitu kendaraan umum dan kendaraan pribadi. Tapi sekarang ini banyak yang lebih memilih kendaraan pribadi dibanding kendaraan umum. Maraknya penggunaan kendaraan pribadi dan penambahan jalur kendaraan yang tidak sebanding banyaknya kendaraan menimbulkan kemacetan. Maraknya penggunaan kendaraan pribadi dikarenakan kurang nyamannya kendaraan umum dan kesulitan dalam menentukan kendaraan umum yang harus dinikmati. Banyaknya rute kendaran umum membuat orang kebingungan dalam memilih kendaraan umum menuju lokasi yang diinginkan. Seseorang cenderung malas untuk bertanya dan mencari rute yang efisien. Karena hal tersebut membuat seseorang lebih memilih menggunakan kendaraan pribadi ketimbang kendaraan umum.

Ide pembuatan aplikasi yang memudahkan seseorang dalam menentukan rute kendaraan umum sudah lebih dulu ada yang dikenal dengan nama Kiri. Kiri dibuat dengan latar belakang tiga masalah besar yaitu pemanasan global, kemacetan, dan harga bahan bakar minyak yang tinggi. Meskipun Kiri pertama dibuat di web tetapi Kiri dapat dimanfaatkan untuk pencarian kendaraan selain di web. Pemanfaatan Kiri tersebut dalam mencari rute kendaraan umum dengan menggunakan Kiri API.

Pesatnya perkembangan teknologi sekarang ini mendorong perkembangan perangkat bergerak (*mobile*). Perangkat bergerak kian digemari orang-orang terutama di Indonesia. Salah satu yang menarik perhatian adalah Windows Phone 8 yang dibuat Microsoft. Antarmuka Windows Phone 8 yang disebut *Metro* cukup menarik dan mudah digunakan. Meskipun jumlah penggunanya masih belum sebanyak pengguna Android dan IOS tapi jumlah penggunanya terus naik di tahun 2014 ini.

Berdasarkan hal tersebut, penulis mencoba mengembangkan aplikasi Pencarian Rute Kendaraan Umum di Windows Phone dalam tugas akhir ini. Aplikasi yang penulis kembangkan akan memungkinkan pengguna menemukan rute kendaraan umum untuk sampai di tujuan. Untuk memudahkan pengguna, penulis akan menampilkan dalam 2 bentuk yaitu peta dan daftar.

1.2 Rumusan Masalah

Sehubung dengan latar belakang diatas timbul permasalahan sebagai berikut:

- Bagaimana membuat aplikasi di Windows Phone?
- Bagaimana mengintegrasikan Kiri API dengan aplikasi pencari rute kendaraan umum di Windows Phone?

1.3 Tujuan

Berdasarkan rumusan masalah pada sub bab 1.2, maka tujuan dari pembuatan tugas akhir ini adalah:

- Mempelajari cara pembuatan perangkat lunak di Windows Phone lalu mengembangkan aplikasi yang akan dibuat.
- Membuat aplikasi di di Windows Phone yang memanfaatkan Kiri API.

1.4 Batasan Masalah

Ruang lingkup pengembangan aplikasi Pencari Rute Kendaraan untuk Windows Phone ini dibatasi hal berikut:

- Aplikasi ini akan berjalan di sistem operasi Windows Phone 8.
- Aplikasi ini membutuhkan koneksi internet.
- Aplikasi ini akan menampilkan rute jalur angkot, bus umum dan travel di tiga kota besar yaitu Bandung, Jakarta, dan Surabaya.

1.5 Ruang Lingkup Masalah

1.6 Metode Penelitian

Metode Penelitian yang penulis gunakan dalam membuat tugas akhir ini adalah sebagai berikut:

- Melakukan studi pustaka mengenai tombol di Windows Phone, navigation di Windows Phone, Map di Windows Phone, GPS di Windows Phone dan Kiri API.
- Melakukan analisis terhadap aplikasi lain yang menggunakan Kiri API.
- Melakukan analisis terhadap dasar teori untuk pembangunan aplikasi Pencarian Rute Kendaraan Umum untuk Windows Phone.
- Melakukan perancangan aplikasi Pencarian Rute Kendaraan Umum untuk Windows Phone.
- Implementasi dari aplikasi Pencarian Rute Kendaraan Umum untuk Windows Phone.

- Menguji aplikasi Pencarian Rute Kendaraan Umum untuk Windows Phone.
- Membuat kesimpulan.

1.7 Teknik Pengumpulan Data

1.8 Sistematika Penulisan

Bab 1 membahas latar belakang masalah, rumusan masalah, tujuan penulisan tugas akhir, batasan masalah, ruang lingkup masalah, metode penelitian, dan teknik pengumpulan data tugas akhir ini.

Bab 2 membahas tentang teori-teori yang akan digunakan dalam tugas akhir ini. Teori-teori yang dijelaskan mengenai Kiri API, *Web Service*, pembangunan aplikasi di Windows Phone, antarmuka di Windows Phone, dan algoritma yang dipakai.

Bab 3 membahas tentang analisi pembangunan aplikasi Pencarian Rute Kendaraan Umum untuk Windows Phone.

BAB 2

DASAR TEORI

Bab ini berisi dasar teori dari pembangunan Aplikasi Pencarian Rute Kendaraan Umum untuk Windows Phone. Beberapa teori yang dibahas dalam bab ini adalah Kiri API, Web Service, Menampilkan Peta, Penggunaan *Global Positioning System* di Windows Phone, dan antarmuka perangkat lunak yang dibuat.

2.1 Windows Phone

Sub bab ini akan membahas pemrograman di Windows Phone. Pembahasan akan dimulai dengan apa itu Windows Phone dan fitur di Windows Phone yang akan digunakan dalam pembangunan perangkat lunak Pencarian Rute Kendaraan di Windows Phone.

Windows Phone merupakan sistem operasi untuk perangkat bergerak yang dikembangkan Microsoft.¹ Untuk mengembangkan aplikasi Windows Phone dibutuhkan Windows Desktop 8 sebagai media pengembangan. Bahasa pemrograman yang digunakan untuk membuat perangkat lunak di Windows Phone yaitu C#.

2.1.1 Phone Control

Phone Control merupakan blok bangunan dari antarmuka aplikasi. Windows Phone 8 menyediakan banyak set untuk hal ini dari tombol, kontrol masukan untuk mendapatkan informasi sampai ke menu.

2.1.1.1 Tata Letak

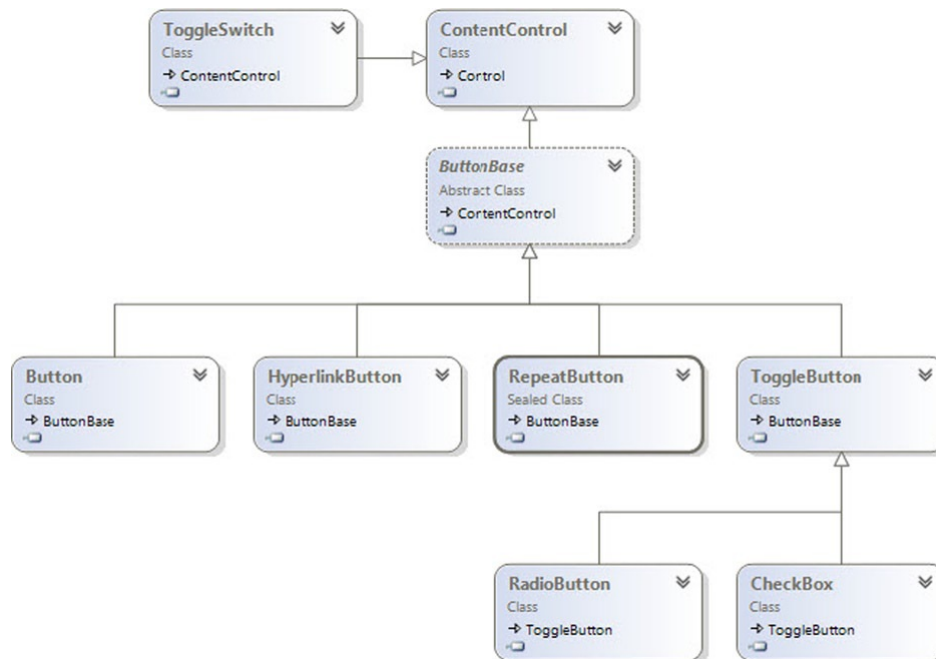
Ada 3 objek yang dipakai untuk menangani Tata Letak yaitu Grid, StackPane, dan Canvas. Perlu diperhatikan bahwa setiap halaman hanya memiliki satu Tata Letak.

- Grid memposisikan elemen di baris dan kolom dari sel.
- StackPanel memposisikan element di jajaran horizontal atau vertical.
- Canvas memposisikan elemen sebagai absolut koordinat.

¹en.wikipedia.org/wiki/Windows_Phone

2.1.1.2 Tombol Kontrol

Windows Phone 8 telah menyiapkan berbagai tombol yang dibutuhkan yaitu radio, check boxes, hyperlink dan repeat. Berikut tombol-tombol yang tersedia dalam bentuk kelas diagram.



Gambar 2.1: Kelas Diagram Button

Listing 2.1: Pemanggilan Search dengan Click

```

1 private void ClickButton_Click(object sender, RoutedEventArgs e)
2 {
3     SearchTask searchTask = new SearchTask();
4     searchTask.SearchQuery = "Windows Phone 8";
5     searchTask.Show();
6 }
  
```

Listing 2.2: Perintah Pemanggilan Search

```

1 public class SearchCommand : ICommand
2 {
3     public bool CanExecute(object parameter)
4     {
5         return true;
6     }
7     public event EventHandler CanExecuteChanged;
8     public void Execute(object parameter)
9     {
10        SearchTask searchTask = new SearchTask();
11        searchTask.SearchQuery = "Windows Phone 8";
12        searchTask.Show();
13    }
14 }
  
```

Listing 2.3: XAML Kode untuk Event Click dan Perintah

```

1 <Button Content="Fire Click" Click="button1_Click" />
2
3 <Button Content="Fire Command" Command="{StaticResource SearchCommand}" />
  
```

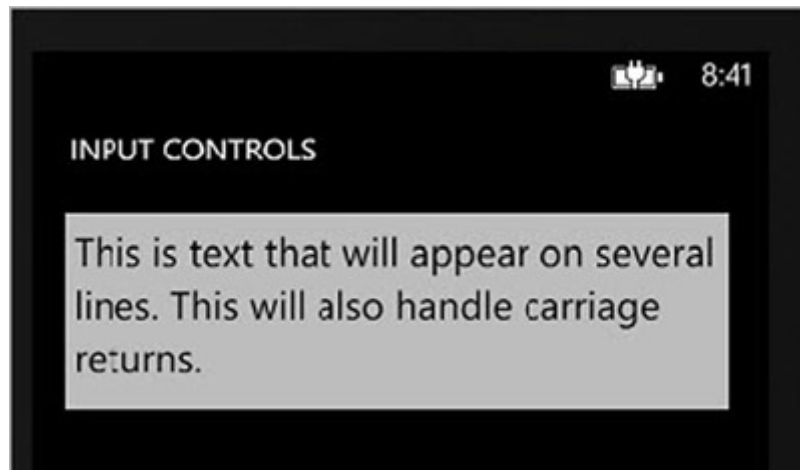
2.1.1.3 Kontrol Masukan

Semua input merupakan turunan dari kontrol, kelas yang memperkenalkan ControlTemplate yang mendefinisikan penampilan.

2.1.1.3.1 TextBox merupakan tempat masukan untuk data. Data dimasukan ketika keyboard muncul. Banyak properti yang bisa diatur dari sebuah TextBox mulai panjang maksimum hingga tata letak hingga *alignment*.

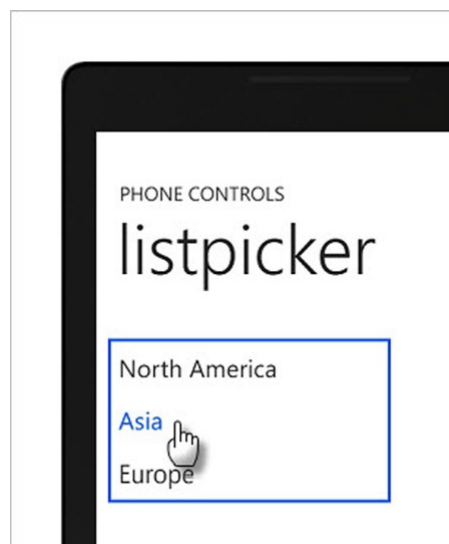
Listing 2.4: Kode Untuk TextBox

```
1 | <TextBox AcceptsReturn="True" TextWrapping="Wrap" Height="150"  
2 | Text="This is text that will appear on several lines. This will also handle carriage returns." />
```



Gambar 2.2: TextBox

2.1.1.3.2 ListPicker mirip dengan ComboBox untuk menampilkan beberapa item dan dapat dipilih salah satunya. Untuk tampilan ListPicker dapat ditampilkan dalam beberapa mode yaitu item yang dipilih yang ditampilkan, semua item ditampilkan di halaman baru, dan *popup*.



Gambar 2.3: Memilih dari ListPicker

Listing 2.5: ListPicker di XAML

```
1 | <toolkit:ListPicker x:Name="RegionPicker">  
2 |   <toolkit:ListPickerItem Content="North America" />  
3 |   <toolkit:ListPickerItem Content="Asia" />
```

```

4      <toolkit:ListPickerItem Content="Europe" />
5  </toolkit:ListPicker>

```

Listing 2.6: Mendefinisikan ListPicker dengan Kode

```

1  RegionPicker.Items.Add(new ListPickerItem ()
2  {
3      Content = "North America",
4      Tag = 123
5  });
6  RegionPicker.Items.Add(new ListPickerItem ()
7  {
8      Content = "Asia",
9      Tag = 456
10 });
11 RegionPicker.Items.Add(new ListPickerItem ()
12 {
13     Content = "Europe",
14     Tag = 789
15 });

```

Listing 2.7: Pengambilan Data ListPicker

```

1  <toolkit:ListPicker x:Name="RegionPicker"
2      ItemsSource="{StaticResource Regions}" >
3      <toolkit:ListPicker.ItemTemplate>
4          <DataTemplate>
5              <StackPanel Orientation="Horizontal">
6                  <TextBlock Text="{Binding Id}" Margin="5"
7                      Style="{StaticResource PhoneTextSubtleStyle}" />
8                  <TextBlock Text="{Binding RegionName}" Margin="5"/>
9              </StackPanel>
10         </DataTemplate>
11     </toolkit:ListPicker.ItemTemplate>
12 </toolkit:ListPicker>

```

2.1.2 Navigation

Ada tiga pilihan untuk bepergian di Windows Phone: dalam halaman, antara halaman, dan URIs eksternal. Navigasi antara halaman yang digunakan untuk perbedaan pandang seperti Panorama dan Pivot. Navigasi antara halaman menggunakan framework navigasi.

2.1.2.1 Navigasi Antar Halaman

2.1.2.1.1 Frame Aplikasi merupakan penampung terluar di hirarki kontrol untuk Windows Phone 8. Frame aplikasi ini mendukung navigasi antara halaman. Navigasi ke halaman yang lain akan menetapkan halaman yang baru sebagai frame.

Listing 2.8: Navigasi dengan Frame

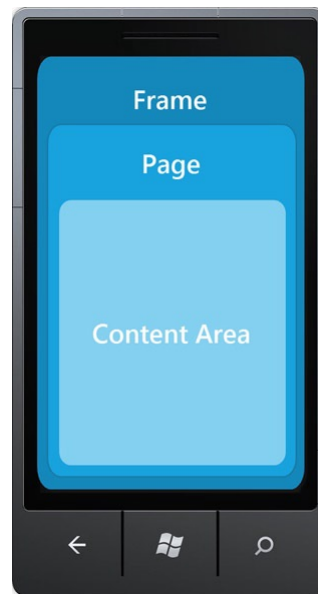
```

1  private void PhoneAppFrame_Click(object sender, RoutedEventArgs e)
2  {
3      App.RootFrame.Navigate(new Uri("/SecondPage.xaml", UriKind.Relative));
4  }

```

2.1.2.1.2 Page Aplikasi dapat dibedakan menjadi 4 yaitu OnFragmentNavigation, OnNavigationForm, OnNavigatedFrom, dan OnNavigatedTo. Berikut penjelasan dan contoh kodenya:

- OnFragmentNavigation: Fragment merupakan bagian URI yang diikuti tanda "#" dan dipakai untuk memberikan detail antar halaman. Pada kode dibawah menunjukan navigasi ke halaman dua sambil memberikan fragment string "detail".



Gambar 2.4: Hirarki Navigasi

Listing 2.9: Halaman Utama melewati fragment ke lamana kedua

```

1 private void PageFragment_Click(object sender, RoutedEventArgs e)
2 {
3     App.RootFrame.Navigate(new Uri("/SecondPage.xaml#detail", UriKind.Relative));
4 }

```

Listing 2.10: Halaman Utama melewati fragment ke lamana kedua

```

1 protected override void OnFragmentNavigation(FragmentNavigationEventArgs e)
2 {
3     // displays "Fragment: Detail"
4     MessageBox.Show("Fragment: " + e.Fragment);
5     base.OnFragmentNavigation(e);
6 }

```

- **OnNavigationForm:** Dipakai untuk langkah pencegahan saat pengguna akan meninggalkan halaman.

Listing 2.11: Sebelum Navigasi

```

1 protected override void OnNavigatingFrom(NavigatingCancelEventArgs e)
2 {
3     e.Cancel = MessageBox.Show("You have unsaved data, do you want to exit?", "Unsaved data",
4     MessageBoxButton.OKCancel) == MessageBoxResult.Cancel;
5     base.OnNavigatingFrom(e);
6 }

```

- **OnNavigatedFrom:** Metode dipanggil ketika halaman sudah tidak aktif lagi. **NavigationEventArgs** merupakan halaman yang dituju. Hal ini untuk mendukung pengiriman data dari halaman yang ditinggalkan dan mengirimnya ke halaman yang mau ditampilkan.

Listing 2.12: Setelah Navigasi

```

1 // Navigated from "SecondPage" back to "MainPage"
2 protected override void OnNavigatedFrom(NavigationEventArgs e)
3 {
4     MainPage mainPage = e.Content as MainPage;
5     if (mainPage != null)
6         mainPage.StatusText.Text = "A message from SecondPage OnNavigatedFrom";
7     base.OnNavigatedFrom(e);
8 }

```

- OnNavigatedTo: Metode dipanggil ketika halaman sudah aktif.

Listing 2.13: Navigasi ke Halaman Berikutnya

```

1 | protected override void OnNavigatedTo(NavigationEventArgs e)
2 | {
3 |     MessageBox.Show(" NavigatedTo: " + e.Uri);
4 |     base.OnNavigatedTo(e);
5 | }

```

2.1.2.1.3 Layanan Navigasi merupakan layanan untuk memantau riwayat navigasi, menyediakan metode navigasi, dan event. Layanan ini juga memungkinkan perpindahan navigasi sesuai riwayat menggunakan GoBack() dan GoForward(). Berikut contoh metodenya.

Listing 2.14: Metode Layanan Navigasi

```

1 | this.NavigationService.Navigate(new Uri("/SecondPage.xaml", UriKind.Relative));

```

2.1.2.2 Navigasi ke URIs Eksternal

Listing 2.15: Navigasi dengan Web Browser

```

1 | WebBrowserTask webBrowserTask = new WebBrowserTask();
2 | webBrowserTask.Uri = new Uri("http://www.falafel.com");
3 | webBrowserTask.Show();

```

2.1.3 Lifecycle Windows Phone

- Running : Aplikasi sedang berada di bagian depan. Hanya satu aplikasi yang diijinkan berada di depan pada satu waktu.
- Dormant : Aplikasi tidak berada di bagian depan, tetapi sistem operasi tidak membutuhkan sumber daya(dalam hal ini memori). Pada tahap ini objek aplikasi masih berada di memori, jadi pada tahap Dormant aplikasi dapat secara otomatis diaktifkan kembali. Tetapi jika sistem operasi membutuhkan sumber daya maka aplikasi akan menjadi *tombstoned* dan dihancurkan.
- Tombstoned : Aplikasi akan dihancurkan dan objek aplikasi akan dihapus dari memori. Ketika aplikasi diaktivasi pengguna akan ditempatkan kembali di halaman yang ditinggalkan sebelumnya, tetapi objek harus di kembalikan dari keadaan sebelumnya.

2.1.3.1 Penyimpanan dan Pengembalian Keadaan Aplikasi

Listing 2.16: App.xaml

```

1 | <Application.ApplicationLifetimeObjects>
2 |     <!--Required object that handles lifetime events for the application-->
3 |     <shell:PhoneApplicationService
4 |         Launching="Application_Launching" Closing="Application_Closing"
5 |         Activated="Application_Activated" Deactivated="Application_Deactivated"/>
6 | </Application.ApplicationLifetimeObjects>

```

Listing 2.17: App.xaml.cs

```

1 | // Code to execute when the application is launching (eg, from Start)
2 | // This code will not execute when the application is reactivated
3 | private void Application_Launching(object sender, LaunchingEventArgs e)
4 | {
5 | }
6 | // Code to execute when the application is activated (brought to foreground)
7 | // This code will not execute when the application is first launched
8 | private void Application_Activated(object sender, ActivatedEventArgs e)

```

```

9      {
10     }
11     // Code to execute when the application is deactivated (sent to background)
12     // This code will not execute when the application is closing
13     private void Application_Deactivated(object sender, DeactivatedEventArgs e)
14     {
15     }
16     // Code to execute when the application is closing (eg, user hit Back)
17     // This code will not execute when the application is deactivated
18     private void Application_Closing(object sender, ClosingEventArgs e)
19     {
20     }

```

Listing 2.18: Penanganan terhadap pengaktifan dan penonaktifan kejadian

```

1 public class MyObject
2 {
3     public DateTime LastUpdate { get; set; }
4 }
5 //...
6 public MyObject MyObject { get; set; }
7
8 private void Application_Activated(object sender, ActivatedEventArgs e)
9 {
10     DateTime lastUpdate =
11         (DateTime)PhoneApplicationService.Current.State["LastUpdate"];
12     Debug.WriteLine("Application_Activated: LastUpdate=" + lastUpdate.ToLongTimeString());
13 }
14 private void Application_Deactivated(object sender, DeactivatedEventArgs e)
15 {
16     Debug.WriteLine("Application_Deactivated: Saving LastUpdate to application state");
17     PhoneApplicationService.Current.State["LastUpdate"] = DateTime.Now;
18 }

```

2.1.4 Peta di Windows Phone

Peta untuk Windows Phone adalah Here Maps. Here Maps yang akan menampilkan lokasi dan melakukan penelusuran. Pada Sub Bab ini akan dibahas mengenai penambahan peta pada aplikasi, mendapatkan lokasi, petunjuk arah dan Pushpins.

2.1.4.1 Penambahan Maps Ke Aplikasi

Ada dua cara untuk menambahkan peta ke aplikasi yang pertama dengan menjalankan MapTask dan yang kedua dengan penambahan ID_CAP_MAP ke \properties\WMAppManifest.xml. Contoh dibawah adalah dengan MapTask yang memanggil metode Show() untuk memunculkan peta.

Listing 2.19: code pemanggilan maps

```

1 private void MapClick(object sender, EventArgs e)
2 {
3     var task = new MapsTask();
4     var goldenGateBridge = new GeoCoordinate(37.8085880, -122.4770175);
5     task.Center = goldenGateBridge;
6     task.ZoomLevel = 9;
7     task.SearchTerm = "Golden Gate Bridge";
8     task.Show();
9 }

```

Listing 2.20: penambahan Map Element menggunakan Map Control

```

1 <Grid x:Name="LayoutRoot" Background="Transparent">
2     <Controls:Map x:Name="WorldMap" />
3 </Grid>

```

2.1.4.2 Posisi Pada Peta

Ketika memulai membuka peta maka akan menampilkan seluruh dunia dan pengguna harus melakukan *pinch-and-zoom* untuk melihat titik tertentu. Agar dapat memudahkan pengguna dalam

memilih titik tertentu secara otomatis maka dapat menggunakan metode `SetView`. Pada Listing dibawah dapat dilihat bahwa `SetView` memiliki 2 parameter. Parameter pertama menyatakan tempat dengan `GeoCoordinate` yang mendefinisikan pusat pandangan. Selanjutnya parameter ke 2 merupakan `zoomLevel` (1 = zoomed out, 20 = zoomed in).

Listing 2.21: Posisi Pada Peta

```
1 GeoCoordinate GoldenGateBridge = new GeoCoordinate(37.8085880, -122.4770175);
2 SanFranciscoMap.SetView(GoldenGateBridge, 15);
```

2.1.4.3 Penambahan Pushpins ke Peta

Peta tidak mendukung langsung cara untuk mengikat data ke `MapLayer` dan `MapOverlay`. Untungnya di Windows Phone memiliki Windows Phone 8 Toolkit yang memiliki set objek yang dapat digunakan sebagai jembatan mengikat data ke `MapLayer` dan `MapOverlay`.

Listing 2.22: Berikut contoh Toolkit Pushpin dengan elemen-element yang di tulisa manual

```
1 <Controls:Map>
2   <Toolkit:MapExtensions.Children>
3     <Toolkit:Pushpin Content="London" GeoCoordinate="51.499493,-0.124753" />
4   </Toolkit:MapExtensions.Children>
5 </Controls:Map>
```

Gambar dibawah merupakan keluaran di peta. Label "London" merupakan bawaan Windows Phone 8 pushpin.



Gambar 2.5: Keluaran Toolkit Pushpin pada Peta

Selain menetapkan Pushpin secara langsung di `MapExtensions`, dapat pula diletakan `MapItemsControl`. Pushpin akan berpindah ke `ItemTemplate` dan `ItemSource` akan ditugaskan `ViewModel`.

Listing 2.23: Berikut contoh Toolkit Pushpin dengan elemen-element yang di tulisa manual

```
1 <Controls:Map x:Name="WorldMap" >
2   <Toolkit:MapExtensions.Children>
3     <Toolkit:MapItemsControl ItemTemplate="{StaticResource MapItemTemplate}" ItemsSource="{Binding
4       Source={StaticResource LocationViewModel}, Path=Locations}" />
5   </Toolkit:MapExtensions.Children>
6 </Controls:Map>
```

Listing 2.24: View Model

```
1 public class Location
2 {
3     public string Title { get; set; }
4     [TypeConverter(typeof(GeoCoordinateConverter))]
```



```

5      public GeoCoordinate GeoCoordinate { get; set; }
6  }
7
8  public class LocationViewModel
9  {
10     public ObservableCollection<Location> Locations { get; set; }
11     public LocationViewModel()
12     {
13         Locations = new ObservableCollection<Location>();
14     }
15 }

```

Listing 2.25: Deklarasi View Model

```

1  <phone:PhoneApplicationPage.Resources>
2      <vm:LocationViewModel x:Key="LocationViewModel" >
3          <vm:LocationViewModel.Locations>
4              <vm:Location Title="London" GeoCoordinate="51.499493,-0.124753" />
5              <vm:Location Title="Paris" GeoCoordinate="48.858222,2.2945" />
6              <vm:Location Title="Rome" GeoCoordinate="41.890268,12.492315" />
7          </vm:LocationViewModel.Locations>
8      </vm:LocationViewModel>
9      <DataTemplate x:Name="MapItemTemplate">
10         <Toolkit:Pushpin GeoCoordinate="{Binding GeoCoordinate}" Content="{Binding Title}" />
11     </DataTemplate>
12 </phone:PhoneApplicationPage.Resources>

```

2.1.4.4 Mendapatkan Posisi Pengguna

Di Windows Phone 8 telah ada `GeoCoordinate` yang dapat digunakan untuk mengetahui posisi pengguna. `Geolocator` dari `Windows.Devices.Geolocation` akan mengembalikan posisi saat ini. Tetapi untuk menggunakan `Geolocator`, penulis perlu menghidupkan `ID_CAP_LOCATION` di `\properties\WMAppManifest.xml`.

Contoh kode dari penggunaan menggunakan XAML didefinisikan pada listing dibawah. `UserLocationMarker` di `MapExtensions.Children.UserLocationMarker` merupakan `Toolkit` objek `Windows Phone 8`. `Visibilitas UserLocationMarker` di set `"collapsed"` agar penanda tetap berada di tempat sampai ada perubahan lokasi dan siap ditampilkan.

Listing 2.26: Map dan UserLocationMarker XAML

```

1  Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0" >
2      <Controls:Map x:Name="WorldMap">
3          <Toolkit:MapExtensions.Children>
4              <Toolkit:UserLocationMarker x:Name="locationMarker" Visibility="Collapsed" />
5          </Toolkit:MapExtensions.Children>
6      </Controls:Map>
7  </Grid>

```

Listing Dibawah akan mendemonstrasikan bagaimana `Geolocator` mendapatkan posisi terkini. Contoh dari metode menggunakan `event` `click`. setelah itu dipanggil `GetGeopositionAsync()` yang akan mengembalikan objek `Geoposition` object. `GetGeopositionAsync` yang sip menunggu, jadi tambahkan `async` untuk pemanggilan metode dan kata kunci `await` untuk mendapatkan nilai kembalian. `Geoposition` mengandung kordinat yang tidak kompatibel dengan `GeoCoordinate()` yang dipakai untuk map. Untungnya toolkit menyediakan metode `ToGeoCoordinate()` untuk menerjemahkan ke `GeoCoordinate` secara otomatis. Kode dibawah juga menjadi referensi pemakaian `UserLocationMarker`, `makes it visible`, dan di set menjadi `GeoCoordinate` baru. Terakhir, metode `Map SetView()` dipanggil untuk menjadi pusat dari posisi yang baru.

Listing 2.27: Setting ulang Geolocator

```

1  private Geolocator _locator;
2  protected override void OnNavigatedTo(NavigationEventArgs e)
3  {

```

```

4      base.OnNavigatedTo(e);
5      _locator = new Geolocator()
6      {
7          // set either ReportInterval (milliseconds) or MovementThreshold (meters)
8          ReportInterval = 5000,
9          //DesiredAccuracy = PositionAccuracy.High
10         DesiredAccuracyInMeters = 1
11     };
12     _locator.PositionChanged += _locator_PositionChanged;
13 }

```

Listing 2.28: Melepas Geolocator yang dapat dipakai untuk melindungi sumber daya

```

1  protected override void OnNavigatedFrom(NavigationEventArgs e)
2  {
3      _locator.PositionChanged -= _locator_PositionChanged;
4      _locator = null;
5      base.OnNavigatedFrom(e);
6  }

```

Listing 2.29: Menangani Perubahan Posisi

```

1  void _locator_PositionChanged(Geolocator sender, PositionChangedEventArgs args)
2  {
3      this.Dispatcher.BeginInvoke(() =>
4      {
5          var geoCoordinate = args.Position.Coordinate.ToGeoCoordinate();
6          var locationMarker = this.FindName("locationMarker") as UserLocationMarker;
7          if (locationMarker != null)
8          {
9              locationMarker.Visibility = System.Windows.Visibility.Visible;
10             locationMarker.GeoCoordinate = geoCoordinate;
11             WorldMap.SetView(geoCoordinate, 15);
12         }
13     });
14 }

```

Listing 2.30: Menangani Perubahan Status

```

1  void _locator_StatusChanged(Geolocator sender, StatusChangedEventArgs args)
2  {
3      this.Dispatcher.BeginInvoke(() =>
4      {
5          var textBlock = this.FindName("StatusText") as TextBlock;
6          textBlock.Text = "Status: " + args.Status.ToString();
7      });
8  }

```

Berikut nilai yang mungkin dari Status Posisi:

- *Ready* : Jika lokasi tersedia.
- *Initializing* : Status jika penangkap GPS belum memiliki cukup satelit untuk mendapatkan posisi yang akurat.
- *NoData* : Data lokasi belum tersedia. Status ini muncul jika aplikasi sedang memanggil `GetGeopositionAsync` atau register.
- *Disable* : Status mengindikasikan tidak diperbolehkannya pengaksesan lokasi.
- *NotInitialized* : Data lokasi belum tersedia. Status ini muncul jika aplikasi belum memanggil `GetGeopositionAsync` atau register.
- *NotAvailable* : Jika Windows sensor dan lokasi tidak tersedia.

2.1.4.5 Arah

Penambahan langkah demi langkah arah rute di aplikasi di Windows Phone 8 mudah dilakukan. Untuk memunculkan arah cukup memiliki titik awal dan titik akhir. Pada listing dibawah mendemonstrasikan pengarahannya di map antara 2 titik. Untuk titik awal dan akhir perlu label lokasi dengan objek `LabeledMapLocation`. Selanjutnya setelah menetapkan dua objek pada `MapsDirectionsTask` lalu panggil metode `show()` dan arah dari titik awal ke titik akhir akan muncul di peta.

Listing 2.31: Penggunaan Arah pada Map

```

1  var wharf = new LabeledMapLocation()
2  {
3      Label = "Wharf",
4      Location = new GeoCoordinate(36.96252, -122.023372)
5  };
6  var park = new LabeledMapLocation()
7  {
8      Label = "Lighthouse Park",
9      Location = new GeoCoordinate(36.95172, -122.026783)
10 };
11 var task = new MapsDirectionsTask() { Start = wharf, End = park };
12 task.Show();

```

Meskipun Map bisa menggambar rute secara otomatis ada satu cara lagi untuk menambahkan objek titik pada Map yaitu menggunakan `RouteQuery`. Pertama yang harus didefinisikan pada `RouteQuery` yaitu `GeoCoordinate` yang harus dikunjungi. Pada listing dibawah dapat dilihat 3 `GeoCoordinate` dijadikan `Waypoints` di `RouteQuery` yang baru. `RouteOptimization` di pakai untuk meminimalisir jarak dan `TravelMode` di setel `Driving`.

Listing 2.32: Mengatur Ulang Rute Query

```

1  var wharf = new GeoCoordinate(36.96252, -122.023372);
2  var park = new GeoCoordinate(36.95172, -122.026783);
3  var lagoon = new GeoCoordinate(36.963365, -122.031932);
4  var routeQuery = new RouteQuery()
5  {
6      Waypoints = new List<GeoCoordinate>() { wharf, park, lagoon },
7      RouteOptimization = RouteOptimization.MinimizeDistance,
8      TravelMode = TravelMode.Driving
9  };
10 routeQuery.QueryCompleted += routeQuery_QueryCompleted;
11 routeQuery.QueryAsync();

```

Untuk melakukan handle terhadap `QueryCompleted`, pertama harus di periksa apakah terdapat kesalahan. Kembalinya adalah objek rute. Lalu jadikan `MapRoute` konstruktor. `MapRoute` menambahkan visualisasi garis pada peta. Contoh implementasinya dapat dilihat di listing dibawah.

Listing 2.33: Melakukan Handle Terhadap QueryCompleted

```

1  void routeQuery_QueryCompleted(object sender, QueryCompletedEventArgs<Route> e)
2  {
3      if (e.Error == null)
4      {
5          this.WorldMap.AddRoute(new MapRoute(e.Result));
6          this.WorldMap.SetView(e.Result.BoundingBox);
7      }
8  }

```

Listing 2.34: Melakukan Perubahan Warna Rute di Map

```

1  var mapRoute = new MapRoute(e.Result)
2  {
3      Color = Colors.Red,
4      RouteViewKind = RouteViewKind.UserDefined
5  };
6  this.WorldMap.AddRoute(mapRoute)

```

Listing 2.35: Menampilkan Arah dalam Bentuk List

```

1  var sb = new StringBuilder();
2  var i = 0;
3  sb.AppendFormat(" Estimated time: {0} minutes\n",
4      e.Result.EstimatedDuration.TotalMinutes.ToString());
5  foreach (var leg in e.Result.Legs)
6  {
7      foreach (var maneuver in leg.Maneuvers)
8      {
9          sb.AppendFormat("{0}. {1}: {2}\n", ++i, maneuver.InstructionKind.ToString(), maneuver.
10             InstructionText);
11      }
12  }
13  MessageBox.Show(sb.ToString());

```

2.1.5 Memanfaatkan Sumber Data

2.1.5.1 Serializing Object

Serialization disini merupakan proses mentransformasikan objek ke format yang bisa dengan mudah dikirim melewati jaringan atau disimpan di database. Formatnya disini berupa string yang direpresentasikan sebagai objek di XML atau JSON (Javascript Object Notation). Ada beberapa objek yang dapat melakukan serialisasi, tetapi yang akan dibahas penulis disini hanya serialisasi JSON.

Banyak *web service* yang mengembalikan data dalam format JSON. JSON memiliki struktur yang mudah dipahami dimana kurung kurawal mengindikasikan objek, kurung siku berarti array, dan properti berupa nama dan nilai pasangan yang dipisahkan oleh titik dua. JSON format memiliki ukuran data yang kecil dan baik untuk penggunaan perangkat bergerak. Untuk contoh format JSON dapat dilihat di bagian Kiri API pada Bab 2 ini karena Kiri API menggunakan format JSON. Serialisasi menggunakan `DataContractJsonSerializer` membuat serialisasi mudah untuk menerjemahkan form String JSON ke objek yang dapat langsung digunakan. `DataContractJsonSerializer` memakai `WriteObject()` untuk serialisasi and `ReadObject()` untuk de-serialisasi.

2.1.5.2 Memanfaatkan Sumber Data Web

Listing 2.36: Pembentuk getPublicPhotos Url pada flickr

```

1  private const string BaseUrl = "http://ycpi.api.flickr.com/services/rest/";
2  private const string QueryStrings =
3      "?method={0}&api_key={1}&user_id={2}&format=json&nojsoncallback=1";
4  private const string FlickrMethod = "flickr.people.getPublicPhotos";
5  private const string YourApiKey = "<replace with api key here>";
6  private const string LibraryOfCongressKey = "8623220@N02";
7  private string FlickrPhotosUrl = BaseUrl +
8      String.Format(QueryStrings, FlickrMethod, YourApiKey, LibraryOfCongressKey);

```

Listing 2.37: Contoh JSON String dari flickr API

```

1  {
2      "photos":{
3          "page":1,
4          "pages":193,
5          "perpage":100,
6          "total":"19222",
7          "photo":[
8              {
9                  "id":"9319237625",
10                 "owner":"8623220@N02",
11                 "secret":"e0d73d680b",
12                 "server":"5537",
13                 "farm":6,

```

Selanjutnya untuk menerjemahkan JSON ke C# bisa lewat web sites json2csharp.com. Menuju ke json2csharp.com lalu paste JSON di text box yang ada dan click Generate. Selanjutnya buat class berekstensi .cs lalu paste text C# yang telah di generate.

Listing 2.38: Contoh JSON String dari flickr API

```

1  using System.Collections.Generic;
2
3  namespace ConsumingXML.Classes
4  {
5      public class Photo
6      {
7          public string id { get; set; }
8          public string owner { get; set; }
9          public string secret { get; set; }
10         public string server { get; set; }
11         public int farm { get; set; }
12         public string title { get; set; }
13         public int ispublic { get; set; }
14         public int isfriend { get; set; }
15         public int isfamily { get; set; }
16     }
17     public class Photos
18     {
19         public int page { get; set; }
20         public int pages { get; set; }
21         public int perpage { get; set; }
22         public string total { get; set; }
23         public List<Photo> photo { get; set; }
24     }
25     public class RootObject
26     {
27         public Photos photos { get; set; }
28         public string stat { get; set; }
29     }
30 }

```

Listing 2.39: Mengembalikan daftar FlickrPhoto

```

1  private static List<FlickrPhoto> GetFlickrPhotos(string json)
2  {
3      const string baseUrl =
4      "http://farm{0}.staticflickr.com/{1}/{2}_{3}_s.jpg";
5      List<FlickrPhoto> FlickrPhotos = null;
6      var serializer = new DataContractJsonSerializer(typeof(RootObject));
7      using (var stream = new MemoryStream(Encoding.UTF8.GetBytes(json)))
8      {
9          var root = serializer.ReadObject(stream) as RootObject;
10         FlickrPhotos = (from photo in root.photos.photo
11             select new FlickrPhoto
12             {
13                 Title = photo.title,
14                 Uri = new Uri(String.Format(baseUrl,
15                     photo.farm, photo.server, photo.id, photo.secret))
16             }).ToList();
17     }
18     return FlickrPhotos;
19 }

```

Listing 2.40: Menggunakan WebClient untuk mendownload String

```

1  private void UseWebClient()
2  {
3      var uri = new Uri(FlickrPhotosUrl);
4      var client = new WebClient();
5      client.DownloadStringCompleted += (sender, e) =>
6      {
7          var photos = GetFlickrPhotos(e.Result);
8          Dispatcher.BeginInvoke(() =>
9          {
10             FlickrListBox.DataContext = photos;
11         });
12     };
13     client.DownloadStringAsync(uri);
14 }

```

Listing 2.41: XAML untuk halaman

```

1  <Grid x:Name="LayoutRoot" >
2      <ListBox x:Name="FlickrListBox" ItemsSource="{Binding}">
3          <ListBox.ItemTemplate>
4              <DataTemplate>
5                  <StackPanel Orientation="Horizontal">
6                      <Image Stretch="UniformToFill" Width="100" Height="100" Source="{Binding Uri}" />
7                      <TextBlock Grid.Column="1" Margin="10,0,0,0" Text="{Binding Title}"
8                          HorizontalAlignment="Stretch" />
9                  </StackPanel>
10             </DataTemplate>
11         </ListBox.ItemTemplate>
12     </ListBox>
</Grid>

```

2.2 Kiri API

Sub bab ini akan membahas Dokumentasi dari Kiri API. Pembahasan dimulai dengan pengantar dari Kiri API dan *Web Service*.

2.2.1 Pengantar Kiri API

Pemanfaatan Kiri API adalah menggunakan *Web Service*. Hal ini memungkinkan pengaksesan dimana saja dengan menggunakan koneksi internet. Pemanfaatan Kiri API cukup dengan melakukan *request* dengan parameter dan Kiri akan mengembalikan hasil dalam format JSON. Untuk setiap *request* membutuhkan *API key* yang didapat dengan mendaftar².

2.2.2 Routing Web Service

Routing Web Service merupakan Kiri API yang digunakan untuk mendapatkan langkah perjalanan dari lokasi asal ke lokasi tujuan.

Berikut parameter *request* yang diperlukan berikut penjelasannya:

version	2	Memberitahukan bahwa layanan yang dipakai adalah protokol veris 2
mode	"findroute"	mengintruksikan layanan untuk mencari rute
locale	"en" or "id"	bahasa yang digunakan untuk balasan
start	lat,lng (both are decimal values)	titk awal <i>Latitude</i> dan <i>longitude</i>
finish	lat,lng (both are decimal values)	titik akhir <i>Latitude</i> dan <i>longitude</i>
presentation	"mobile" or "desktop"	Menentukan tipe prensentasi untuk keluaran. Contoh, jika tipe presentasi "mobile", maka link "tel:" akan ditambahkan di hasil.
apikey	16-digit hexadecimals	API key yang digunakan

Berikut format Kiri API *responds*:

Listing 2.42: code *respond* pencarian rute

```

1  {
2      "status": "ok" or "error"
3      "routingresults": [
4          {
5              "steps": [
6                  [
7                      "walk" or "none" or others,
8                      "walk" or vehicle_id or "none",

```

²https://bitbucket.org/projectkiri/kiri_api/wiki/KIRIAPIv2Documentation

```

9 |         ["lat_1,lon_1", "lan_2,lon_2", ... "lat_n,lon_n"],
10 |         "human readable description, dependant on locale",
11 |         URL for ticket booking or null (future)
12 |     ],
13 |     [
14 |         "walk" or "none" or others,
15 |         "walk" or vehicle_id or "none",
16 |         ["lat_1,lon_1", "lan_2,lon_2", ... "lat_n,lon_n"],
17 |         "human readable description, dependant on locale",
18 |         URL for ticket booking or null (future)
19 |     ]
20 | ],
21 | "traveltime": any text string, null if and only if route is not found.
22 | },
23 | {
24 |     "steps": [ ... ],
25 |     "traveltime": "... "
26 | },
27 | {
28 |     "steps": [ ... ],
29 |     "traveltime": "... "
30 | },
31 | ...
32 | ]
33 | }

```

Berikut maksud dari listing 2.1: Ketika pencarian rute sukses dilakukan maka status akan memberitahukan "ok" seperti di baris 2. Selanjutnya setiap langkah dari posisi awal ke posisi tujuan akan ditampung di array dari langkah. Berikut keterangan dari setiap array tersebut:

- Index ke 0 atau baris 7 pada listing 2.1 dapat berisi "walk" atau "none" atau "others". Artinya jika "walk" berarti berjalan kaki, "none" jika rute tidak ditemukan dan "others" berarti menggunakan kendaraan.
- Index ke 1 atau baris 8 pada listing 2.1 merupakan detail dari index 0. Artinya jika index 0 "walk" berarti index 1 harus "walk", "none" berarti index 1 harus "none" dan selain itu menyatakan id kendaraan yang mana bisa dipakai untuk ditampilkan gambarnya.
- Index ke 2 atau baris 9 pada listing 2.1 adalah array string yang berisi jalur dalam format "lat,lon". Maksud dari "lat,lon" disini adalah titik permulaan dan titik selesai.
- Index ke 3 atau bari 10 pada listing 2.1 merupakan berisi bentuk yang akan ditampilkan kepada pengguna. Informasi yang disampaikan dapat berupa:
 - %fromicon = untuk menunjukan ikon "from". Biasanya untuk mode presentasi di perangkat bergerak.
 - %toicon = untuk menunjukan ikon "to". Biasanya untuk mode presentasi di perangkat bergerak.
- Index ke 4 atau bari 11 pada listing 2.1 berisi URL untuk pemesanan tiket jika tersedia. Jika tidak tersedia akan bernilai null.

2.2.3 Web Service Pencarian Lokasi

Merupakan Kiri API yang digunakan untuk mencari lokasi beserta kordinat *latitude* dan *longitude*

Berikut parameter *request* yang diperlukan berikut penjelasanya:

version	2	Memberitahukan bahwa layanan yang dipakai adalah protokol veris 2
mode	"searchplace"	mengintruksikan layanan untuk mencari tempat
region	"cgk" or "bdo" or "sub"	kota yang akan dicari tempatnya
querystring	text apa saja dengan minimum text satu karakter	query string yang akan dicari menggunakan layanan ini
apikey	16-digit hexadecimals	API key yang digunakan

Berikut format Kiri API *responds*:

Listing 2.43: code *respond* pencarian lokasi

```

1 {
2   "status": "ok" or "error"
3   "searchresult": [
4     {
5       "placename": "place name"
6       "location": "lat,lon"
7     },
8     {
9       "placename": "place name"
10      "location": "lat,lon"
11    },
12    ...
13  ]
14  "attributions": [
15    "attribution_1", "attribution_2", ...
16  ]
17 }
```

Berikut maksud dari listing 2.2: Ketika pencarian lokasi sukses dilakukan maka status akan memberitahukan "ok" seperti di baris 2. Selanjutnya akan ditampilkan hasil dari lokasi yang ada beserta atributnya. Berikut keterangan dari format dari pencarian lokasi:

- Searchresult (pada bari 4 sampai 7, 8 sampai 11, dan seterusnya) berisi array dari tempat:
 - placename: nama tempat.
 - location: latitude dan longitude dari tempat.
- Attributions berisi array string yang berisikan atribut tambahan untuk dimunculkan.

2.2.4 Web Service Menemukan Transportasi Terdekat

Merupakan Kiri API yang digunakan untuk menemukan rute transportasi terdekat sesuai titik yang diinginkan pengguna.

Berikut parameter *request* yang diperlukan berikut penjelasannya:

version	2	Memberitahukan bahwa layanan yang dipakai adalah protokol veris 2
mode	"nearbytransports"	mengintruksikan layanan untuk mencari rute transportasi terdekat
start	latitude dan longitude (keduanya menggunakan nilai desimal)	kota yang akan dicari tempatnya
apikey	16-digit hexadecimals	API key yang digunakan

Berikut format Kiri API *responds*:

Listing 2.44: code *respond* menemukan lokasi terdekat

```
1 {  
2   "status": "ok" or "error"  
3   "nearbytransports": [  
4     [  
5       "walk" or "none" or others ,  
6       "walk" or vehicle_id or "none",  
7       text_string ,  
8       decimal value  
9     ],  
10    [  
11      "walk" or "none" or others ,  
12      "walk" or vehicle_id or "none",  
13      text_string ,  
14      decimal value  
15    ],  
16    ...  
17  ]  
18 }
```

Berikut maksud dari listing 2.3: Ketika pencarian rute sukses dilakukan maka status akan memberitahukan "ok" seperti di baris 2. Selanjutnya akan diberikan array yang berisi transportasi terdekat yang diurutkan dari yang terdekat ke yang terjauh. Berikut keterangan dari setiap array tersebut:

- Index ke 0 atau baris 5 pada listing 2.1 dapat berisi "walk" atau "none" atau "others". Artinya jika "walk" berarti berjalan kaki, "none" jika rute tidak ditemukan dan "others" berarti menggunakan kendaraan.
- Index ke 1 atau baris 6 pada listing 2.1 merupakan detail dari index 0. Artinya jika index 0 "walk" berarti index 1 harus "walk", "none" berarti index 1 harus "none" dan selain itu menyatakan id kendaraan yang mana bisa dipakai untuk ditampilkan gambarnya.
- Index ke 2 atau baris 7 pada listing 2.1 berisi nama kendaraan.
- Index ke 3 atau baris 8 pada listing 2.1 berisi jarak dengan satuan kilometer.

DAFTAR REFERENSI

- [1] Manning, Paul *Pro Windows Phone App Development* 2013: Apress.
- [2] Szostak, Tomasz *Windows Phone 8 Application Development Essentials* 2013: PACKT.