

**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

|   |             |
|---|-------------|
| <b>KATA PENGANTAR</b>                                       | <b>xv</b>   |
| <b>DAFTAR ISI</b>   | <b>xvii</b> |
| <b>DAFTAR GAMBAR</b>  | <b>xix</b>  |
| <b>DAFTAR TABEL</b>   | <b>xx</b>   |
| <b>1 PENDAHULUAN</b>  | <b>1</b>    |
| 1.1 Latar Belakang . . . . .                                | 1           |
| 1.2 Rumusan Masalah . . . . .                               | 2           |
| 1.3 Tujuan . . . . .  | 2           |
| 1.4 Batasan Masalah . . . . .                               | 2           |
| 1.5 Ruang Lingkup Masalah . . . . .                         | 3           |
| 1.6 Metode Penelitian . . . . .                             | 3           |
| 1.7 Teknik Pengumpulan Data . . . . .                       | 3           |
| 1.8 Sistematika Penulisan . . . . .                         | 3           |
| <b>2 DASAR TEORI</b>  | <b>5</b>    |
| 2.1 Windows Phone . . . . .                                 | 5           |
| 2.1.1 Lingkungan Kerja . . . . .                            | 5           |
| 2.1.2 XAML . . . . .  | 6           |
| 2.1.3 Kontrol terhadap Ponsel . . . . .                     | 6           |
| 2.1.4 Siklus Hidup Aplikasi . . . . .                       | 9           |
| 2.1.5 Peta di Windows Phone . . . . .                       | 10          |
| 2.1.6 Lokasi . . . . .                                      | 15          |
| 2.1.7 Memanfaatkan Sumber Data . . . . .                    | 17          |
| 2.2 Kiri API . . . . .                                      | 19          |
| 2.2.1 Routing Web Service . . . . .                         | 20          |
| 2.2.2 Web Service Pencarian Lokasi . . . . .                | 21          |
| 2.2.3 Web Service Menemukan Transportasi Terdekat . . . . . | 22          |
| <b>3 ANALISIS</b>   | <b>25</b>   |
| 3.1 Analisis Aplikasi Sejenis . . . . .                     | 25          |
| 3.2 Analisis Aplikasi . . . . .                             | 28          |
| 3.2.1 Kebutuhan Aplikasi . . . . .                          | 28          |
| 3.2.2 Analisis Kontrol yang Dipakai . . . . .               | 28          |
| 3.2.3 Analisis Terhadap Siklus Hidup Aplikasi . . . . .     | 30          |
| 3.2.4 Analisis Peta . . . . .                               | 30          |
| 3.2.5 Kiri API . . . . .                                    | 33          |
| 3.2.6 Diagram Use-Case dan Scenario . . . . .               | 36          |
| 3.2.7 Class Diagram . . . . .                               | 36          |



## DAFTAR GAMBAR

|      |   |    |
|------|---|----|
| 1.1  | Statistik Pengguna Windows Phone . . . . .                          | 2  |
| 2.1  | Hirarki navigasi . . . . .  | 7  |
| 2.2  | TextBlock, TextBox dan PasswordBox . . . . .                        | 8  |
| 2.3  | TextBlock, TextBox dan PasswordBox . . . . .                        | 9  |
| 2.4  | Gambar siklus hidup aplikasi[3] . . . . .                           | 10 |
| 2.5  | Tampilan peta pada Windows Phone . . . . .                          | 11 |
| 2.6  | Kartografi . . . . .  | 11 |
| 2.7  | Keluaran Toolkit Pushpin pada peta [3] . . . . .                    | 12 |
| 3.1  | Tampilan awal aplikasi Public Transport . . . . .                   | 25 |
| 3.2  | Menunjuk lokasi pada peta . . . . .                                 | 26 |
| 3.3  | Memberikan daftar nama tempat dan nama jalan terkait . . . . .      | 26 |
| 3.4  | Tampilan rute kendaraan umum dalam bentuk daftar . . . . .          | 27 |
| 3.5  | Tampilan rute kendaraan umum di peta . . . . .                      | 27 |
| 3.6  | antarmuka TextBlock dan TextBox . . . . .                           | 29 |
| 3.7  | Antarmuka kontrol tombol . . . . .                                  | 29 |
| 3.8  | Antarmuka ListBox . . . . .   | 30 |
| 3.9  | Antarmuka pencari rute kendaraan umum di Windows Phone . . . . .    | 30 |
| 3.10 | Tampilan peta dengan pandangan road dan mode warna terang . . . . . | 31 |
| 3.11 | Tampilan pushpin pada Peta . . . . .                                | 32 |
| 3.12 | Tampilan polyline pada Peta . . . . .                               | 33 |
| 3.13 | Diagram Use Case . . . . .  | 37 |

## DAFTAR TABEL

|     |   |    |
|-----|---|----|
| 2.1 | Properti kelas Map . . . . .                | 13 |
| 2.2 | Properti Kelas Polyline . . . . .           | 14 |
| 2.3 | Kelas pada Namespace Geolocator . . . . .   | 16 |
| 2.4 | Properti pada Kelas Geocoordinate . . . . . | 16 |

# BAB 1

## PENDAHULUAN

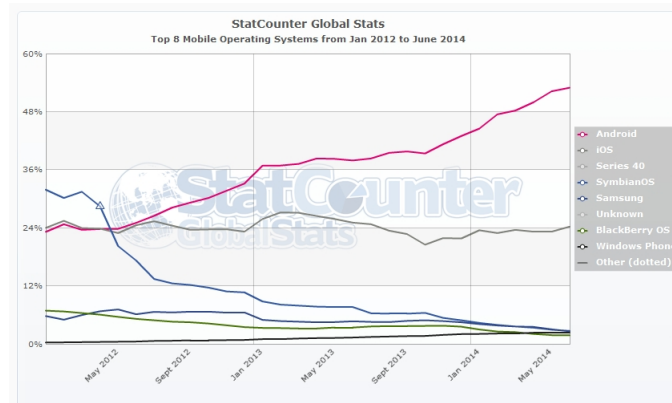
Pada Bab satu akan dibahas pendahuluan dari penelitian yang dilakukan. Bab satu terbagi dalam delapan *sub bab*, yaitu *latar belakang*, *rumusan masalah*, *tujuan*, *batasan masalah*, *ruang lingkup masalah*, *metode penelitian*, *teknik pengumpulan data*, dan *sistematika penulisan*.

### 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. Berikut statistik peningkatan jumlah pengguna di Windows Phone dari tahun 2012 hingga 2014.



Gambar 1.1: Statistik Pengguna Windows Phone

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. Bahasan yang dijelaskan pada bab ini adalah Windows Phone dan Kiri API. Teori Windows Phone yang dijelaskan meliputi lingkungan kerja, xaml, kontrol terhadap ponsel, siklus hidup aplikasi, peta di Windows Phone, lokasi, dan memanfaatkan sumber data. Teori Kiri API yang dijelaskan meliputi *routing web service*, *web service* pencarian lokasi, dan *web service* menemukan transportasi terdekat.

Bab 3 membahas tentang analisi pembangunan aplikasi Pencarian Rute Kendaraan Umum untuk Windows Phone. Analisa yang dibahas mengenai analisa kebutuhan aplikasi, analisa kontrol yang dipakai, analisa terhadap siklus hidup aplikasi, analisa terhadap siklus hidup aplikasi, analisa peta, analisa Kiri API, diagram use case, dan diagram kelas.





## 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 antarmuka Windows Phone, Menampilkan Peta, Penggunaan *Global Positioning System* di Windows Phone, Web Service, dan Kiri API.

#### 2.1 Windows Phone

Windows Phone merupakan sistem operasi untuk perangkat bergerak yang dikembangkan Microsoft.<sup>1</sup> 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# dan Visual Basic.

Pada sub bab 2.2 sampai 2.8 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.

##### 2.1.1 Lingkungan Kerja

Microsoft .NET framework merupakan sebuah perangkat lunak yang dibangun untuk membantu dalam pembangunan aplikasi di Windows, Windows Phone, Windows Server, and Microsoft Azure[1]. Ini terdiri dari runtime bahasa umum dan perpustakaan kelas NET Framework, yang meliputi kelas, interface, dan jenis nilai yang mendukung berbagai teknologi. NET Framework menyediakan lingkungan yang mudah dikelola, pengembangan disederhanakan dan penyebaran, dan integrasi dengan berbagai bahasa pemrograman, termasuk Visual Basic dan Visual C#.

Seperti yang telah disebutkan ada dua bahasa pemrograman dalam .NET Framework yang dipakai dalam pembangunan aplikasi di Windows Phone 8 yaitu Visual Basic dan Visual C#. Untuk masalah kehandalan keduanya menawarkan kehandalan yang baik. Kelebihan dari Visual Basic adalah bahasa pemrograman berorientasi objek yang kuat dan memiliki banyak pengembangan fitur di inheritance, polymorphism, interfaces, and overloading[1]. Kelebihan dari C# yang merupakan pengembangan dari C/C++ adalah sederhana, modern, aman dan berorientasi objek[1]. Satu hal yang dirasakan penulis adalah kenyamanan ketika memilih bahasa .NET tersebut. Akan lebih mudah bagi developer yang menggunakan Visual Basic 6.0 untuk menggunakan Visual Basic .NET.

---

<sup>1</sup>[en.wikipedia.org/wiki/Windows\\_Phone](http://en.wikipedia.org/wiki/Windows_Phone)

Tetapi bagi developer yang menggunakan C++ atau java sebelumnya akan lebih mudah menggunakan C#.

### 2.1.2 XAML

Extensible Application Markup Language (XAML) merupakan bahasa deklaratif yang dipakai untuk membuat antarmuka aplikasi. XAML merupakan bahasa yang digunakan untuk membuat antarmuka di Windows Phone 8. Pada dasarnya penggunaan XAML sama dengan HTML pada pembuatan antarmuka web. XAML dapat menginisialisasi objek dan mengatur properti untuk menunjukkan hubungan antar objek.

Untuk aturan penulisan sintak XAML didasarkan pada XML. Setiap XAML Windows Runtime menggunakan konvensi XAML language dan ditulis pada *namespace* yang ditandai dengan prefix x sebagai elemen paling atas. Setelah itu di baris ke dua dimulai dengan xmlns diikuti titik dua, lalu nama dari *namespace*, diikuti tanda sama dengan dan *path* perepresentasian *namespace*. Prefix x pada XAML mengandung beberapa struktur program yang sering kita gunakan yaitu :

- x:Key : sebuah nama unik untuk menunjuk referensi ke suatu resource atau berkas lain. Nilai ini dapat dipanggil kembali untuk menggunakan resource tersebut.
- x:Class : menunjukkan nama kelas.
- x>Name : menunjukkan nama sebuah obyek dan untuk membedakan antar obyek yang satu dengan obyek yang lain.
- x:Uid : mengidentifikasi elemen objek dalam XAML. Objek elemen merupakan objek yang dapat melakukan kontrol terhadap kelas atau elemen lain yang ditampilkan di desain antarmuka.

### 2.1.3 Kontrol terhadap Ponsel

Kontrol terhadap ponsel yang dimaksudkan disini adalah pengaturan tata letak terhadap antarmuka di Windows Phone. Windows Phone 8 menyediakan banyak set kontrol yaitu tata letak, tombol, kontrol masukan untuk mendapatkan informasi sampai ke menu.

#### 2.1.3.1 Navigasi

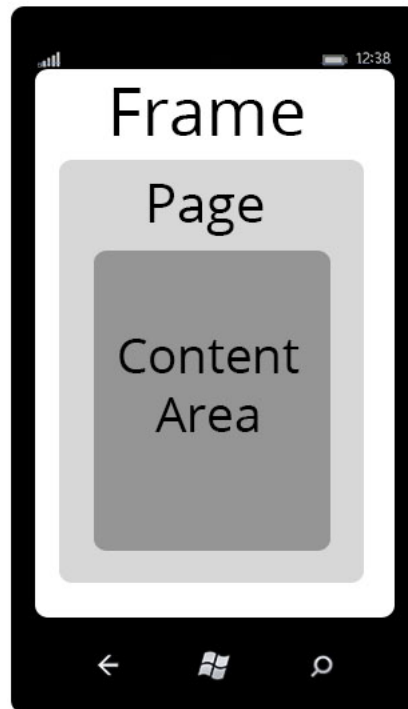
Aplikasi yang dibuat di Windows Phone didasarkan pada model halaman. Maksud dari model halaman adalah pengguna berpindah dari satu halaman ke halaman lain dengan konten yang berbeda-beda dengan frame sebagai pengontrolnya. Setiap antarmuka aplikasi dibungkus dengan frame. Frame inilah yang melakukan kontrol terhadap aplikasi dan memungkinkan berpindah dari satu halaman ke halaman lain. Sedangkan halaman merupakan pembungkus dari elemen di dalamnya saja. Untuk lebih jelas mengenai frame, halaman dan area konten dapat dilihat pada gambar di bawah ini.

---

<sup>2</sup><http://msdn.microsoft.com/en-us/library/vstudio/w0x726c2%28v=vs.110%29>

<sup>3</sup><http://msdn.microsoft.com/en-us/library/aa903378%28v=vs.71%29.aspx>

<sup>4</sup><http://msdn.microsoft.com/en-us/library/aa287558%28v=vs.71%29.aspx>



Gambar 2.1: Hirarki navigasi

### 2.1.3.2 Kontrol Tata Letak

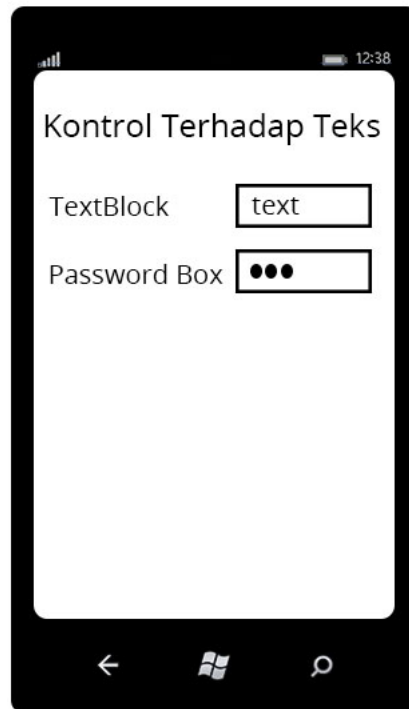
Kontrol Tata Letak merupakan penampung pada antarmuka Windows Phone untuk objek di antarmuka dan kontrol yang lain (tombol radio, textbox, dan lai-lain). Kontrol tata letak digunakan untuk meletakkan objek-objek di layar. Ketika pertama membuat aplikasi Windows Phone maka tata letak dasar sebagai penampung akan langsung dibuat berikut panel judul dan panel konten. Selanjutnya untuk penambahan kontrol tata letak yang lain dapat ditambahkan di panel konten.

Ada 3 macam panel yang dipakai untuk menangani Tata Letak yaitu Grid, StackPane, dan Canvas. Perlu diperhatikan bahwa setiap halaman hanya memiliki satu macam panel. Berikut 3 macam panel di Windows Phone:

- StackPanel merupakan panel yang memposisikan element menjadi 1 baris dan beberapa elemen di setiap halaman diposisikan horizontal atau vertical saja.
- Grid merupakan panel yang mendukung tata letak yang rumit. Panel ini memposisikan elemen di baris dan kolom mana saja di setiap halaman.
- Canvas memposisikan elemen sebagai absolut kordinat. Jadi setiap elemen di dalam Canvas dapat diposisikan spesifik sesuai kordinat x dan y.

### 2.1.3.3 Kontrol Terhadap Teks

Kontrol Terhadap Teks secara menampilkan konten String. Ada berbagai macam Kontrol Terhadap Teks di Windows Phone yaitu TextBlock, TextBox dan PasswordBox. Ketiga macam kontrol tersebut dibedakan menurut tujuannya. Berikut gambar dan keterangan masing-masing:



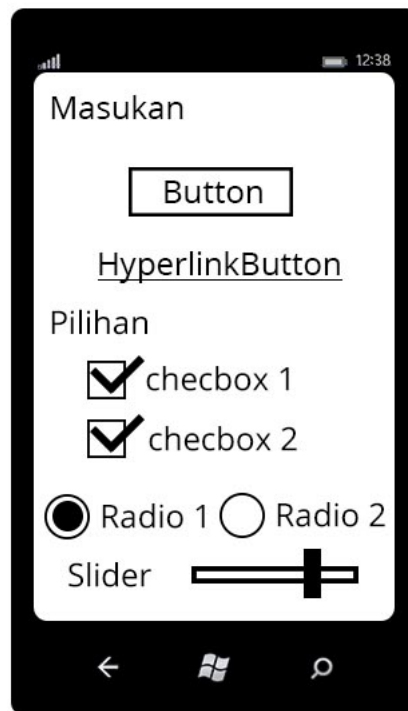
Gambar 2.2: TextBlock, TextBox dan PasswordBox

- TextBlock merupakan tempat menaruh potongan teks yang hanya bisa dilihat.
- TextBox biasanya digunakan untuk teks masukan yang pendek. Tapi bisa juga dipakai untuk masukan yang banyak dan beberapa baris.
- PasswordBox biasanya digunakan untuk masukan yang bersifat rahasia. Karakter yang dimasukan langsung disamarkan menjadi bentuk titik.

#### 2.1.3.4 Tombol dan Kontrol Pilihan

Tombol memungkinkan pengguna untuk bernavigasi. Sedangkan kontrol pilihan memudahkan dalam memilih. Berikut gambar dan keterangan masing-masing:

- Button merupakan kontrol yang dipakai pengguna untuk mengaktifkan *event* klik.
- HyperlinkButton merupakan kontrol yang menampilkan hyperlink. Jika di tekan maka akan menunjuk ke halaman yang dituju.
- CheckBox merupakan kontrol yang memungkinkan pengguna memilih beberapa item.
- RadioButton merupakan kontrol yang memungkinkan pengguna memilih satu pilihan dari beberapa pilihan.
- Slider merupakan kontrol yang memungkinkan user memilih nilai kisaran dari jalur yang sudah disediakan.



Gambar 2.3: TextBlock, TextBox dan PasswordBox

#### 2.1.3.5 Kontrol Daftar

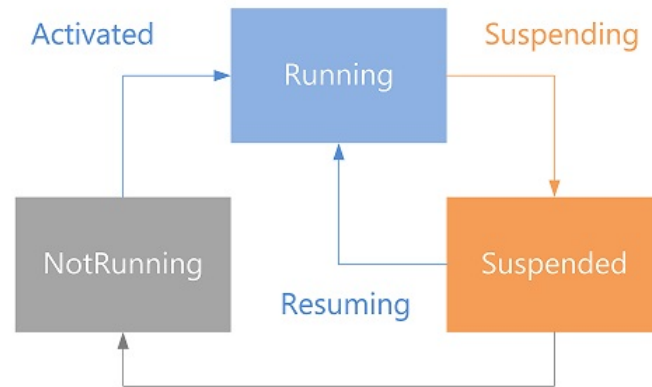
Kontrol yang dipakai untuk menampilkan daftar dari beberapa item. Berikut keterangan masing-masing:

- ListBox akan menampilkan daftar item. Daftar ini dapat dipilih dengan cara di klik.
- LongListSelector dipakai untuk mengelompokan, menampilkan, dan melakukan penggulangan terhadap daftar yang panjang.

#### 2.1.4 Siklus Hidup Aplikasi

Siklus hidup aplikasi merupakan waktu mulai dari aplikasi dijalankan sampai aplikasi dibuang dari memori. Siklus hidup aplikasi penting diketahui agar pengguna tidak kecewa menggunakan aplikasi yang penulis buat. Seringkali pengguna tidak berhati-hati dalam menggunakan aplikasi, makadari itu penulis harus paham kapan aplikasi harus diaktifkan, ditangguhkan atau bahkan dihapus karena sudah tidak digunakan. Berikut gambar beserta keterangan siklus hidup pada Windows Phone.

- Running : Ketika aplikasi diluncurkan, aplikasi membutuhkan *event* Activated | activated maka aplikasi akan masuk keadaan "Running". Pada saat *event* Activated | activated dipanggil *event* ini akan memeriksa apakah aplikasi sudah siap untuk keadaan "Running".
- Suspended : Aplikasi akan masuk keadaan "Suspended" ketika pengguna memilih untuk pindah aplikasi atau perangkat Windows Phone masuk ke keadaan hemat tenaga (misal: layar



Gambar 2.4: Gambar siklus hidup aplikasi[3]

dimatikan sementara). Ketika pengguna berpindah aplikasi atau membuat aplikasi berjalan dibelakang layar untuk sementara waktu dan user tidak kembali ke aplikasi maka sistem operasi akan membuat aplikasi masuk ke keadaan "Suspended".

- Resume : Aplikasi akan masuk keadaan "Resume" ketika pengguna kembali menjalankan aplikasi atau perangkat Windows Phone keluar dari keadaan hemat tenaga (misal: layar dimatikan sementara). Saat keadaan ini semua data masih disimpan di memori jadi tidak akan ada data yang hilang.
- Close : Secara general pengguna tidak usah repot-repot keluar dari aplikasi karena semua sudah diurus sistem operasi. Meskipun pengguna juga dapat keluar dari aplikasi. Setelah aplikasi di keluarkan oleh pengguna dan masuk dalam keadaan "Suspended", selanjutnya aplikasi akan masuk dalam keadaan "NotRunning".

### 2.1.5 Peta di Windows Phone

Peta yang dipakai di Windows Phone adalah Windows Phone Maps. Windows Phone menawarkan beberapa pilihan dalam tampilan peta mulai dari kartografi, pencahayaan dan pandangan. Tidak hanya tampilan pada Sub Bab ini akan dibahas mengenai mendapatkan lokasi, petunjuk arah, MapPolyline dan Pushpins.

#### 2.1.5.1 Penambahan Peta Ke Aplikasi

Untuk penambahan Peta pada Windows Phone menggunakan kontrol peta. Kontrol peta merupakan bagian dari perpustakaan Windows Phone. Dengan begitu untuk dapat menggunakannya perlu direferensikan. Untuk dapat menggunakannya juga harus ditambah *capability* ID\_CAP\_MAP. Selanjutnya barulah peta dapat ditampilkan.



Gambar 2.5: Tampilan peta pada Windows Phone

### 2.1.5.2 Tampilan Peta di Windows Phone

Dalam tampilannya ada beberapa hal yang perlu diperhatikan agar pengguna merasa nyaman saat melihat peta di Windows Phone. Beberapa tampilan yang bisa ditampilkan dibuat untuk hal yang berbeda-beda. Berikut akan dibahas menentukan pusat dan tingkat zoom, kartografi, warna dan tampilan peta.

- Menentukan pusat peta berarti menentukan titik tengah sebagai pandangan awal di peta. Untuk penentuan titik tengah dibutuhkan 2 nilai yaitu latitude dan longitude. Sedangkan tingkat zoom merupakan properti untuk mengatur seberapa dekat atau jauh pandangan yang akan ditampilkan di peta. Tingkat zoom memiliki nilai yang bisa diatur dari satu hingga 20.
- Kartografi peta di Windows Phone merupakan cara pandang dalam melihat dan menerjemahkan peta. Ada beberapa 4 jenis kartografi, yaitu:
  - Road: Tampilan normal 2 dimensi.
  - Aerial: Tampilan peta yang diambil dari foto di udara.
  - Hybrid: Tampilan Aerial yang digabung dengan jalan dan label.
  - Terrain: Menampilkan gambar fisik bumi termasuk ketinggian dan air.



Gambar 2.6: Kartografi

- Mode warna yang disediakan Windows Phone ada 2 yaitu terang dan gelap. Secara bawaan mode pada peta di Windows Phone adalah terang.

- Tampilan pada Peta di Windows Phone dapat berubah karena hasil diputar, dimiringkan, ditarik, dan diturunkan. Berikut beberapa hal yang dapat diatur sebagai tampilan di peta.
  - *Heading* merupakan representasi dari derajat secara geometri. Derajat ini didefinisikan dalam 0 sampai 360 yang dipakai untuk memutar peta. Contoh, 0 atau 360 ke arah utara, 90 ke arah barat, 180 ke arah selatan, dan 270 derajat ke arah timur.
  - *Pitch* merupakan derajat kemiringan dari peta dari sudut pengguna. Contoh, *Pitch* = 0 berarti melihat dari atas ke bawah sedangkan *Pitch* = 45 berarti melihat dari samping ke bawah dengan sudut 45 derajat.

### 2.1.5.3 Pushpins ke Peta

Pushpin merupakan elemen yang dapat ditempatkan pada peta secara spesifik dan bisa dipakai untuk interaksi pada peta. Peta tidak mendukung langsung penggunaan pushpin karena pushpin merupakan elemen MapOverlay (bagian/lapisan terpisah dari peta). Untungnya di Windows Phone memiliki Windows Phone 8 Toolkit yang memiliki set objek agar dapat menggunakan pushpin pada peta di Windows Phone. Contoh keluaran pushpin dapat dilihat pada Figure 2.7 berikut.



Gambar 2.7: Keluaran Toolkit Pushpin pada peta [3]

### 2.1.5.4 Polyline pada Peta

Dalam menentukan arah dibutuhkan 2 titik yaitu titik awal dan titik tujuan. Tentu saja arah tersebut butuh ditandai dengan garis. Polyline merupakan tentetan garis lurus yang saling terhubung satu sama lain. Dengan polyline arah pada peta dapat ditandai dengan warna maupun tebal atau tipisnya garis.

### 2.1.5.5 Namespace Kontrol Map

Namespace merupakan nama yang dipakai untuk mengatur kelas-kelas. Windows Phone 8 sudah menyediakan namespace bawaan untuk mengatur peta. Namespace yang disediakan adalah Maps.Controls. Namespace ini yang berisi kelas-kelas yang paling sering digunakan untuk mengatur peta pada Windwows Phone. Agar dapat menggunakan kelas pada namespace tersebut perlu ditambahkan namespace dan capabilities. Namespace yang harus ditambahkan pada baris awal XAML



adalah `Microsoft.Phone.Maps.Controls`. Selanjutnya ada penambahan capabilities `ID_CAP_MAP`. Penambahan capabilities ditambahkan pada `WMAppManifest.xml`.

### 2.1.5.6 Kelas Map

Merupakan kelas yang mewakili kontrol map.

Berikut properti yang dapat digunakan pada kelas ini.

| Nama                                   | Deskripsi   |
|--|---|
| <code>CartographicMode</code>          | Mengatur dan mendapatkan tipe dari peta.          |
| <code>Center</code>                    | Mengatur dan mendapatkan lokasi tengah pada peta. |
| <code>ColorMode</code>                 | Mengatur dan mendapatkan mode warna peta          |
| <code>Heading</code>                   | Mengatur dan mendapatkan arah pandang peta.       |
| <code>Height</code>                    | Mengatur dan mendapatkan tinggi.                  |
| <code>LandmarksEnabled</code>          | Mengindikasikan apakah bangunan 3D ditampilkan.   |
| <code>Name</code>                      | Mengatur dan mendapatkan nama identifikasi objek. |
| <code>PedestrianFeaturesEnabled</code> | Mengindikasikan fitur pejalan kaki ditampilkan.   |
| <code>Pitch</code>                     | Mengatur dan mendapatkan derajat kemiringan peta. |
| <code>Tag</code>                       | Mengatur dan mendapatkan nilai objek.             |
| <code>TileSources</code>               | Mendapatkan koleksi lapisan lantai.               |
| <code>Width</code>                     | Mengatur dan mendapatkan lebar.                   |
| <code>ZoomLevel</code>                 | Mengatur dan mendapatkan tingkat zoom pada peta.  |

Tabel 2.1: Properti kelas Map

Berikut metode yang dapat digunakan pada kelas ini.

- `SetView(LocationRectangle)`  
Metode untuk mengatur pandangan di atas peta secara spesifik sesuai region geografis. Metode ini tidak mengembalikan nilai.
- `SetView(GeoCoordinate, Double)`  
Metode untuk mengatur pandangan di atas peta secara spesifik sesuai titik tengah dan tingkat zoom. Metode ini tidak mengembalikan nilai.
- `SetView(LocationRectangle, MapAnimationKind)`  
Metode untuk mengatur pandangan di atas peta secara spesifik sesuai region geografis dan animasi. Metode ini tidak mengembalikan nilai.
- `SetView(LocationRectangle, Thickness)`  
Metode untuk mengatur pandangan di atas peta secara spesifik sesuai region geografis dengan batas tertentu. Metode ini tidak mengembalikan nilai.
- `SetView(GeoCoordinate, Double, MapAnimationKind)`  
Metode untuk mengatur pandangan di atas peta secara spesifik sesuai titik tengah, tingkat zoom, dan animasi. Metode ini tidak mengembalikan nilai.
- `SetView(GeoCoordinate, Double, Double)`  
Metode untuk mengatur pandangan di atas peta secara spesifik sesuai titik tengah, tingkat zoom, dan heading. Metode ini tidak mengembalikan nilai.

- `SetView(LocationRectangle, Thickness, MapAnimationKind)`  
Metode untuk mengatur pandangan di atas peta secara spesifik sesuai region geografis dengan batas tertentu, dan animasi. Metode ini tidak mengembalikan nilai.
- `SetView(GeoCoordinate, Double, Double, MapAnimationKind)`  
Metode untuk mengatur pandangan di atas peta secara spesifik sesuai titik tengah, tingkat zoom, heading, dan animasi. Metode ini tidak mengembalikan nilai.
- `SetView(GeoCoordinate, Double, Double, Double)`  
Metode untuk mengatur pandangan di atas peta secara spesifik sesuai titik tengah, tingkat zoom, heading, pitch. Metode ini tidak mengembalikan nilai.
- `SetView(GeoCoordinate, Double, Double, Double, MapAnimationKind)`  
Metode untuk mengatur pandangan di atas peta secara spesifik sesuai titik tengah, tingkat zoom, heading, pitch, dan animasi. Metode ini tidak mengembalikan nilai.
- `UpdateLayout`  
Metode yang memastikan semua posisi objek turunan mengikuti tata letak.

#### 2.1.5.7 Kelas Polyline

Merupakan kelas yang dipakai untuk menggambarkan garis lurus yang saling terhubung. Kelas ini tergabung ke dalam namespace `Microsoft.Phone.Maps.Controls`.

Berikut properti yang dapat digunakan pada kelas ini.

| Nama                         | Deskripsi  |
|------------------------------|--|
| <code>Dispatcher</code>      | Mendapatkan objek yang terkait.  |
| <code>Path</code>            | Mengatur dan mendapatkan kumpulan <code>GeoCoordinates</code> yang membuat polyline. |
| <code>StrokeColor</code>     | Mengatur dan mendapatkan warna garis.  |
| <code>StrokeDashed</code>    | Mengatur dan mendapatkan nilai untuk mengetahui jika polyline dihancurkan.           |
| <code>StrokeThickness</code> | Mengatur dan mendapatkan lebar garis untuk menggambar polyline.                      |

Tabel 2.2: Properti Kelas Polyline

Berikut metode yang dapat digunakan pada kelas ini.

- `CheckAccess`  
Metode yang menentukan bisa atau tidaknya pemanggilan thread untuk mengakses objek ini.
- `ClearValue`  
Metode yang akan membersihkan nilai lokal
- `Finalize`  
Metode yang dipakai untuk melakukan pembersihan pada sumber daya yang tidak terpakai sebelum objek dihancurkan.

### 2.1.5.8 Kelas Pushpin

Merupakan kelas yang dipakai untuk menggambarkan elemen terpisah diatas peta. Meskipun pushpin merupakan bawaan pada peta untuk menunjuk suatu lokasi tetapi pushpin dari peta tidak dapat diubah-ubah. Pushpin pada Windows Phone 8 dapat dibuat sesuai ketertarikan<sup>1</sup>. Namun ada cara lain dengan menambahkan Windows Phone Toolkit. Windows Phone Toolkit mempunyai komponen untuk menggambar pushpin diatas peta.

### 2.1.6 Lokasi

Aplikasi di Windows Phone 8 dapat memanfaatkan lokasi dimana perangkat berada. Aplikasi dapat melacak lokasi sesaat pengguna atau pelacakan selama periode tertentu. Data lokasi perangkat berasal dari berbagai sumber termasuk *Global Positioning System* atau GPS, Wi-Fi, dan seluler. Ada 2 set API berbeda yang dapat dimanfaatkan di Windows Phone yaitu *Runtime Location* API dan *.NET Location* API. Windows Phone *Runtime Location* memiliki keunggulan fitur yang banyak sedangkan *.NET Location* direkomendasikan jika aplikasi ditargetkan pada Windows Phone 7.1 dan Windows Phone 8.

Hal yang perlu diperhatikan dalam menentukan layanan lokasi adalah penangkap GPS, Wi-Fi, dan radio seluler. Perangkat tersebut berfungsi sebagai penyedia data lokasi dengan berbagai tingkat akurasi dan konsumsi daya. Perangkat diatas juga berkomunikasi langsung untuk memutuskan sumber mana yang digunakan untuk menentukan lokasi perangkat berdasarkan ketersediaan data lokasi dan prasyarat yang ditentukan aplikasi. Lapisan diatas penyedia data lokasi tersebut adalah pengelola antarmuka. Aplikasi akan menggunakan antarmuka tersebut untuk memulai dan menghentikan layanan lokasi, mengatur tingkat akurasi, dan menerima data lokasi.

#### 2.1.6.1 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. Untuk menggunakan *Geolocator*, perlu menghidupkan *ID\_CAP\_LOCATION* di *\properties\WMAAppManifest.xml*. Dalam mendapatkan posisi juga perlu diperhatikan status dari GPS karena GPS membutuhkan waktu dari mengaktifkan GPS hingga mendapatkan lokasi pengguna secara akurat. Untuk lebih jelas mengenai status dapat dilihat pada nilai status dibawah ini.

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.

---

<sup>1</sup><http://developer.nokia.com/resources/library/Lumia/change-history/archived-content/maps-and-navigation/guide-to-the-wp8-maps-api.html>

- *NotInitialized* : Data lokasi belum tersedia. Status ini muncul jika aplikasi belum memanggil `GetGeopositionAsync` atau register.
- *NotAvailable* : Jika Windows sensor dan lokasi tidak tersedia.

### 2.1.6.2 Namespace Geolocator

Namespace merupakan nama yang dipakai untuk mengatur kelas-kelas. Windows Phone 8 sudah menyediakan namespace bawaan untuk mengakses lokasi. Namespace yang disediakan adalah namespace `Geolocator`. Namespace ini akan mengakses lokasi geografis dari perangkat dan mendukung pelacakan lokasi dari waktu ke waktu. Agar dapat menggunakan kelas pada namespace tersebut perlu ditambahkan namespace dan *capabilities*. Namespace yang harus ditambahkan pada baris awal XAML adalah **Windows.Device.Geolocator**. Selanjutnya ada penambahan *capabilities* `ID_CAP_LOCATION`. Penambahan *capabilities* ditambahkan pada `WMAppManifest.xml`. Kelas yang diatur oleh namespace geolocator dapat di lihat pada tabel 2.3.

| Kelas         | Deskripsi   |
|---------------|---|
| Geocoordinate | Berisi informasi untuk mengidentifikasi lokasi geografis.               |
| Geolocator    | Mendukung dalam pengaksesan lokasi perangkat.                           |
| Geoposition   | Memberikan data lokasi beserta latitude dan longitude atau data alamat. |

Tabel 2.3: Kelas pada Namespace Geolocator

### 2.1.6.3 Kelas Geocoordinate

Kelas `Geocoordinate` adalah kelas yang menunjukan lokasi sebagai kordinat geografis. Kelas ini hanya menyediakan properti yang hanya bisa dibaca. Kelas ini menyediakan properti yang ditunjukan pada tabel 2.4.

| Properti  | Deskripsi  |
|-----------|--|
| Altitude  | Ketinggian lokasi dalam satuan meter.  |
| Heading   | Arah menghadap perangkat dalam satuan derajat yang relative terhadap mata angin utara. |
| Latitude  | Garis lintang dalam satuan derajat.  |
| Longitude | Garis bujur dalam satuan derajat.  |
| Point     | Lokasi dari <code>Geocoordinate</code> .   |
| Speed     | Kecepatan dalam satuan meter per detik.  |

Tabel 2.4: Properti pada Kelas `Geocoordinate`

### 2.1.6.4 Kelas Geolocator

Kelas `Geolocator` merupakan kelas yang mendukung pengaksesan terhadap lokasi.

Berikut metode yang disediakan kelas `Geolocator`:

- `public IAsyncOperation<Geoposition> GetGeopositionAsync()`  
Operator `await` diatas dimaksudkan untuk meminta posisi lokasi terus menerus sampai selesai dan menunda tugas yang lain.

Metode `GetGeopositionAsync` yang merupakan bawaan Kelas `Geolocator` akan meminta data lokasi dan menanganinya sampai selesai. Kembalian dari metode `GetGeopositionAsync` adalah `Geoposition`.

Berikut Properti yang disediakan kelas `Geolocator`:

- `public PositionStatus LocationStatus get;`  
Merupakan properti dari kelas `geolocator` untuk mendapatkan status posisi dengan mengembalikan kelas `PositionStatus`. Status pada kelas `PositionStatus` adalah `Ready`, `Initializing`, `NoData`, `Disable`, `NotInitialized`, dan `NotAvailable`.
- `public PositionAccuracy DesiredAccuracy get; set;`  
Properti yang digunakan untuk mengatur dan mendapatkan tingkat akurasi. Untuk tingkat akurasi dapat dipilih tingkat *High* untuk tingkat akurasi tinggi dan dipilih tingkat `Default` untuk menghemat daya. Keluaran dari properti ini adalah tipe data `PositionAccuracy`.
- `public Nullable<uint> DesiredAccuracyInMeters get; set;`  
Sama seperti properti `DesiredAccuracy` diatas tetapi dalam satuan meter. Keluaran dari properti ini adalah tipe data `uint`.
- `public uint ReportInterval get; set;`  
Merupakan properti untuk mendapatkan selang waktu pembaruan lokasi. Properti ini mengeluarkan tipe data `uint`.

#### 2.1.6.5 Kelas `Geoposition`

Kelas `Geoposition` merupakan kelas yang memuat lokasi (latitude dan longitude). Berikut Properti yang disediakan kelas `Geoposition`:

- `public CivicAddress CivicAddress get;`  
Data alamat sipil yang terkait dengan lokasi geografis.
- `public Geocoordinate Coordinate get;`  
Data latitude dan longitude yang terkait lokasi geografis.

#### 2.1.7 Memanfaatkan Sumber Data

Hal yang penting dari sebuah aplikasi adalah informasi. `Windows Phone 8` memiliki kemampuan dalam menghubungkan aplikasi dengan sumber data lainnya. Pemanfaatan data ada 2 macam yaitu yang lokal atau berada di perangkat dan `web service`. `Web Service` merupakan metode komunikasi antara 2 perangkat melalui jaringan.

Sebelum data dapat dikirim antar perangkat perlu dilakukan *Serialization*. *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.

#### 2.1.7.1 Kelas `HttpClient`

Merupakan Kelas yang dipakai untuk mengirim permintaan HTTP dan menerima kembalian HTTP dari *Uniform Resource Identifier*(URI) yang dapat diidentifikasi. Berikut metode yang disediakan kelas `HttpClient`.

- `DeleteAsync`

Metode yang dipakai untuk mengirimkan permintaan DELETE ke URI yang spesifik sebagai operasi *asynchronous*. Maksud dari operasi *asynchronous* adalah memungkinkan aplikasi untuk melanjutkan pekerjaan selagi metode ini dipanggil<sup>2</sup>. Metode ini membutuhkan parameter URI sebagai tujuan dari permintaan. Sedangkan kembaliannya berupa objek.

- `GetAsync(Uri)`

Metode yang dipakai untuk mengirimkan permintaan GET ke URI yang spesifik sebagai operasi *asynchronous*. Maksud dari operasi *asynchronous* adalah memungkinkan aplikasi untuk melanjutkan pekerjaan selagi metode ini dipanggil<sup>2</sup>. Metode ini membutuhkan parameter URI sebagai tujuan dari permintaan. Sedangkan kembaliannya berupa objek.

- `GetAsync(Uri,HttpCompletionOption)`

Metode yang dipakai untuk mengirimkan permintaan GET ke URI yang spesifik sebagai operasi *asynchronous*. Maksud dari operasi *asynchronous* adalah memungkinkan aplikasi untuk melanjutkan pekerjaan selagi metode ini dipanggil<sup>2</sup>. Metode ini membutuhkan parameter URI sebagai tujuan dari permintaan dan nilai tambahan yang dimaksudkan sebagai indikasi operasi dianggap selesai. Sedangkan kembaliannya berupa objek.

- `GetBufferAsync`

Metode yang dipakai untuk mengirimkan permintaan GET ke URI yang spesifik sebagai operasi *asynchronous*. Maksud dari operasi *asynchronous* adalah memungkinkan aplikasi untuk melanjutkan pekerjaan selagi metode ini dipanggil<sup>2</sup>. Metode ini membutuhkan parameter URI sebagai tujuan dari permintaan. Sedangkan kembaliannya adalah kembalian yang dikirimkan secara buffer(menyimpan dalam memori).

- `GetInputStreamAsync`

Metode yang dipakai untuk mengirimkan permintaan GET ke URI yang spesifik sebagai operasi *asynchronous*. Maksud dari operasi *asynchronous* adalah memungkinkan aplikasi untuk melanjutkan pekerjaan selagi metode ini dipanggil<sup>2</sup>. Metode ini membutuhkan parameter URI sebagai tujuan dari permintaan. Sedangkan kembaliannya adalah kembalian yang dikirimkan secara stream(langsung sesuai waktu).

- **GetStringAsync**  
Metode yang dipakai untuk mengirimkan permintaan GET ke URI yang spesifik sebagai operasi *asynchronous*. Maksud dari operasi *asynchronous* adalah memungkinkan aplikasi untuk melanjutkan pekerjaan selagi metode ini dipanggil<sup>2</sup>. Metode ini membutuhkan parameter URI sebagai tujuan dari permintaan. Sedangkan kembaliannya adalah kembalian yang dikirimkan dalam bentuk string.
- **PostAsync**  
Metode yang dipakai untuk mengirimkan permintaan POST ke URI yang spesifik sebagai operasi *asynchronous*. Maksud dari operasi *asynchronous* adalah memungkinkan aplikasi untuk melanjutkan pekerjaan selagi metode ini dipanggil<sup>2</sup>. Metode ini membutuhkan parameter URI sebagai tujuan dari permintaan. Sedangkan kembaliannya berupa objek.
- **SendRequestAsync(HttpRequestMessage)**  
Metode yang dipakai untuk mengirimkan permintaan HTTP sebagai operasi *asynchronous*. Maksud dari operasi *asynchronous* adalah memungkinkan aplikasi untuk melanjutkan pekerjaan selagi metode ini dipanggil<sup>2</sup>. Metode ini membutuhkan parameter pesan dari permintaan. Sedangkan kembaliannya berupa objek.
- **SendRequestAsync(HttpRequestMessage, HttpCompletionOption)**  
Metode yang dipakai untuk mengirimkan permintaan HTTP sebagai operasi *asynchronous*. Maksud dari operasi *asynchronous* adalah memungkinkan aplikasi untuk melanjutkan pekerjaan selagi metode ini dipanggil<sup>2</sup>. Metode ini membutuhkan parameter pesan dari permintaan dan nilai tambahan yang dimaksudkan sebagai indikasi operasi dianggap selesai. Sedangkan kembaliannya adalah kembalian yang dikirimkan secara stream(langsung sesuai waktu).

## 2.2 Kiri API

API atau *Application Programming Interface* merupakan aturan yang dikodekan secara spesifik yang dapat digunakan untuk komunikasi antar aplikasi. Jadi API disini memfasilitasi untuk pemanggilan fungsi-fungsi tertentu diluar aplikasi itu sendiri. Pemanfaatan Kiri API adalah JSON atau *JavaScript Object Notation* format.

Pemanfaatan Kiri API dengan melakukan permintaan dengan parameter POST atau GET dan Kiri akan mengembalikan hasil dalam format JSON. Permintaan tersebut dikirimkan ke URL. Ada URL yang disediakan Kiri Api, yaitu:

- <http://preview.kiri.travel/handle.php>  
Merupakan URL untuk uji coba. Untuk kemampuannya juga menurut dokumentasi Kiri API masih tidak stabil.
- <http://kiri.travel/handle.php>  
Merupakan URL produksi. Ini merupakan URL rekomendasi terhadap permintaan.

---

<sup>2</sup><http://msdn.microsoft.com/en-us/library/ms734701%28v=vs.110%29.aspx>

Untuk setiap permintaan membutuhkan *API key* yang didapat dengan mendaftar[2]. Penggunaan API memungkinkan pengaksesan dimana saja dengan menggunakan koneksi internet. Pada Sub Bab dibawah penulis akan membahas beberapa pemanggilan pesan yang terdapat pada Kiri API.

### 2.2.1 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) | titik 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 kembalian dari Kiri API:

Listing 2.1: Kode kembalian 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",
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      "travelttime": any text string , null if and only if route is not found.
22    } ,
23    {
24      "steps": [ ... ],
25      "travelttime": "... "
26    } ,
27    {
28      "steps": [ ... ],
29      "travelttime": "... "
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 baris 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 baris 11 pada listing 2.1 berisi URL untuk pemesanan tiket jika tersedia. Jika tidak tersedia akan bernilai null.

### 2.2.2 Web Service Pencarian Lokasi

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

Berikut parameter *request* yang diperlukan berikut penjelasannya:

|             |   |   |
|-------------|---|---|
| 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 hexadecimal                            | API key yang digunakan  |

Berikut format kembalian dari Kiri API:

Listing 2.2: Kode kembalian 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.

### 2.2.3 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<br>(keduanya menggunakan nilai desimal) | kota yang akan dicari tempatnya                                   |
| apikey  | 16-digit hexadecimals  | API key yang digunakan  |

Berikut format kembalian dari Kiri API:

Listing 2.3: Kode kembalian 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.



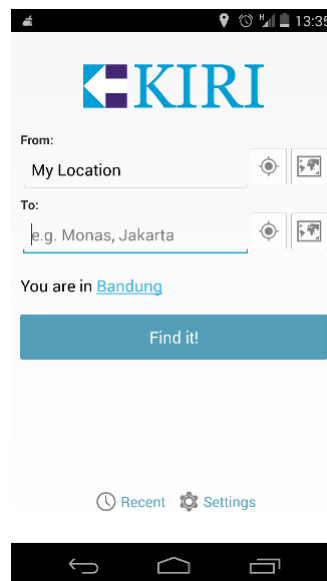
## BAB 3

### ANALISIS

#### 3.1 Analisis Aplikasi Sejenis

Aplikasi sejenis penulis temui bernama "Public Transport"<sup>1</sup>. Namun aplikasi tersebut hanya dapat dijalankan di sistem aplikasi android. Aplikasi "Public Transport" ini memanfaatkan Kiri API. Penggunaannya cukup sederhana. Di halaman awal pengguna dapat mengetikkan lokasi awal dan tujuan. Selain dengan mengetik pengguna juga dapat menunjuk lokasi pada peta. Setelah lokasi dipilih sistem akan memastikan dengan memberi daftar nama jalan dan tempat terkait. Jika sudah memilih maka sistem akan mengeluarkan hasil pencarian rute.

Berikut adalah tampilan dari aplikasi "Public Transport" (Gambar 3.1 sampai 3.5):



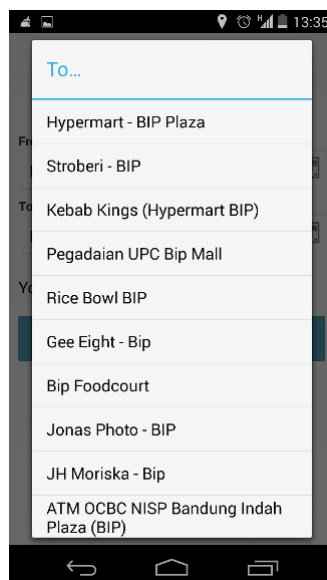
Gambar 3.1: Tampilan awal aplikasi Public Transport

---

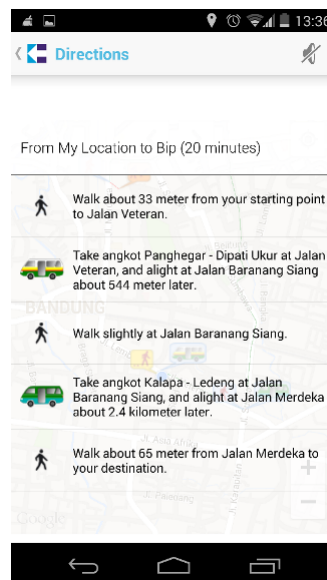
<sup>1</sup><https://play.google.com/store/apps/details?id=travel.kiri.smarttransportapp>



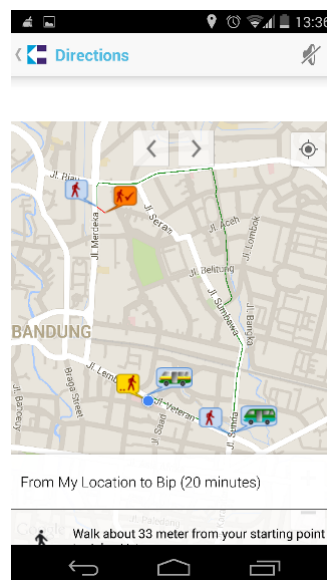
Gambar 3.2: Menunjuk lokasi pada peta



Gambar 3.3: Memberikan daftar nama tempat dan nama jalan terkait



Gambar 3.4: Tampilan rute kendaraan umum dalam bentuk daftar



Gambar 3.5: Tampilan rute kendaraan umum di peta

## 3.2 Analisis Aplikasi

Aplikasi akan dibuat menggunakan bahasa pemrograman C#. Aplikasi yang digunakan untuk membangun Aplikasi Pencari Rute Kendaraan Umum untuk Windows Phone adalah Visual Studio Express 2012. Pada sub bab ini akan dibahas kebutuhan aplikasi, analisa kontrol yang dipakai, analisa terhadap siklus hidup aplikasi, analisa peta, analisa Kiri API, diagram use case, dan diagram kelas dari aplikasi yang akan dibangun.

### 3.2.1 Kebutuhan Aplikasi

Dari hasil observasi penulis dalam menentukan lokasi asal dan lokasi tujuan ada 2 cara. 2 cara tersebut yaitu dengan menulis alamat atau tempat dan dengan menunjuk pada peta. Cara menuliskan alamat atau tempat yaitu dengan menuliskan alamat atau tempat pada tempat yang disediakan untuk asal dan tujuan. Cara menunjuk pada peta yaitu dengan mengetuk layar di posisi yang diinginkan. Kedua hal tersebut pada dasarnya sama saja tetapi ada faktor kemudahan pengguna dalam pemakaiannya. Jadi penulis menyediakan 2 cara tersebut pada aplikasi yang penulis buat agar pengguna dapat memilih salah satunya.

Pada saat menuliskan lokasi atau tempat atau menunjuk langsung pada peta mungkin saja terjadi kesalahan. Kesalahan tersebut bisa saja salah pengetikan atau nama tempat yang tidak ada. Maka dari itu dibutuhkan pemeriksaan terhadap masukan pengguna. Pemeriksaan tersebut dilakukan setelah pengguna memulai pencarian dengan menekan tombol "FIND".

Untuk hasil keluaran ada 2 tipe seperti aplikasi peta lainnya. Kedua tipe tersebut adalah bentuk daftar dan bentuk peta. Bentuk daftar memudahkan dalam melihat tiap langkah rute. bentuk daftar memudahkan pengguna dalam melihat arah dan posisi lingkungan pada rute yang dipilih.

Aplikasi yang penulis bangun didasarkan pada kebutuhan adalah sebagai berikut:

1. Pengguna dapat memasukan lokasi asal dan lokasi tujuan pada bidang yang disediakan atau menunjuk langsung lokasi pada peta.
2. Mendapatkan lokasi terkait menurut lokasi yang dimasukan pengguna.
3. Menampilkan hasil rute angkutan umum dari lokasi asal ke lokasi tujuan.

### 3.2.2 Analisis Kontrol yang Dipakai

Dari kebutuhan yang telah disebutkan diatas penulis menyadari pentingnya kontrol yang harus dipakai. Kontrol yang dimaksud termasuk tata letak, teks, pilihan, dan daftar. Kebutuhan akan kontrol penting bukan hanya untuk kebutuhan tapi memudahkan pengguna.

Untuk kontrol tata letak penulis membayangkan pengaturan yang tertata rapih dan beberapa elemen dalam satu baris atau kolom. Tetapi juga penulis tidak mengharapkan penggunaan kontrol tata letak yang rumit. Dari hasil pengamatan penulis kontrol tata letak yang cocok adalah Grid. Kontrol tata letak ini penulis pilih karena mudah diposisikan sesuai baris dan kolom. Selain itu tampilan Grid akan menyesuaikan jika layar diputar dari posisi pemandangan ke posisi potret dan sebaliknya. Berikut kode untuk membuat tata letak pada halaman Windows Phone menjadi tata letak Grid.



Listing 3.1: Tata letak Grid

```

1| <Grid x:Name="ContentPanel">
2| </Grid>

```

Kontrol terhadap teks juga diperlukan untuk aplikasi ini. Kebutuhan yang diperlukan adalah mengeluarkan potongan teks yang dapat dibaca dan tempat pengguna memasukan teks. Untuk mengeluarkan teks yang dapat dilihat penulis akan menggunakan TextBlock. untuk masukan pengguna terhadap aplikasi penulis akan menyediakan TextBox sebagai tempat teks. Berikut kode dan hasil yang ditampilkan pada Windows Phone.

Listing 3.2: Kode untuk menampilkan kontrol terhadap teks

```

1| <TextBlock Text="TextBlock"/>
2| <TextBox Text="TextBox"/>

```



Gambar 3.6: antarmuka TextBlock dan TextBox

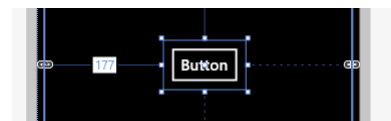
Suatu aplikasi tentunya tidak hanya mempunyai satu halaman. Sama hal dengan aplikasi yang penulis buat memiliki beberapa halaman yang mempunyai tugas berbeda. Karena hal tersebut dibutuhkan kontrol untuk berpindah dari satu halaman ke halaman lain. Kontrol yang dibutuhkan yaitu kontrol tombol. Kontrol tombol akan mengeksekusi *event click* yang memungkinkan pindah halaman dan melakukan perintah.

Listing 3.3: Kode untuk menampilkan kontrol tombol

```

1| <Button Content="Button"/>

```



Gambar 3.7: Antarmuka kontrol tombol

Untuk menampilkan rute salah satu cara yang penulis lakukan adalah menampilkan dalam bentuk daftar. Bentuk daftar yang dapat dipakai di Windows Phone adalah menggunakan ListBox. ListBox akan menampilkan daftar rute satu per satu menurun ke bawah.

Listing 3.4: Kode untuk menampilkan listBox

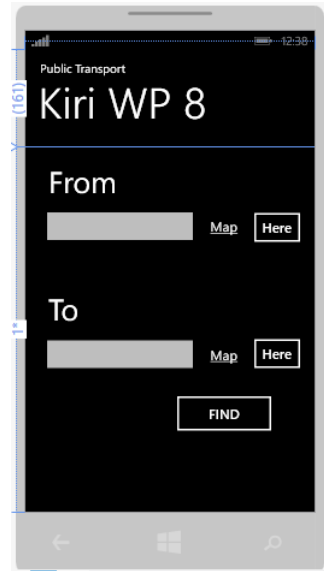
```

1| <ListBox>
2|   <ListBoxItem Content="Item 1" />
3|   <ListBoxItem Content="Item 2" />
4|   <ListBoxItem Content="Item 3" />
5|   <ListBoxItem Content="Item 4" />
6|   <ListBoxItem Content="Item 5" />
7| </ListBox>

```



Gambar 3.8: Antarmuka ListBox



Gambar 3.9: Antarmuka pencari rute kendaraan umum di Windows Phone

### 3.2.3 Analisis Terhadap Siklus Hidup Aplikasi

Aplikasi pada Windows Phone memiliki siklus hidup yang dijelaskan pada bab 2. Pengaturan aplikasi ini diatur sesuai konfigurasi awal sistem operasi Windows Phone. Tetapi pengaturan ini dapat diatur sesuai kebutuhan aplikasi. Karena di aplikasi ini terdapat keadaan yang berbeda dengan konfigurasi awal sistem operasi Windows Phone maka perlu adanya pengaturan ulang.

Saat aplikasi dijalankan, pengguna memasukkan lokasi asal dan lokasi tujuan dimasukkan sampai pencarian rute dilakukan aplikasi akan berada di keadaan "Running". Tetapi ada kemungkinan pengguna berpindah aplikasi atau mematikan layar untuk menghemat daya maka aplikasi akan masuk ke keadaan "Suspended" dan setelah beberapa saat akan masuk ke keadaan "NotRunning". Jika dalam keadaan "Suspended" atau "NotRunning" maka begitu aplikasi dibuka kembali keadaan aplikasi akan sama seperti keadaan terakhir sebelum keluar. Tetapi pada pemakaian oleh pengguna ada kemungkinan pengguna terlalu banyak mengakses aplikasi lain terlalu banyak dan mematikan layar terlalu lama maka aplikasi akan masuk ke keadaan "Close". Untuk menghindari hal pengguna mematikan layar terlalu lama maka keadaan "Suspended" atau "NotRunning" harus diatur agar saat melihat peta dan mematikan layar aplikasi tidak masuk ke keadaan "Close".

### 3.2.4 Analisis Peta

Untuk tampilan peta ada beberapa aspek yang perlu diperhatikan untuk memudahkan pengguna. Aspek yang perlu diperhatikan adalah tingkat zoom, pandangan terhadap peta atau kartografi, dan mode warna. Setelah itu karena tampilan rute pada hasil pencarian dapat dimasukkan gambar angkutan umum maka diperlukan penggunaan pushpin. Sesuai keluaran Kiri API yang mengelu-

arkan titik lokasi maka diperlukan teknik penggambaran rute yaitu menggunakan polyline.

Untuk cara pandang peta terdapat 4 pandangan yang disediakan Peta Windows Phone yaitu Road, Aerial, Hybrid, dan Terrain. Aplikasi ini adalah aplikasi pencari rute dan pandangan lebih banyak diarahkan ke jalan. Kebutuhan pengguna adalah nama jalan, kondisi jalan, dan kondisi sekitar. Untuk dasar pandangan tersebut pandangan yang penulis pilih untuk aplikasi ini adalah Road. Tambahan setelah mengatur pandangan peta yaitu mengatur warna dan penulis akan menggunakan mode warna terang sesuai bawaan. Berikut kode menampilkan peta dengan pandangan Road dan mode warna terang.

Listing 3.5: Menampilkan peta dengan nama MyMap dari XAML

```
1 | <Controls:Map x:Name="MyMap"/>
```

Listing 3.6: Menampilkan peta dengan nama MyMap dari kode program

```
1 | public mapFrom()
2 | {
3 |     InitializeComponent();
4 |     Map MyMap = new Map();
5 |     ContentPanel.Children.Add(MyMap);
6 | }
```



Gambar 3.10: Tampilan peta dengan pandangan road dan mode warna terang

Tampilan awal peta di Windows Phone akan mengeluarkan peta dengan pandangan dunia. Karena aplikasi pencarian rute ini masih terbatas di Pulau Jawa, Indonesia terutama Jawa Barat maka tingkat zoom harus diatur agar mengikuti lokasi pengguna dan di satu daerah saja. Jika pengguna berada di daerah Bandung maka tingkat zoom pada peta disesuaikan pada daerah tersebut. Tingkat zoom dapat diatur dari kode dan XAML. Berikut cara mengatur tingkat zoom.

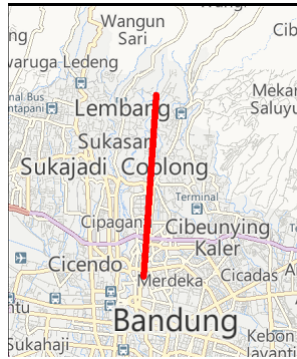
Listing 3.7: Mengatur tingkat zoom dari XAML

```
1 | <Controls:Map x:Name="MyMap" ZoomLevel="10" Margin="-25,0,-16,0"/>
```

Listing 3.8: Mengatur tingkat zoom dari dari kode program

```
1 | public mapFrom()
2 | {
3 |     InitializeComponent();
4 |     Map MyMap = new Map();
5 |
6 |     //Mengatur titik tengah peta
7 |     MyMap.Center = new GeoCoordinate(47.6097, -122.3331);
8 |
9 |     //mengatur tingkat zoom
```





Gambar 3.12: Tampilan polyline pada Peta

### 3.2.5 Kiri API

Kiri API menyediakan 2 parameter untuk permintaan yaitu POST dan GET. Parameter GET berarti data dikirim melalui *Uniform resource locator* atau disingkat URL. Sedangkan permintaan POST akan menyertakan data pada baris permintaan. Selanjutnya permintaan tersebut akan dikirim ke URL. Untuk hal ini penulis akan mengirim ke URL <http://kiri.travel/handle.php>.

Untuk setiap permintaan terhadap Kiri API dibutuhkan *API key*. Kegunaan *API key* adalah password untuk mengakses Kiri API. *API key* dapat didapatkan di [dev.kiri.travel](http://dev.kiri.travel). Berikut langkah-langkah dari membuka [dev.kiri.travel](http://dev.kiri.travel) sampai penulis mendapatkan *API key*.

- Masuk ke situs [dev.kiri.travel](http://dev.kiri.travel).
- Register dengan memasukkan alamat email, nama, dan nama perusahaan.
- Password akan dikirimkan ke alamat email. Tentunya password akan dibuat otomatis oleh pihak Kiri.
- Login setelah mengetahui password yang diterima dari alamat email.
- Setelah berhasil login, di menu utama terdapat API Keys Managements yang dapat dipilih.
- Pilih tombol Add lalu masukan deskripsi penggunaan *API key* yaitu untuk Skripsi.
- *API key* didapat dan dapat digunakan.

Untuk tugas akhir ini penulis akan menggunakan 2 layanan yang ada pada Kiri API. Layanan yang digunakan adalah pencarian lokasi dan routing. Pencarian lokasi adalah layanan untuk menemukan tempat atau nama jalan yang terkait dengan masukan pengguna. Routing adalah layanan untuk menemukan langkah yang harus ditempuh pengguna untuk sampai ke lokasi tujuan dari lokasi asal.

Pemanfaatan layanan pencarian lokasi yaitu dengan parameter GET melalui protokol HTTP. Berikut parameter yang harus dikirimkan:

- version: 2
- mode: "searchplace"
- region: "cgk" untuk Jakarta, "bdo" untuk Bandung, dan "sub" untuk Surabaya

- `querystring`: merupakan kata kunci lokasi yang pengguna tuju
- `apikey`: 16 digit hexadecimal

Format layanan yang dikirim melalui URL adalah `kiri.travel/handle.php?version=2&mode=searchplace&region=cgk/bdo/sub&querystring="string"&apikey=97A7A1157A05ED6F`.

Penulis mencoba mencari lokasi bip dari kata kunci "bip" yang berada di bandung. Layanan dikirimkan ke URL `kiri.travel/handle.php`. Berikut format layanan yang penulis kirim:

`http://kiri.travel/handle.php?version=2&mode=searchplace&region=bdo&querystring=bip&apikey=97A7A1157A05ED6F`

Berikut hasil kembalian dari Kiri API:

Listing 3.11: code *respond* pencarian rute

```

1 {
2   "status": "ok",
3   "searchresult": [
4     {
5       "placename": "Hypermart - BIP Plaza",
6       "location": "-6.90864,107.61108"
7     },
8     {
9       "placename": "Stroberi - BIP",
10      "location": "-6.90834,107.61115"
11    },
12    {
13      "placename": "Kebab Kings (Hypermart BIP)",
14      "location": "-6.91503,107.61017"
15    },
16    {
17      "placename": "Pegadaian UPC Bip Mall",
18      "location": "-6.90916,107.61052"
19    },
20    {
21      "placename": "Rice Bowl BIP",
22      "location": "-6.90873,107.61088"
23    },
24    {
25      "placename": "Gee Eight - Bip",
26      "location": "-6.90817,107.61080"
27    },
28    {
29      "placename": "Jonas Photo - BIP",
30      "location": "-6.91066,107.61016"
31    },
32    {
33      "placename": "Bip Foodcourt",
34      "location": "-6.91081,107.61015"
35    },
36    {
37      "placename": "Mister Baso BIP",
38      "location": "-6.90348,107.61709"
39    },
40    {
41      "placename": "JH Moriska - Bip",
42      "location": "-6.90868,107.61070"
43    }
44  ],
45  "attributions": null
46 }
```

Pemanfaatan layanan routing untuk mendapatkan langkah yang harus ditempuh pengguna untuk mencapai lokasi tujuan dari lokasi asal. Pemanfaatan layanan ini yaitu dengan parameter GET melalui protokol HTTP. Berikut parameter yang harus dikirim:

- `version`: 2
- `mode`: "findroute"

- locale: "en" untuk bahasa Inggris dan "id" untuk bahasa Indonesia.
- start: koordinat lokasi awal dalam berupa latitude dan longitude.
- finish: koordinat lokasi tujuan dalam berupa latitude dan longitude.
- presentation: "mobile" untuk perangkat bergerak dan "desktop" untuk komputer.
- apikey: 16 digit hexadesimal

Format layanan yang dikirim melalui URL adalah [kiri.travel/handle.php?version=2&mode=findroute&locale=en/id&start=lat,lng&finish=lat,lng&presentation=mobile/desktop&apikey=97A7A1157A05ED6](http://kiri.travel/handle.php?version=2&mode=findroute&locale=en/id&start=lat,lng&finish=lat,lng&presentation=mobile/desktop&apikey=97A7A1157A05ED6)

Penulis mencoba menuju jalan merdeka dari jalan ciumbuleuit. Layanan dikirimkan ke URL [kiri.travel/handle.php](http://kiri.travel/handle.php?version=2&mode=findroute&locale=en&start=-6.8747337,107.6048829&finish=-6.9114646,107.6104887&presentation=mobile&apikey=97A7A1157A05ED6F). Berikut format layanan yang penulis kirim <http://kiri.travel/handle.php?version=2&mode=findroute&locale=en&start=-6.8747337,107.6048829&finish=-6.9114646,107.6104887&presentation=mobile&apikey=97A7A1157A05ED6F>.

Berikut hasil kembalian dari Kiri API:

Listing 3.12: code *respond* routing

```

1 {
2   "status": "ok",
3   "routingresults": [
4     {
5       "steps": [
6         [
7           "walk",
8           "walk",
9           ["-6.8747337,107.6048829", "-6.87445,107.60465"],
10          "Walk about 41 meter from your starting point \\\fromicon to Jalan Ciumbuleuit \\\toicon",
11          null
12        ],
13        [
14          "angkot",
15          "ciumbuleuitsthalllurus",
16          ["-6.87445,107.60465", "-6.87541,107.60443", "-6.87637,107.60421", "-6.87734,107.60400",
17           "-6.87830,107.60378", "-6.87926,107.60356", "-6.87926,107.60356", "-6.87963,107.60352",
18           "-6.87978,107.60352", "-6.88093,107.60392", "-6.88209,107.60433", "-6.88209,107.60433",
19           "-6.88328,107.60490", "-6.88328,107.60490", "-6.88347,107.60481", "-6.88452,107.60459",
20           "-6.88556,107.60436", "-6.88660,107.60413", "-6.88764,107.60390", "-6.88764,107.60391",
21           "-6.88782,107.60392", "-6.88887,107.60404", "-6.88991,107.60416", "-6.88991,107.60416",
22           "-6.89161,107.60428", "-6.89161,107.60428", "-6.89166,107.60421", "-6.89275,107.60424",
23           "-6.89275,107.60424", "-6.89405,107.60408", "-6.89405,107.60408", "-6.89496,107.60400"],
24          "Take angkot Ciumbuleuit - St. Hall (lurus) at Jalan Ciumbuleuit \\\fromicon, and
           alight at Jalan Cihampelas
25          \\\toicon about 2.3 kilometer later.",
26          null
27        ],
28        [
29          "angkot",
30          "kalapaledeng",
31          ["-6.89501,107.60403", "-6.89562,107.60398", "-6.89623,107.60395", "-6.89732,107.60401",
32           "-6.89732,107.60401", "-6.89882,107.60414", "-6.89882,107.60414", "-6.89969,107.60418",
33           "-6.90071,107.60426", "-6.90173,107.60433", "-6.90173,107.60433", "-6.90297,107.60437",
34           "-6.90420,107.60440", "-6.90420,107.60440", "-6.90426,107.60456", "-6.90422,107.60481",
35           "-6.90399,107.60546", "-6.90406,107.60617", "-6.90454,107.60697", "-6.90454,107.60697",
36           "-6.90512,107.60745", "-6.90618,107.60778", "-6.90618,107.60778", "-6.90643,107.60787",
37           "-6.90651,107.60807", "-6.90675,107.60914", "-6.90675,107.60914", "-6.90694,107.60939",
38           "-6.90723,107.60939", "-6.90891,107.60943", "-6.90891,107.60943", "-6.90909,107.60934",
39           "-6.90914,107.60857", "-6.90933,107.60846", "-6.91021,107.60887", "-6.91021,107.60887",
40           "-6.91030,107.60897", "-6.91028,107.60927", "-6.90986,107.61040", "-6.90986,107.61040"],
41          "Take angkot Kalapa - Ledeng at Jalan Cihampelas \\\fromicon, and alight at Jalan Aceh
42          \\\toicon about 2.3 kilometer later.",
43          null
44        ],
45        [
46          "walk",

```

```

47 |         "walk",
48 |         ["-6.90986,107.61040",-6.9114646,107.6104887"],
49 |         "Walk about 178 meter from Jalan Aceh \\\%fromicon to your destination \\\%toicon.",
50 |         null
51 |     ],
52 |     ],
53 |     "traveltime":"30 minutes"
54 | }
55 |
56 | }

```

### 3.2.6 Diagram Use-Case dan Scenario

Diagram use-case adalah diagram yang menjelaskan interaksi sistem dengan lingkungan (contoh: pengguna). Berdasarkan analisa di atas maka pengguna dapat:

- Mendapatkan lokasi pengguna berada.
- Memasukan lokasi asal.
- Memasukan lokasi lokasi tujuan.
- Menunjuk langsung lokasi asal dan tujuan pada peta.
- Memilih alamat atau tempat dari pilihan yang disediakan.
- Melihat rute kendaraan umum dalam bentuk titik dan *pushpin* pada peta atau bentuk daftar dari tempat asal ke tempat tujuan.

Berikut adalah diagram use case saat pengguna mencari rute kendaraan umum (Gambar: 3.13):

**Skenario melakukan pencarian rute kendaraan umum** Nama: Mencari rute kendaraan umum Aktor: Pengguna Kondisi Awal: Perangkat lunak dijalankan dan pengguna tidak tahu harus memakai kendaraan umum apa Deskripsi: Pengguna memasukan lokasi awal dan lokasi tujuan Kondisi akhir: Aplikasi memberitahukan kendaraan umum yang harus dinaiki pengguna. Skenario:

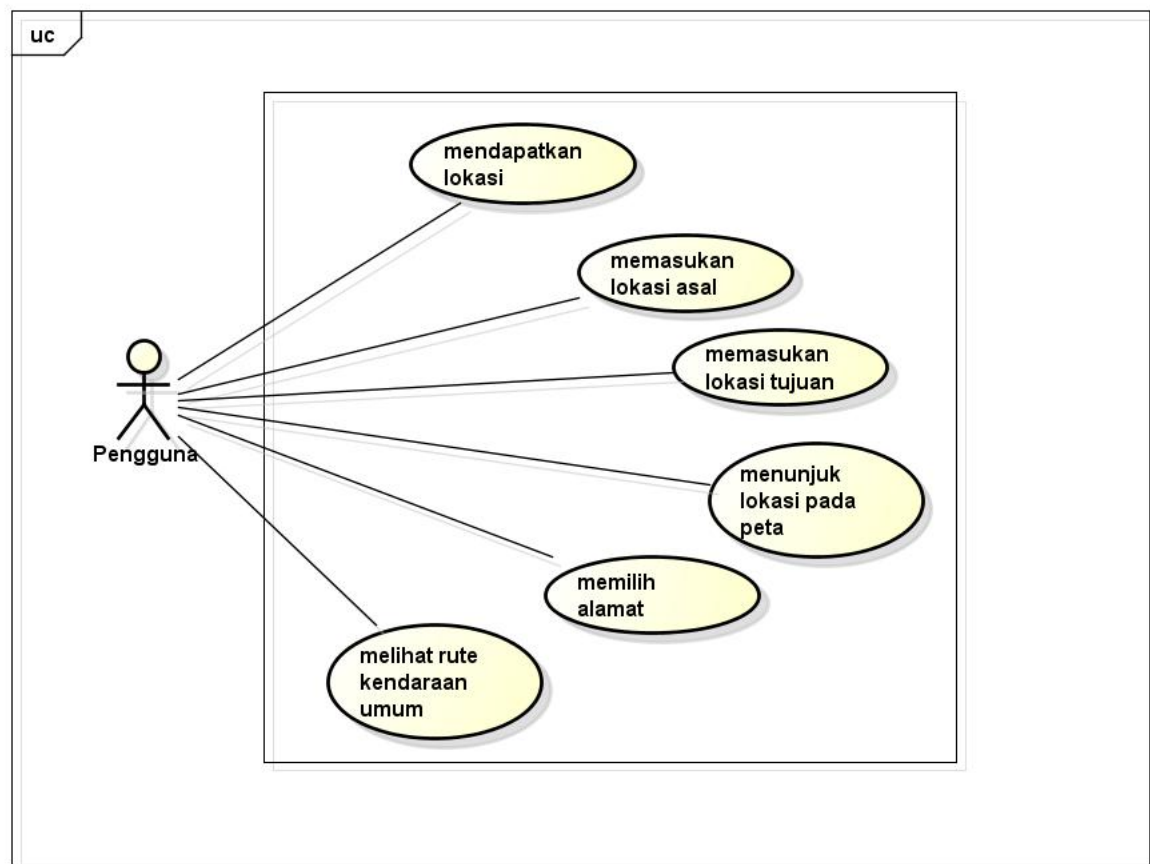
1. Pengguna memasukan lokasi awal dan tujuan atau menunjuk langsung pada peta.
2. Sistem memberikan daftar tempat atau jalan terkait.
3. Pengguna memilih dari daftar tempat atau jalan terkait.
4. Sistem menentukan rute terbaik.
5. Sistem menampilkan rute dalam 2 bentuk yaitu daftar dan titik pada peta.

Eksepsi:

1. Pengguna memasukan lokasi yang tidak terdaftar di sistem.
2. Sistem memberi notifikasi bahwa lokasi tidak ditemukan.

### 3.2.7 Class Diagram





powered by Astah

Gambar 3.13: Diagram Use Case



## DAFTAR REFERENSI

- [1] Microsoft *Windows Phone Silverlight development* 2014 : <http://msdn.microsoft.com/library/windows/apps/ff402535.aspx>.
- [2] Kiri Team *KIRI API v2 Documentation* 2014 : [https://bitbucket.org/projectkiri/kiri\\_api/wiki/KIRI%20API%20v2%20Documentation](https://bitbucket.org/projectkiri/kiri_api/wiki/KIRI%20API%20v2%20Documentation).
- [3] Manning, Paul *Pro Windows Phone App Development* 2013: Apress.
- [4] Szostak, Tomasz *Windows Phone 8 Application Development Essentials* 2013: PACKT.