

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**

DAFTAR ISI

DAFTAR ISI	ii
DAFTAR GAMBAR	iii
DAFTAR TABEL	iv
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	3
1.6 Metode Penelitian	3
1.7 Teknik Pengumpulan Data	3
1.8 Sistematika Penulisan	3
2 DASAR TEORI	5
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 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	20
2.2.4 Web Service Menemukan Transportasi Terdekat	20
BIBLIOGRAFI	23

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	10
2.5	Tampilan Peta pada Windows Phone	11
2.6	Kartografi	11
2.7	Keluaran Toolkit Pushpin pada Peta	12

DAFTAR TABEL

BAB 1

PENDAHULUAN

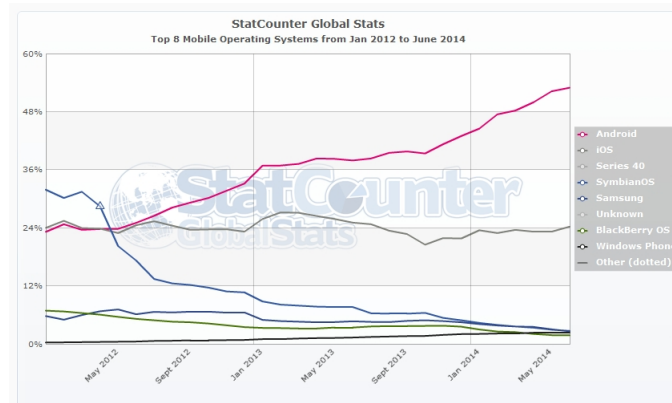
Pada Bab satu akan dibahas pendahuluan dari penelitian yang dilakukan. Bab satu terbagi dalam delapan *sub bab*, yaitu *latar belakang*, *rumusan masalah*, *tujuan*, *batasan masalah*, *ruang lingkup masalah*, *metode penelitian*, *teknik pengumpulan data*, dan *sistematika penulisan*.

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

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# dan Visual Basic.

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 Azure². 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 overloading³. Kelebihan dari C# yang merupakan pengembangan dari C/C++ adalah sederhana, modern, aman dan berorientasi objek⁴. 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. Tetapi

¹en.wikipedia.org/wiki/Windows_Phone

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* representasi *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 objek dan untuk membedakan antar objek yang satu dengan objek 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.

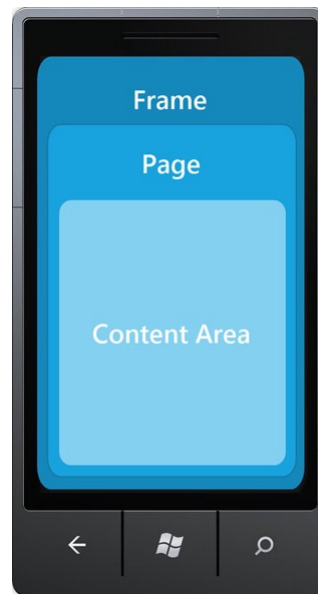
2.1.3.1 Navigasi

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.

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

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

⁴<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.

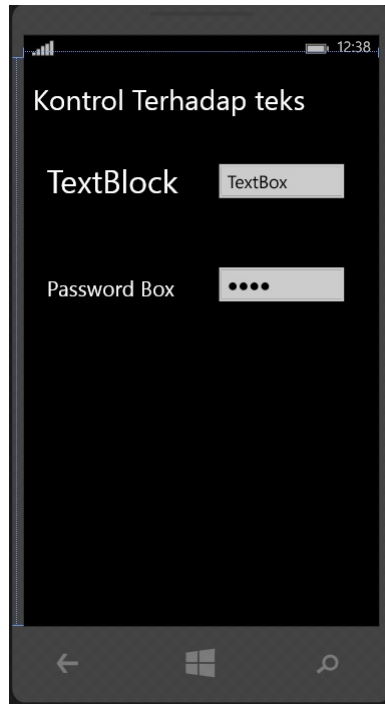
Ada 3 macam panel yang dipakai untuk menangani Tata Letak yaitu Grid, StackPane, dan Canvas. Perlu diperhatikan bahwa setiap halaman hanya memiliki satu macam panel. Berikut 3 macam panel di Windows Phone:

- 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:

- 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.



Gambar 2.2: TextBlock, TextBox dan PasswordBox

- PasswordBox biasanya digunakan untuk masukan yang bersifat rahasia. Karakter yang dimasukan langsung disamarkan menjadi bentuk titik.

2.1.3.4 Tombol dan Kontrol Pilihan

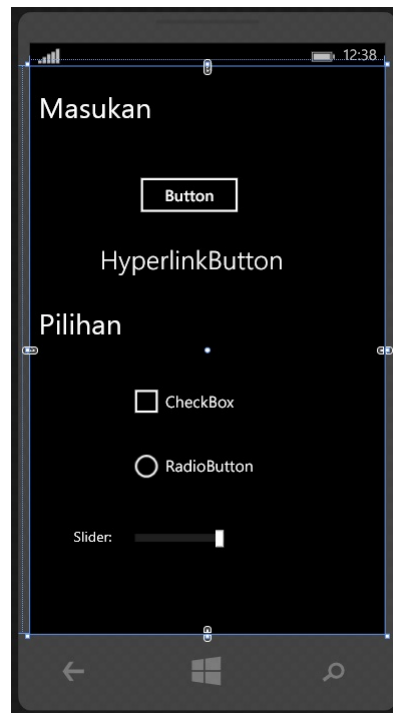
Tombol memungkinkan pengguna untuk bernavigasi. Sedangkan kontrol pilihan memudahkan dalam memilih. Berikut gambar dan keterangan masing-masing:

- 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.

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.



Gambar 2.3: TextBlock, TextBox dan PasswordBox

- LongListSelector dipakai untuk mengelompokan, menampilkan, dan melakukan penggulungan terhadap daftar yang panjang.

2.1.3.6 Gambar, Peta, dan Media

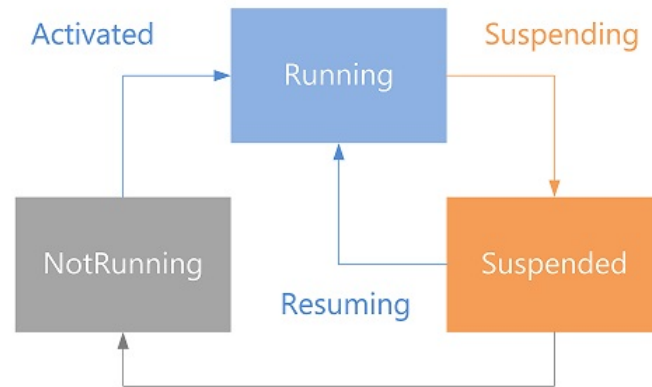
Menampilkan gambar, map, dan konten media penting sebagai bagian dari antarmuka. Berikut keterangan masing-masing:

- Image dipakai untuk menampilkan gambar. Aplikasi di Windows Phone mendukung format jpeg dan png.
- Map dipakai menampilkan peta.
- MediaElement dipakai untuk memainkan audio dan video.

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".



Gambar 2.4: Gambar Siklus Hidup Aplikasi

- **Suspended** : Aplikasi akan masuk keadaan "Suspended" ketika pengguna memilih untuk pindah aplikasi atau perangkat Windows Phone masuk ke keadaan hemat tenaga (misal: layar 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 dikeluarkan 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

2.1.5.4 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. 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.
- *NotInitialized* : Data lokasi belum tersedia. Status ini muncul jika aplikasi belum memanggil `GetGeopositionAsync` atau `register`.
- *NotAvailable* : Jika Windows sensor dan lokasi tidak tersedia.

2.1.5.5 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.

2.1.6 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.

2.2 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

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 *request* dengan parameter POST atau GET dan Kiri akan mengembalikan hasil dalam format JSON. Untuk setiap *request* membutuhkan *API key* yang didapat dengan mendaftar². 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.

2.2.2 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 2.1: 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",
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": "..."

```

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

```

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 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.

2.2.3 Web Service Pencarian Lokasi

Merupakan Kiri API yang digunakan untuk mencari lokasi beserta koordinat *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 hexadecimals	API key yang digunakan

Berikut format Kiri API *responds*:

Listing 2.2: 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 2.3: 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.

BIBLIOGRAFI

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