

**APLIKASI PENDUKUNG KEPUTUSAN PANDUAN WISATA BERBASIS *MOBILE*  
MENGUNAKAN METODE Pencarian Buta dan Terbimbing  
(STUDI KASUS: Kota Pontianak)**

<sup>[1]</sup>Ahmad Tarmizan Kusuma, <sup>[2]</sup>Cucu Suhery, <sup>[3]</sup>Yulrio Brianorman  
<sup>[1]</sup><sup>[2]</sup><sup>[3]</sup>Jurusan Sistem Komputer, Fakultas MIPA Universitas Tanjungpura  
Jl. Prof. Dr. H. Hadari Nawawi, Pontianak  
Telp./Fax.: (0561) 577963  
e-mail:  
<sup>[1]</sup>ahmadtarmizankusuma@gmail.com, <sup>[2]</sup>csuhery@siskom.untan.ac.id,  
<sup>[3]</sup>yulrio.brianorman@siskom.untan.ac.id

**Abstrak**

*Pada umumnya wisatawan selalu berusaha untuk mencari sarana akomodasi/hotel dan restoran yang sesuai dengan dana yang dimilikinya. Wisatawan juga cenderung mengunjungi beberapa tempat sekaligus dalam satu kali perjalanan. Pada penelitian ini dibuat suatu aplikasi yang dapat digunakan untuk membantu wisatawan dalam mengoptimalkan penggunaan dana wisata dan mencari rute perjalanan terpendek. Algoritma yang diterapkan untuk mencari saran penggunaan dana wisata optimal adalah algoritma Depth First Search (DFS) yang merupakan salah satu algoritma pencarian buta. Algoritma yang diterapkan dalam mencari rute perjalanan terpendek adalah algoritma Best-First Search (BeFS) yang merupakan salah satu algoritma pencarian terbimbing dengan menggunakan informasi jarak antartitik yang dihitung menggunakan metode Haversine Formula. Saran rute perjalanan tersebut kemudian ditampilkan ke dalam peta way-point yang disediakan oleh Google Maps. Penelitian ini membangun 2 buah aplikasi yaitu aplikasi backend berbasis web yang digunakan untuk pengelolaan data dan penerapan algoritma, dan aplikasi frontend berbasis mobile yang dibangun dengan menggunakan framework PhoneGap. Pertukaran data antara aplikasi backend dan aplikasi frontend menggunakan format pertukaran data berupa JSON. Studi kasus yang digunakan pada penelitian ini adalah wisata di Kota Pontianak yang merupakan ibu kota Provinsi Kalimantan Barat. Berdasarkan perbandingan hasil pengujian antara cara perhitungan dan implementasi program dari kasus-kasus yang digunakan, dapat diambil kesimpulan bahwa algoritma DFS berhasil diterapkan untuk mencari saran penggunaan dana wisata yang optimal dan algoritma BeFS berhasil diterapkan untuk menghitung saran rute perjalanan terpendek.*

**Kata Kunci:** *Depth First Search, Best-First Search, Haversine Formula, Google Maps, way-point, mobile web, JSON, PhoneGap*

**1. PENDAHULUAN**

Kota Pontianak merupakan tujuan wisata yang ramai dikunjungi oleh wisatawan setiap tahun. Menurut data yang diperoleh dari *website* resmi Badan Pusat Statistik Kota Pontianak yaitu pontianakkota.bps.go.id, pada tahun 2013 tercatat sebanyak 25.814 orang wisatawan asing yang datang ke Kota Pontianak. Jumlah tersebut meningkat menjadi 32.233 orang pada tahun 2014 [1].

Situs Traveloka yang mencatat ada sebanyak 31 hotel berbintang di Pontianak

dengan harga kamar mulai dari Rp 210.000,- sampai dengan Rp 1.160.000,- untuk harga bulan Mei 2016 [2]. Selain keperluan hotel para wisatawan juga membutuhkan referensi terkait restoran-restoran yang ada di Kota Pontianak sehingga dibutuhkan solusi untuk memberikan saran kepada wisatawan dalam memilih hotel dan restoran yang sesuai dengan dana yang dimiliki.

Wisatawan biasanya mengunjungi banyak tempat sekaligus dalam satu kali perjalanan. Wisatawan yang tidak memiliki pengetahuan akan jalan dan lokasi tempat di Kota Pontianak tentu sulit menentukan rute

perjalanan terpendek untuk menghemat waktu dan biaya perjalanan. Berdasarkan permasalahan tersebut dibutuhkan solusi untuk memberikan saran rute perjalanan terpendek kepada wisatawan. Rute perjalanan terpendek tersebut harus dilengkapi dengan tampilan peta rute agar wisatawan tidak tersesat.

Algoritma *Depth First Search (DFS)* merupakan salah satu algoritma pada pencarian buta. Algoritma DFS dapat digunakan untuk menelusuri pohon data dengan tujuan tertentu sehingga dihasilkan solusi pencarian. Penelitian sebelumnya yang menerapkan algoritma DFS dilakukan oleh Arif Herwanto dan Bambang Eka Purnama yang berjudul "Penerapan *Metode Depth First Search* pada Pencarian Rute Bus Kota Berbasis *Web Mobile* di Solo". Hasil yang dipilih berupa bus terbaik yang dapat digunakan dengan pertimbangan rute yang dilalui dan tarif termurah [3].

Algoritma *Best-First Search (BeFS)* merupakan salah satu algoritma pencarian terbimbing yang sesuai untuk diterapkan dalam pencarian nilai optimal seperti pencarian rute terpendek. Salah satu penelitian tentang pencarian rute terpendek untuk suatu perjalanan wisata menggunakan algoritma BeFS dilakukan oleh Rike Nur Setiyani dengan judul penelitian "Implementasi Algoritma *Best-First Search (BeFS)* pada *Penyelesaian Traveling Salesman Problem (TSP)* (Studi Kasus: Perjalanan Wisata di Kota Yogyakarta)". Tempat-tempat tujuan perjalanan telah ditentukan sebelumnya yaitu sebanyak 8 titik tujuan dari 3 titik awal yang berbeda sehingga dihasilkan rute perjalanan terpendek dari setiap titik awal [4].

Algoritma BeFS membutuhkan pengetahuan akan jarak antartitik. Metode *Haversine Formula* dapat mengukur jarak antartitik berdasarkan koordinat pada garis lintang dan bujur. Salah satu penelitian yang menggunakan metode *Haversine Formula* dilakukan oleh Ryan Herwan Dwi Putra, dkk. yang berjudul "Penerapan Metode *Haversine Formula* pada Sistem Informasi Geografis Pengukuran Luas Tanah". Metode *Haversine Formula* pada penelitian tersebut diterapkan untuk memudahkan pegawai Badan Pertanahan Nasional (BPN) dalam melakukan pengukuran luas tanah [5].

Berdasarkan hal-hal yang telah dijabarkan, dilakukan penelitian untuk membuat suatu aplikasi yang dapat menghitung saran-saran penggunaan dana wisata optimal menggunakan metode pencarian buta algoritma DFS. Algoritma DFS dipilih dalam penyelesaian masalah penggunaan wisata optimal karena algoritma tersebut memiliki aturan-aturan penelusuran pohon data. Penelitian ini juga mengukur jarak antar titik menggunakan metode *Haversine Formula* dan menggunakan metode pencarian terbimbing algoritma BeFS untuk mencari rute perjalanan terpendek. Algoritma BeFS dipilih dalam penyelesaian masalah pencarian rute terpendek karena dapat digunakan untuk mengurutkan titik-titik tujuan perjalanan sehingga dihasilkan rute perjalanan terpendek.

## 2. LANDASAN TEORI

### 2.1. Sistem Pendukung Keputusan (SPK)

Sistem Pendukung Keputusan (SPK) atau *Decision Support System (DSS)* adalah sebuah sistem yang mampu memberikan kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah dengan kondisi semi terstruktur dan tak terstruktur. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi semi terstruktur dan situasi yang tidak terstruktur, di mana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat [6].

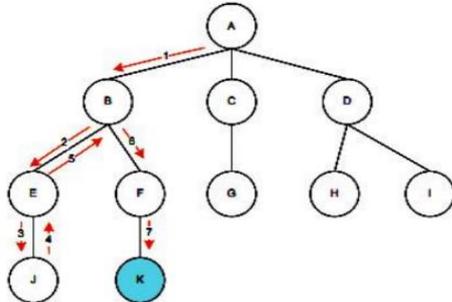
Sistem Pendukung Keputusan (SPK) memiliki lima karakteristik utama yaitu [7]:

- a. Sistem yang berbasis komputer
- b. Membantu pengambilan keputusan
- c. Memecahkan masalah-masalah rumit yang mustahil dilakukan dengan kalkulasi manual
- d. Melalui cara simulasi yang interaktif
- e. Data dan model analisis sebagai komponen utama.

### 2.2. Algoritma *Depth First Search (DFS)*

*Depth First Search* merupakan pencarian mendalam dan salah satu bagian dari pencarian buta. Pencarian dimulai dari level paling pertama, kemudian dilanjutkan ke anak paling kiri pada level berikutnya. Demikian seterusnya sampai tidak terdapat anak lagi atau level yang paling dalam. Jika

pencarian belum menemukan solusi, maka dilakukan penelusuran kembali ke simpul sebelumnya dan dilanjutkan ke simpul tetangga. Proses ini diulangi terus hingga menemukan solusi [8].



Gambar 1. Pohon *Depth First Search* [3]

Gambar 1 merupakan contoh pohon data dengan simpul A sebagai permulaan dan simpul K sebagai tujuan. Pohon tersebut dapat ditelusuri dengan menggunakan aturan algoritma DFS untuk mendapatkan solusi yang diinginkan. Penelusuran harus berhenti jika telah sampai pada simpul K. Penelusuran dimulai dari simpul A sebagai simpul akar dan dilanjutkan ke simpul di level selanjutnya yang paling kiri yaitu simpul B. Simpul tujuan belum ditemukan sehingga penelusuran dilanjutkan ke simpul di level selanjutnya yaitu simpul E yang merupakan anak simpul dari simpul B. Simpul E bukan merupakan tujuan sehingga penelusuran dilanjutkan ke simpul J. Penelusuran telah sampai ke level terdalam dan seluruh simpul di bawah simpul E telah ditelusuri namun, simpul tujuan belum ditemukan sehingga penelusuran dilanjutkan ke simpul tetangga dari simpul E yaitu simpul F. Simpul F bukan merupakan simpul tujuan, penelusuran dilanjutkan ke anak simpul F yaitu simpul K. Simpul tujuan (simpul K) telah ditemukan sehingga penelusuran berhenti. Solusi penelusuran yang didapat menggunakan algoritma DFS adalah A-B-E-J-F-K.

### 2.3. Algoritma *Best-First Search (BeFS)*

Algoritma *Best-First Search* merupakan salah metode pencarian terbimbing yang menggunakan kombinasi dari algoritma *Depth First Search* dengan algoritma *Breadth First Search* dengan mengambil kelebihan dari kedua algoritma tersebut. Apabila

pada pencarian dengan algoritma *Hill Climbing* tidak diperbolehkan untuk kembali ke simpul pada level yang lebih rendah meskipun simpul di level yang lebih rendah tersebut memiliki nilai *heuristic* yang lebih baik, lain halnya pada algoritma BeFS, pencarian diperbolehkan mengunjungi simpul di level yang lebih rendah, jika ternyata simpul di level yang lebih tinggi memiliki nilai *heuristic* yang lebih buruk [8].

Penerapan algoritma BeFS secara umum membutuhkan dua antrian yang berisi simpul-simpul, yaitu:

- Open* yaitu berisi simpul-simpul yang sudah dibangkitkan, sudah memiliki fungsi *heuristic* namun belum diuji. Umumnya berupa antrian berprioritas yang berisi elemen-elemen dengan nilai *heuristic* tertinggi.
- Closed* yaitu berisi simpul-simpul yang sudah diuji.

Salah satu jenis algoritma BeFS yang paling sederhana adalah algoritma *Greedy Best-First Search*. Algoritma *Greedy BeFS* menyelesaikan permasalahan langkah per langkah dan mengambil keputusan pada setiap langkahnya dengan memilih maksimum lokal atau keputusan terbaik pada langkah tersebut. Keputusan yang telah diambil tidak dapat diubah kembali dengan harapan keputusan tersebut merupakan solusi terbaik atau maksimum global. Fungsi *heuristic* pada algoritma *Greedy BeFS* ditulis dalam persamaan [9]:

$$f(n) = h'(n)$$

Keterangan:

$f(n)$  = prakiraan nilai dari titik awal ke titik tujuan

$h'(n)$  = prakiraan nilai dari posisi sekarang ke posisi tujuan

Pencarian rute terpendek dengan algoritma *Greedy BeFS* menggunakan langkah-langkah sebagai berikut:

- Ambil seluruh titik yang dapat dikunjungi dari titik sekarang.
- Cari maksimum lokal ke titik selanjutnya.
- Tandai titik sekarang sebagai titik yang telah dikunjungi, dan pindah ke maksimum lokal yang telah ditentukan sebagai titik berada sekarang.

4. Kembali ke langkah 1 sampai titik tujuan didapatkan.

#### 2.4. Metode Haversine Formula

*Haversine Formula* merupakan metode untuk mengetahui jarak antar dua titik dengan memperhitungkan bahwa bumi adalah sebuah bidang yang memiliki derajat kelengkungan. Metode *Haversine Formula* menghitung jarak antara 2 titik berdasarkan panjang garis lurus pada garis bujur dan lintang [5]. Metode *Haversine Formula* pada penelitian ini digunakan untuk menghitung prakiraan nilai dari posisi sekarang ke posisi tujuan. Jari-jari bumi rata-rata adalah 6378 Km [10]. Berikut bentuk Rumus *Haversine Formula* :

$$\Delta\phi = \text{lat } 2 - \text{lat } 1$$

$$\Delta\lambda = \text{long } 2 - \text{long } 1$$

$$a = \sin^2(\Delta\phi/2) + \cos \phi 1 \cdot \cos \phi 2 \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

Keterangan:

$\Delta\phi$  = selisih koordinat lintang (rad)

$\Delta\lambda$  = selisih koordinat bujur (rad)

R = radius bumi

d = jarak dua titik

#### 2.5. UML

UML yang merupakan singkatan dari *Unified Modelling Language* adalah sekumpulan pemodelan konvensi yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak dalam kaitannya dengan objek [11]. UML dapat juga diartikan sebuah bahasa grafik standar yang digunakan untuk memodelkan perangkat lunak berbasis objek. UML pertama kali dikembangkan pada pertengahan tahun 1900-an dengan kerja sama antara James Rumbaugh, Grady Booch dan Ivar Jacobson, yang masing-masing telah mengembangkan notasi mereka sendiri di awal tahun 1900-an [12]. Penelitian ini menggunakan UML dalam perancangan sistem. Diagram-diagram UML yang digunakan adalah *use case diagram*, dan *activity diagram*.

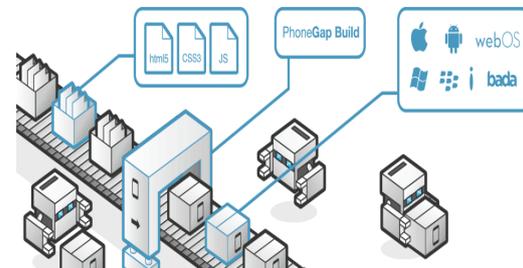
#### 2.6. Google Maps

Google Maps adalah layanan peta gratis dan *online* yang disediakan oleh Google.

Layanan ini dapat diakses melalui situs <http://maps.google.com>. Pada situs tersebut kita dapat melihat informasi geografis pada hampir semua wilayah di bumi. Layanan Google Maps interaktif, karena di dalamnya peta dapat digeser sesuai dengan keinginan pengguna, mengubah tingkat *zoom*, serta mengubah tampilan peta [13]. Layanan Google Maps pada penelitian ini digunakan sebagai pemetaan lokasi tempat-tempat dan juga untuk menampilkan rute perjalanan.

#### 2.7. PhoneGap

Pengembangan aplikasi berbasis Android dapat dilakukan menggunakan berbagai macam perangkat lunak pengembang aplikasi. Pada penelitian ini pengembangan aplikasi Android menggunakan PhoneGap. PhoneGap adalah sebuah *framework* pembuatan aplikasi *mobile* berbasis *open source* yang memungkinkan pengembangan *native mobile* dengan keahlian HTML5, CSS, dan JavaScript [14]. Aplikasi yang dibangun dengan menggunakan *framework* PhoneGap dapat dijadikan ke dalam aplikasi *mobile* berbagai *platform* seperti iOS, Android, Windows Phone, Symbian, dan Blackberry OS.



Gambar 2. Cara Kerja PhoneGap [15]

### 3. METODOLOGI PENELITIAN

Proses penelitian dimulai dengan melakukan studi pustaka terkait algoritma DFS, Algoritma BeFS, metode *Haversine Formula*, *web service*, Phonegap, dan teori-teori penunjang lainnya yang diperlukan. Selanjutnya dilakukan pengumpulan data-data objek penelitian yang berupa data-data objek wisata, hotel, restoran, dan *event* yang ada di Kota Pontianak baik dilakukan dengan survei lapangan mau pun ke dinas-dinas terkait.

Tahap selanjutnya adalah tahap perancangan yang berupa perancangan diagram blok dan perancangan aplikasi. Diagram

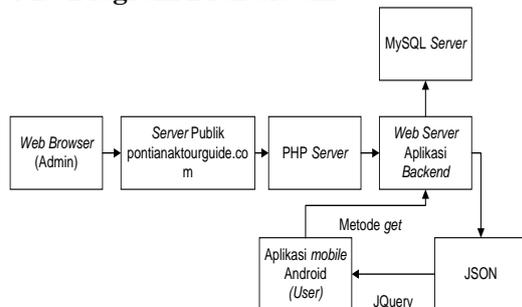
blok menunjukkan hubungan antaraplikasi yang dibangun yaitu aplikasi *backend* dan aplikasi *frontend*. Perancangan perangkat lunak menggunakan metode UML. Kebutuhan-kebutuhan fungsional yang harus bisa dilakukan oleh aplikasi yang dibangun dirancang menggunakan *use case diagram* dan *activity diagram*.

Setelah tahap perancangan selesai dilakukan, penelitian dilanjutkan ke tahap implementasi. Tahap implementasi menerapkan rancangan perangkat lunak ke dalam kode-kode program. Tahap ini membangun aplikasi *backend* yang berbasis *web* dan aplikasi *frontend* berbasis *mobile* Android. Algoritma-algoritma yang digunakan pada penelitian ini yaitu algoritma DFS, algoritma BeFS, dan metode *Haversine Formula* diimplementasikan pada aplikasi *backend*.

Tahap yang terakhir yaitu pengujian aplikasi. Pengujian diperlukan untuk mengetahui kinerja aplikasi. Jika terdapat kesalahan pada implementasi, dilakukan evaluasi kesalahan dimulai dari tahap perancangan. Jika aplikasi sudah bekerja seperti yang diharapkan maka penelitian ini selesai dilakukan.

#### 4. PERANCANGAN SISTEM

##### 4.1. Diagram Blok Sistem



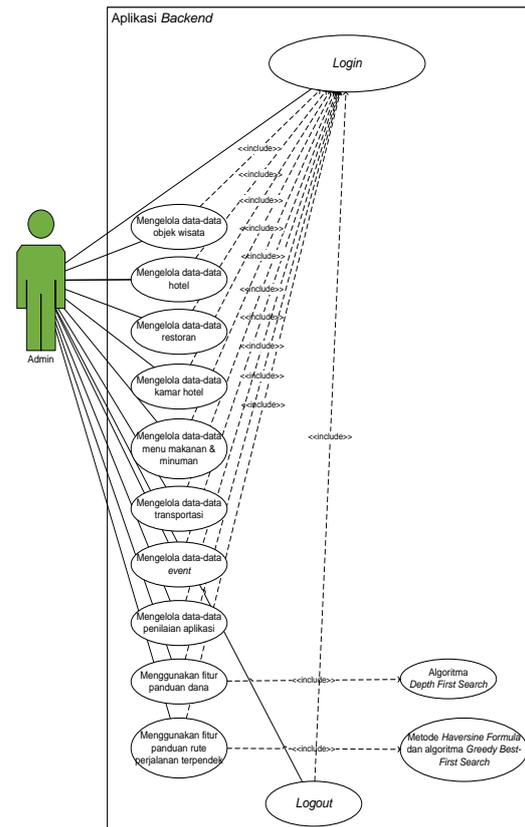
Gambar 3. Diagram Blok Sistem

Gambar 3 menunjukkan hubungan antar-komponen pada sistem yang dirancang. Aplikasi *backend* yang berbasis *web* dapat diakses oleh *admin* melalui *web browser*. *Admin* dapat melakukan manajemen (membuat, mengubah dan menghapus) data-data yang ada dan menyimpan data-data tersebut ke dalam *database* MySQL.

Aplikasi *frontend* digunakan pada sisi pengguna. Pengguna menggunakan fitur-fitur yang disediakan oleh sistem dengan

mengirimkan parameter-parameter yang diperlukan menggunakan metode *get* ke aplikasi *backend*. Aplikasi *backend* akan mengirimkan hasil atau *output* yang diminta oleh pengguna dalam bentuk JSON yang diterima pada sisi aplikasi *mobile* dengan menggunakan JQuery.

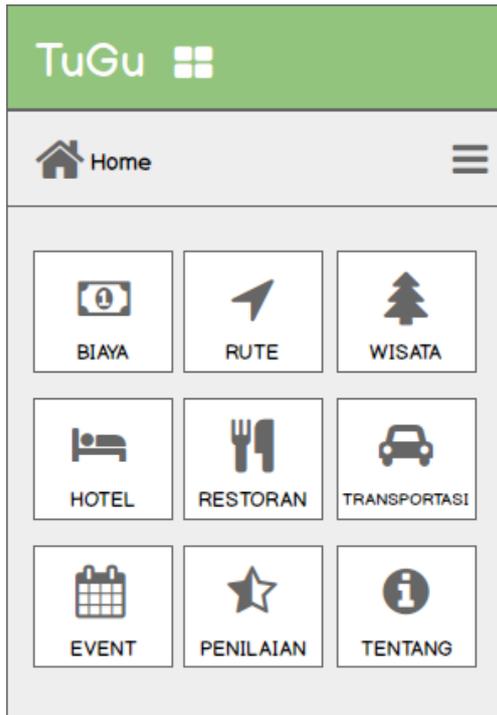
##### 4.2. Use case Diagram dan Rancangan Tampilan Antarmuka



Gambar 4. Aktivitas-aktivitas pada Aplikasi Backend

Gambar 4 menunjukkan aktivitas *admin* menjalankan aplikasi *backend*. *Admin* memiliki hak akses khusus untuk menggunakan aplikasi *backend* sehingga dibutuhkan proses *login* dan *logout*. *Admin* dapat melakukan pengolahan data seperti menambah data, mengubah data, dan menghapus data. Data-data yang dikelola oleh *admin* yang meliputi data objek wisata, data hotel, data kamar hotel, data restoran, data menu makanan dan minuman, data transportasi, dan data *event*.

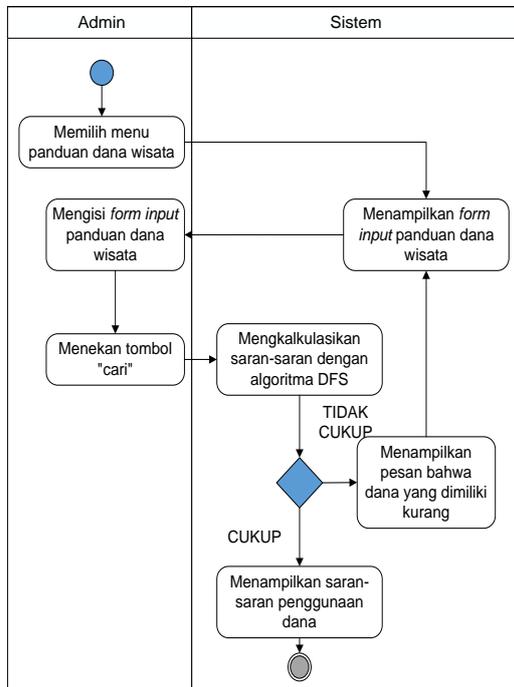
Rancangan tampilan menu utama aplikasi *frontend* dapat dilihat pada Gambar 6:



Gambar 5. Rancangan Tampilan Halaman Utama Aplikasi *Frontend*

### 4.3. Activity Diagram

#### 4.3.1. Panduan Dana

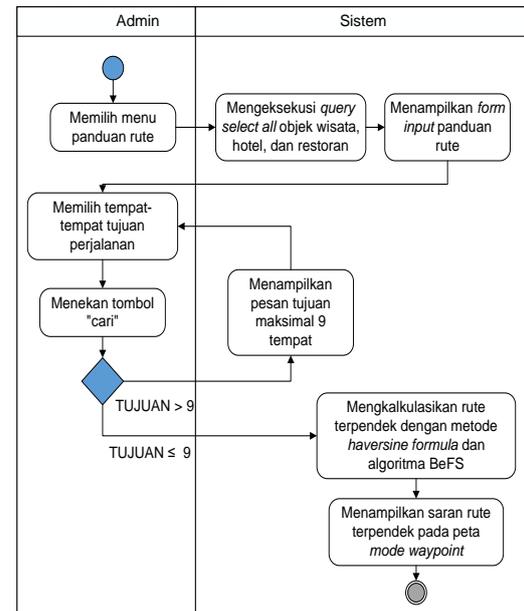


Gambar 6. Activity Diagram Panduan Dana Wisata Aplikasi *Backend*

Gambar 6 menunjukkan alur aktivitas penggunaan fitur panduan dana oleh *admin*.

*Admin* memilih menu panduan dana wisata kemudian sistem menampilkan *form input* panduan dana wisata. *Admin* mengisi *form* tersebut dan kemudian menekan tombol “cari”. Sistem mengkalkulasikan kebutuhan wisata dengan algoritma DFS sesuai dengan informasi yang dimasukkan oleh *admin*. Jika dana wisata kurang, sistem menampilkan pesan bahwa dana wisata yang dimiliki tidak cukup. Jika dana yang dimiliki cukup, sistem menampilkan saran-saran penggunaan dana wisata optimal.

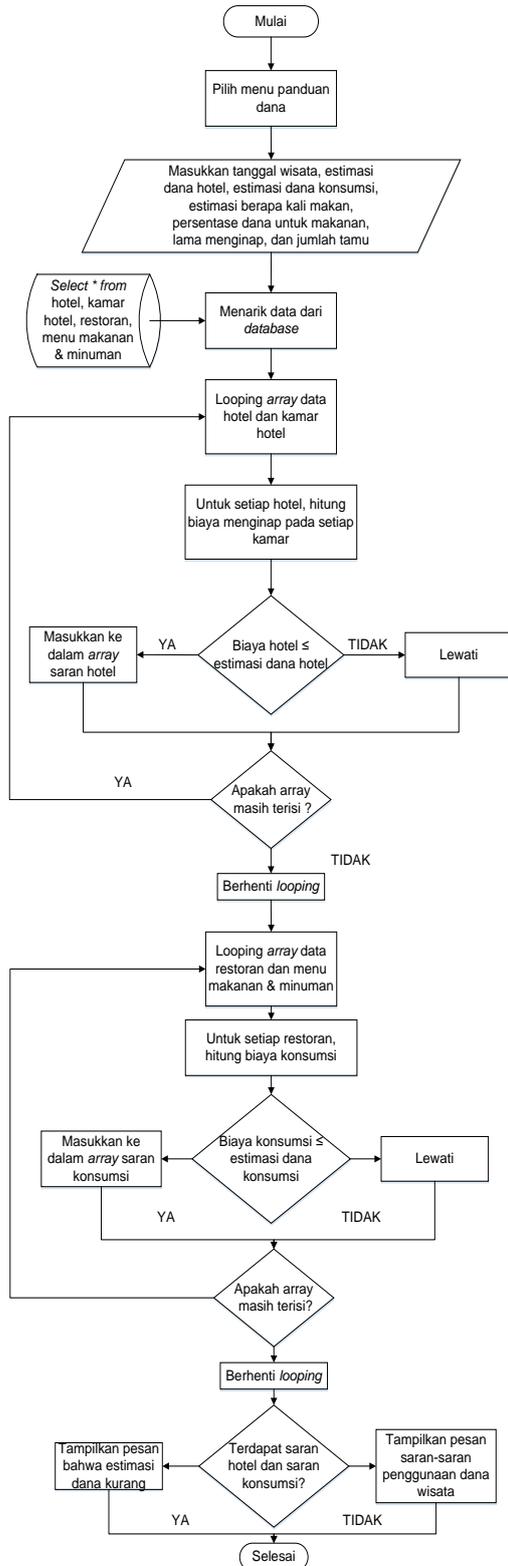
#### 4.3.2. Panduan Route



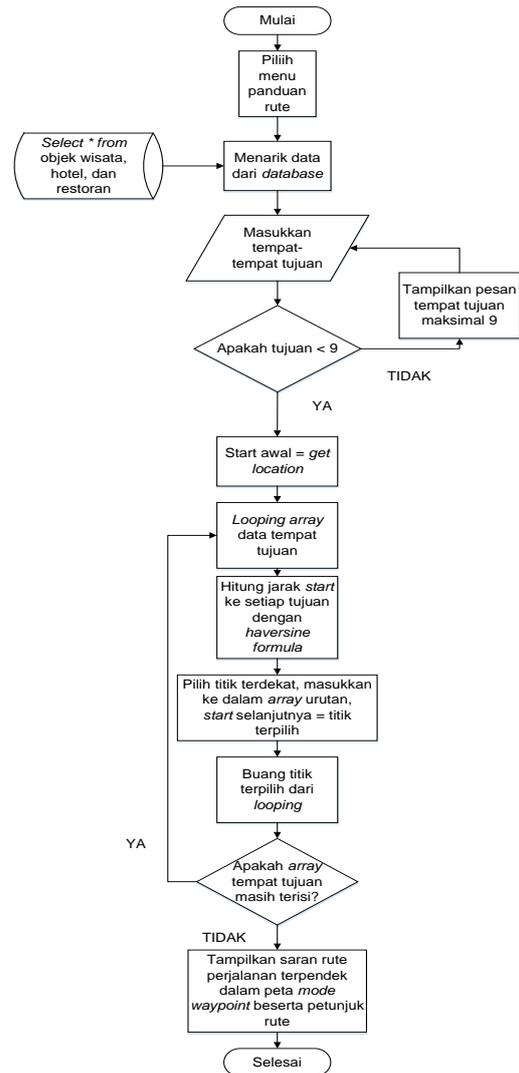
Gambar 7. Activity Diagram Panduan Route Aplikasi *Backend*

Gambar 7 menunjukkan alur aktivitas penggunaan fitur panduan rute. *Admin* memilih menu panduan rute, kemudian sistem menampilkan pilihan tempat-tempat tujuan perjalanan. Setelah selesai memilih tempat-tempat tujuan, *admin* menekan tombol “cari”. Jika tempat yang dipilih lebih dari 9, sistem menampilkan pesan bahwa maksimal tempat yang dituju adalah 9 titik. Jika tempat yang ingin dituju tidak melebihi batas, sistem kemudian mengkalkulasi rute terpendek dengan metode *haversine formula* dan algoritma BeFS. Sistem kemudian menampilkan halaman saran panduan rute terpendek yang berupa peta *mode waypoint*.

#### 4.4. Perancangan Sistem Proses Panduan Wisata



Gambar 8. Flowchart Perhitungan Panduan Dana



Gambar 9. Flowchart Perhitungan Panduan Route

## 5. PENGUJIAN DAN PEMBAHASAN

### 5.1. Pengujian Fitur Panduan Dana

Pengujian fitur panduan dana menggunakan data uji sebanyak 10 hotel dan 5 restoran.

Tabel 1. Data Uji Hotel

No	Hotel	Jenis Kamar	Kapasitas (tamu)	Harga (Rp)
1.	Aston	Deluxe	2	1.045.440
		Superior Room Only	2	755.040
2.	Borneo Hotel	Executive	2	476.000
		Superior	2	410.667
3.	Golden Tulip	Superior Twin Room Only	2	570.667
		Superior Queen	2	770.667

No	Hotel	Jenis Kamar	Kapasitas (tamu)	Harga (Rp)
4.	Green Leaf Inn	Standard Single	1	234.667
		Superior Double	2	328.534
5.	HARRIS Hotel	Harris Room	2	548.000
		Harris Unique Room	2	864.000
6.	Hotel Gajah-mada	Deluxe	2	670.000
		Executive	2	825.000
7.	Hotel Garuda	Standard Double Or Twin Room	2	413.334
		Executive Suite	2	786.667
8.	Hotel Kini	Superior Twin	2	616.000
		Deluxe Twin	2	739.200
9.	Orchardz Hotel Gajah-mada	Superior	2	493.334
		Deluxe	2	560.000
10.	Transera Hotel	Superior Room Only	2	420.233
		Executive Room	2	686.900

No	Restoran	Makanan		Minuman	
		Nama	Harga (Rp)	Nama	Harga (Rp)
3.	D'Bamboo	Chai Kue Kukus	10.000	Es Air Tahu Spesial	8.000
		Traditional Chai Kue	15.000	Liang Teh	4.000
4.	Pondok Ale Ale	Asam Pedas Pontianak	31.000	Es Jeruk Pontianak	17.000
		Ayam Opor Dada	17.000	Es Kalimantan	20.000
5.	Restoran Pondok Nelayan	Ikan Lais Asam Pedas	24.000	Es Lemon Madu	10.000
		Ayam Kecap	40.000	Es Susu Kedelai	8.000

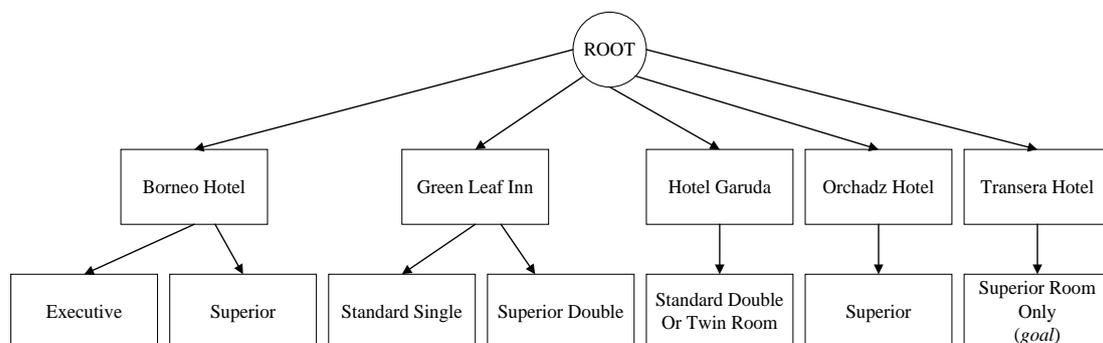
Tabel 2. Data Uji Restoran

No	Restoran	Makanan		Minuman	
		Nama	Harga (Rp)	Nama	Harga (Rp)
1.	Ayam Pak Usu	Ayam Ponti	17.000	Es Kelapa Muda	13.000
		Bubur Ikan Alu-alu	21.000	Es Jeruk Besar	12.500
2.	Bubur Pedas Pak Ngah	Bubur Paddas	9.000	Es Sari Timun	6.000
		Bubur Paddas Kikil	12.000	Es Cincau	7.000

Pengujian penerapan algoritma *Depth First Search* dilakukan dengan membandingkan hasil perhitungan manual dan hasil perhitungan pada aplikasi (implementasi program). Kasus yang diangkat adalah sebagai berikut:

1. Tanggal perjalanan wisata 18 Agustus 2016
2. Memiliki dana untuk keperluan hotel sebanyak Rp 500.000,-
3. Memiliki dana untuk keperluan konsumsi Rp 300.000,-
4. Persentase dana konsumsi untuk makanan sebesar 60%
5. Menginap selama 1 malam
6. Makan sebanyak 2 kali untuk setiap wisatawan
7. Jumlah wisatawan 2 orang

#### 5.1.1. Saran Penggunaan Dana Hotel



Gambar 10. Pohon Pencarian Hotel

Solusi penelusuran pohon pencarian hotel pada Gambar 10 menggunakan aturan algoritma *Depth First Search* adalah sebagai berikut: Borneo Hotel - Executive - Superior - Green Leaf Inn - Standard Single - Superior Double - Hotel Garuda - Standard Double Or Twin Room - Orchardz Hotel - Superior - Transera Hotel - Superior Room Only (goal).

Pengujian saran penggunaan dana hotel yang dilakukan pada aplikasi menghasilkan saran-saran sebagai berikut:

Gambar 11. Tampilan Halaman Saran Penggunaan Dana Hotel Alternatif 1

Tabel 3. Hasil Pengujian Saran Hotel pada Aplikasi

No	Hotel	Jenis Kamar	Biaya Total (Rp)
1.	Orchardz Hotel	Superior	493.334
2.	Borneo Hotel	Executive	476.000
3.	Green Leaf Inn	Standard Single	469.334

No	Hotel	Jenis Kamar	Biaya Total (Rp)
4.	Transera Hotel	Superior Room Only	420.233
5.	Hotel Garuda	Standard Double Or Twin Room	413.334
6.	Borneo Hotel	Superior	410.667
7.	Green Leaf Inn	Superior Double	328.534

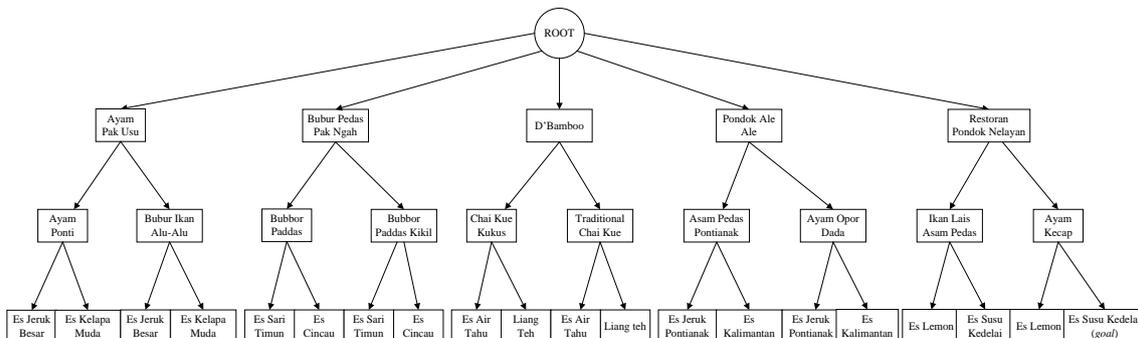
Perbandingan hasil pengujian algoritma DFS dalam pencarian saran penggunaan dana hotel 1 adalah sebagai berikut:

Tabel 4. Perbandingan Hasil Pengujian Saran Hotel

No	Perhitungan		Aplikasi	
	Hotel	Jenis Kamar	Hotel	Jenis Kamar
1.	Orchardz	Superior	Orchardz	Superior
2.	Borneo	Executive	Borneo	Executive
3.	Green Leaf Inn	Standard Single	Green Leaf Inn	Standard Single
4.	Transera Hotel	Superior Room Only	Transera Hotel Pontianak	Superior Room Only
5.	Hotel Garuda	Standard Double	Hotel Garuda	Standard Double
6.	Borneo	Superior	Borneo	Superior
7.	Green Leaf Inn	Superior Double	Green Leaf Inn	Superior Double

Tabel 4. menunjukkan semua saran penggunaan dana hotel 100% sama.

### 5.1.2. Saran Penggunaan Dana Konsumsi



Gambar 12. Pohon Pencarian Saran Konsumsi

Solusi penelusuran pohon pencarian konsumsi pada Gambar 12 berdasarkan aturan algoritma *Depth First Search* adalah sebagai berikut: Ayam Pak Usu - Ayam Ponti - Es Jeruk Besar - Es Kelapa Muda - Bubur Ikan Alu-Alu - Es Jeruk Besar - Es Kelapa Muda - Bubur Pedas Pak Ngah - Bubor Paddas - Es Sari Timun - Es Cincau - Bubor Paddas Kikil - Es Sari Timun - Es Cincau - D'Bamboo - Chai Kue Kukus - Es Air Tahu - Liang Teh - Traditional Chai Kue - Es Air Tahu - Liang Teh - Pondok Ale Ale - Asam Pedas Pontianak - Es Jeruk Pontianak - Es Kalimantan - Ayam Opor Dada - Es Jeruk Pontianak - Es Kalimantan - Restoran Pondok Nelayan - Ikan Lais Asam Pedas - Es Lemon - Es Susu Kedelai - Ayam Kecap - Es Lemon - Es Susu Kedelai (*goal*).

Pengujian saran penggunaan dana konsumsi yang dilakukan pada aplikasi menghasilkan saran-saran sebagai berikut:

Restoran	Pondok Ale Ale
Makanan	Asam Pedas Pontianak <b>KHAS</b>
Harga makanan	Rp 31.000,00
Minuman	Es Kalimantan <b>KHAS</b>
Harga minuman	Rp 20.000,00
Total porsi	4 porsi
Total Harga Makanan	Rp 124.000,00
Total Harga Minuman	Rp 80.000,00
<b>Total</b>	<b>Rp 204.000,00</b>

Gambar 13. Tampilan Halaman Saran Penggunaan Dana Konsumsi Alternatif 1

Tabel 5. Hasil Pengujian Saran Konsumsi pada Aplikasi

No	Restoran	Menu Makanan	Menu Minuman	Biaya Total (Rp)
1.	Pondok Ale Ale	Asam Pedas Pontianak	Es Kalimantan	204.000
2.	Pondok Nelayan	Ayam Kecap	Es Lemon Madu	200.000
3.	Pondok Nelayan	Ayam Kecap	Susu Kedelai	192.000
4.	Pondok Ale Ale	Asam Pedas Pontianak	Es Jeruk Pontianak	192.000
5.	Pondok Ale Ale	Ayam Opor Dada	Es Kalimantan	148.000
6.	Pondok Ale Ale	Ayam Opor Dada	Es Jeruk Pontianak	136.000
7.	Pondok Nelayan	Ikan Lais Asam Pedas	Es Lemon Madu	136.000
8.	Ayam Pak Usu	Bubur Ikan Alu-Alu	Es Kelapa Muda	136.000
9.	Ayam Pak Usu	Bubur Ikan Alu-Alu	Es Jeruk Besar	134.000
10.	Pondok Nelayan	Ikan Lais Asam Pedas	Susu Kedelai	128.000

Tabel 6. Perbandingan Hasil Pengujian Saran Konsumsi

No	Perhitungan			Aplikasi		
	Restoran	Makanan	Minuman	Restoran	Makanan	Minuman
1.	Pondok Ale Ale	Asam Pedas	Es Kalimantan	Pondok Ale Ale	Asam Pedas	Es Kalimantan
2.	Pondok Nelayan	Ayam Kecap	Es Lemon Madu	Pondok Nelayan	Ayam Kecap	Es Lemon Madu
3.	Pondok Ale Ale	Asam Pedas	Es Jeruk Pontianak	Pondok Nelayan	Ayam Kecap	Susu Kedelai
4.	Pondok Nelayan	Ayam Kecap	Susu Kedelai	Pondok Ale Ale	Asam Pedas	Es Jeruk Pontianak
5.	Pondok Ale Ale	Ayam Opor	Es Kalimantan	Pondok Ale Ale	Ayam Opor	Es Kalimantan

No	Perhitungan			Aplikasi		
	Restoran	Makanan	Minuman	Restoran	Makanan	Minuman
6.	Ayam Pak Usu	Bubur Alu-Alu	Es Kelapa Muda	Pondok Ale Ale	Ayam Opor	Es Jeruk Pontianak
7.	Pondok Ale Ale	Ayam Opor	Es Jeruk Pontianak	Pondok Nelayan	Ikan Lais Asam	Es Lemon Madu
8.	Pondok Nelayan	Ikan Lais Asam	Es Lemon Madu	Ayam Pak Usu	Bubur Alu-Alu	Es Kelapa Muda
9.	Ayam Pak Usu	Bubur Alu-Alu	Es Jeruk Besar	Ayam Pak Usu	Bubur Alu-Alu	Es Jeruk Besar
10.	Pondok Nelayan	Lais Asam Pedas	Susu Kedelai	Pondok Nelayan	Lais Asam Pedas	Susu Kedelai

Tabel 6 menunjukkan alternatif saran yang dihasilkan 100% sama. Terdapat perbedaan urutan alternatif saran dengan biaya total yang sama. Perbedaan urutan tersebut disebabkan oleh perbedaan teknik pengurutan. Teknik pengurutan pada aplikasi hanya memperhatikan parameter biaya total dengan menggunakan *function sorting usort*, sedangkan pada perhitungan juga memperhatikan urutan nama restoran secara alfabetis.

## 5.2. Pengujian Fitur Panduan Rute

Pengujian penerapan algoritma BeFS dan metode *Haversine Formula* dilakukan dengan membandingkan hasil perhitungan manual dan hasil perhitungan pada aplikasi (implementasi program). Kasus yang diangkat adalah:

Posisi awal:

Rektorat Universitas Tanjungpura, koordinat -0,0597392, 109,3455852

Tujuan:

1. Museum Provinsi Kalimantan Barat, koordinat (-0,048287, 109,342734)
2. Rumah Radakng, koordinat (-0,046981, 109,319142)
3. Taman Alun Kapuas, koordinat (-0,021621, 109,339357)

Jarak antartitik dihitung menggunakan rumus *Haversine Formula*:

Tabel 7. Jarak Antartitik

No	Titik 1	Titik 2	Jarak (Km)
1.	Rektorat Untan	Museum Provinsi	1,31
2.	Rektorat Untan	Rumah Radakng	3,27

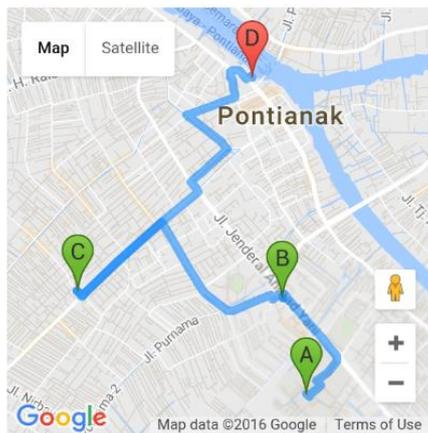
No	Titik 1	Titik 2	Jarak (Km)
3.	Rektorat Untan	Taman Alun Kapuas	4,30
4.	Museum Provinsi	Rumah Radakng	2,63
5.	Museum Provinsi	Taman Alun Kapuas	2,99
6.	Rumah Radakng	Taman Alun Kapuas	3,61

Perhitungan algoritma *Greedy BeFS* untuk mencari urutan tempat-tempat yang dikunjungi dapat dilihat pada tabel 8:

Tabel 8. Perhitungan Algoritma BeFS

No	Posisi Mulai	Tempat Tujuan	Jarak / f(n) (Km)	Tujuan Terdekat
1.	Rektorat Untan	Museum Provinsi	1,31	Museum Provinsi
		Rumah Radakng	3,27	
		Taman Alun Kapuas	4,30	
2.	Museum Provinsi	Rumah Radakng	2,63	Rumah Radakng
		Taman Alun Kapuas	2,99	
3.	Rumah Radakng	Taman Alun Kapuas	3,61	Taman Alun Kapuas

Rute perjalanan terpendek berdasarkan tabel 8 yaitu: Rektorat Universitas Tanjungpura - Museum Provinsi Kalimantan Barat - Rumah Radakng - Taman Alun Kapuas.



Gambar 14. Peta Rute Perjalanan

Perbandingan jarak antartitik dengan cara perhitungan dan implementasi program dapat dilihat pada Tabel 9:

Tabel 9. Perbandingan Jarak Antartitik

No	Titik 1	Titik 2	Jarak Perhitungan (Km)	Jarak Program (Km)
1.	Rektorat Untan	Museum	1,31	1,31
2.	Rektorat Untan	Rumah Radakng	3,27	3,27
3.	Rektorat Untan	Taman Alun Kapuas	4,30	4,30
4.	Museum	Rumah Radakng	2,63	2,63
5.	Museum	Taman Alun Kapuas	2,99	2,99
6.	Rumah Radakng	Taman Alun Kapuas	3,61	3,61

Perbandingan urutan tempat yang dikunjungi berdasarkan cara perhitungan dan pengujian aplikasi dapat dilihat pada Tabel 10:

Tabel 10. Perbandingan Hasil Pengujian Algoritma BeFS

No	Perhitungan	Aplikasi
1.	Rektorat Universitas Tanjungpura (titik awal)	Rektorat Universitas Tanjungpura (titik awal)
2.	Museum Provinsi Kalimantan Barat	Museum Provinsi Kalimantan Barat
3.	Rumah Radakng	Rumah Radakng
4.	Taman Alun Kapuas (titik akhir)	Taman Alun Kapuas (titik akhir)

Tabel 10 menunjukkan kesamaan urutan-urutan tempat yang dikunjungi dengan cara perhitungan dan pengujian aplikasi. Gambar 14 menunjukkan peta rute perjalanan yang ditempuh. Jalan yang dipilih tersebut merupakan jalan yang terbaik menurut *Google Maps*. Jarak tempuh yang sebenarnya diketahui dengan menggunakan *Google Maps API* yaitu 10,9 Km.

## 6. KESIMPULAN DAN SARAN

### 6.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan dalam membangun aplikasi pendukung keputusan panduan wisata dapat diambil kesimpulan sebagai berikut:

1. Kriteria utama objek wisata, hotel, dan restoran yang dimasukkan ke dalam penelitian adalah terdaftar di Google Maps. Khusus untuk data transportasi, objek wisata, dan *event* tahunan harus terdata pada Dinas Kebudayaan dan Pariwisata Provinsi Kalimantan Barat.
2. Berdasarkan tabel perbandingan hasil pengujian algoritma *Depth First Search (DFS)* dapat diambil kesimpulan bahwa penerepan algoritma DFS untuk mencari saran penggunaan dana wisata berhasil dilakukan. Alternatif-alternatif saran penggunaan dana hotel yang dihasilkan dengan cara perhitungan dan implementasi program 100% sama. Alternatif-alternatif saran penggunaan dana konsumsi yang dihasilkan dengan cara perhitungan dan implementasi program 100% sama dari sisi nama-nama menu yang disarankan, namun terdapat perbedaan urutan alternatif saran.
3. Perbedaan urutan alternatif saran pada saran penggunaan dana konsumsi dikarenakan perbedaan teknik pengurutan yang diterapkan. Teknik pengurutan pada perhitungan manual mengurutkan nama restoran secara alfabetis jika terdapat alternatif saran dengan biaya total yang sama. Teknik pengurutan pada implementasi program hanya memperhatikan biaya total.
4. Jumlah alternatif saran yang dihasilkan pada fitur panduan dana bervariasi dari 1 alternatif sampai dengan 10 alternatif yang bergantung kepada biaya total.

5. Berdasarkan tabel perbandingan hasil pengujian algoritma *Best-First Search (BeFS)* dengan cara perhitungan dan implementasi program dapat diambil kesimpulan bahwa algoritma BeFS berhasil diterapkan pada fitur panduan rute perjalanan terpendek. Urutan-urutan tempat tujuan perjalanan yang disarankan dengan cara perhitungan dan implementasi program 100%.
6. Rute perjalanan terpendek yang dihasilkan melalui penerapan metode *Haversine Formula* dan algoritma BeFS tidak selalu merupakan rute terpendek karena tidak memperhatikan jalan yang dilalui dan tidak memerhatikan keadaan alam yaitu Sungai Kapuas.
7. Fitur panduan rute tidak optimal jika digunakan di luar Kota Pontianak.

## 6.2. Saran

Penelitian dalam membuat aplikasi pendukung pengambilan keputusan panduan wisata ini pada prosesnya memiliki permasalahan-permasalahan yang perlu dipikirkan untuk menjadi bahan pembelajaran bagi penelitian-penelitian selanjutnya. Ada pun saran pengembangan aplikasi ini adalah sebagai berikut:

1. Menggunakan algoritma yang dapat menemukan solusi pencarian yang lebih cepat untuk diterapkan pada fitur panduan dana.
2. Algoritma *Greedy Best-First Search* yang diterapkan dapat lebih optimal jika jarak antartitik menggunakan panjang jalan sesungguhnya yang dilewati.
3. Penggunaan *framework* PhoneGap dalam membangun aplikasi *mobile* sangat disarankan bagi pengembang yang memiliki dasar pemrograman *web*.

## DAFTAR PUSTAKA

- [1] Badan Pusat Statistik Kota Pontianak, "BPS Kota Pontianak," 24 Juli 2016. [Online]. Available: <https://pontianak-kota.bps.go.id/Subjek/view/id/16#subjek-ViewTab3|accordion-daftar-subjek2>.
- [2] Traveloka, "Hotel di Pontianak," 15 Mei 2016. [Online]. Available: <https://www.-traveloka.com>.
- [3] A. Herwanto, "Penerapan Metode Depth First Search pada Pencarian Rute Bus Kota Berbasis Web Mobile di Solo," *JURNAL ILMIAH GO INFO-TECH*, pp. 7-13, 2013.
- [4] R. N. Setiyani, "Implementasi Algoritma Best-First Search (BeFS) pada Penyelesaian Traveling Salesman Problem (TSP) (Studi Kasus: Perjalanan Wisata di Kota Yogyakarta)," *Jurnal Fourier*, pp. 129-154, 2015.
- [5] R. H. D. Putra, "Penerapan Metode Haversine Formula Pada Sistem Informasi Geografis Pengukuran Luas Tanah," *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, pp. 1-6, 2015.
- [6] E. Turban, *Decision Support Systems and Intelligent Systems 6th Edition*, Upper Saddle River, NJ: Prentice Hall, 2001.
- [7] R. Sprague, *Decision Support Systems: Putting Theory Into Practice*, Englewood Cliffs, NJ: Prentice Hall, 1993.
- [8] S. Kusumadewi, *Artificial Intelligence (Teknik dan Aplikasinya)*, Yogyakarta: Graha Ilmu, 2003.
- [9] S. Russel, *Artificial Intelligence A Modern Approach*, Boston: Pearson, 2010.
- [10] Hadiat, *Kamus Sains*, Jakarta: Balai Pustaka, 2004.
- [11] J. L. Whitten, *Metode Desain dan Analisis Sistem*, Yogyakarta: ANDI, 2004.
- [12] T. Lethbridge, *Object Oriented Software Engineering: Pratical Software Development Using UML and Java*, UK: Mc Graw Hill-Education, 2002.
- [13] C. Ichtara, *Implementasi Aplikasi Sistem Informasi Geografis (SIG) Universitas Indonesia Berbasis Web Menggunakan Google Maps API*, Depok, 2008.
- [14] Y. Yulistira, *Membuat Aplikasi iPhone, Android, BlackBerry itu Gampang*, Mediakita: Jakarta, 2011.
- [15] Adobe, "Adobe PhoneGap Build," 14 September 2016. [Online]. Available: <https://build.phonegap.com/>.