

---

## Pemberian Rekomendasi Hotel Berdasarkan Jarak di Kota Palangka Raya Berbasis Web

Efrans Christian<sup>1)</sup>, Viktor Handrianus Pranatawijaya<sup>2)</sup>, Nova Noor Kamala Sari<sup>3)</sup>, Aprimikardo<sup>4)</sup>

<sup>1)2)3)4)</sup>Jurusan Teknik Informatika Fakultas Teknik Universitas Palangka Raya Kampus Tunjung  
Nyaho Jl. Yos Sudarso Palangka Raya (73112)

<sup>1)</sup>efrans@it.upr.ac.id

<sup>2)</sup>viktorhp@it.upr.ac.id

<sup>3)</sup>novanoorks@it.upr.ac.id

<sup>4)</sup>aprimikardo.ti.2012@gmail.com

### Abstrak

Pada tahun 2015 terdapat 2 hotel berbintang 4, 2 hotel berbintang 3 dan 46 hotel berkelas melati yang ada di Kota Palangka Raya. Karena semakin banyaknya pembangunan hotel yang di bangun di Kota Palangka Raya, sering kali menimbulkan permasalahan bagi para wisatawan yaitu dalam melakukan pencarian hotel atau menentukan hotel berdasarkan jarak jalur terpendek menuju lokasi hotel yang ada di Kota Palangka Raya dengan tujuan untuk meminimalkan biaya transportasi maupun untuk mempercepat suatu perjalanan.

Penelitian ini di buat dengan menggunakan algoritma *dijkstra* untuk menentukan jalur jarak terpendek menuju lokasi hotel tujuan yang ada di Kota Palangka Raya dan metodologi pengembangan perangkat lunak yang di gunakan pada penelitian ini yaitu menggunakan metode *waterfall* yang terdiri dari lima langkah yaitu *requirement analysis*, *design*, *coding*, implementasi dan testing, dan *maintenance*. Pengujian yang dilakukan pada penelitian ini, yaitu menggunakan pengujian blackbox untuk menguji fungsi-fungsi yang ada pada website yang di buat.

**Kata Kunci : Hotel, Waterfall, algoritma *dijkstra***

### Abstract

*The City Government of Palangka Raya, in the year 2015 there was 2 4 star hotels, 2 star hotels 3 and 46 classy hotel jasmine in the City of Palangka Raya. Because of the increasing number of the construction of the hotel built in the City of Palangka Raya, often cause problems for travelers that in doing a hotel search or specify the hotel based on the distance of the shortest path to the location of the hotel in the City of Palangka Raya with objective to minimize the cost of transportation and to speed up a journey.*

*The research is made by using the algorithm of *dijkstra* to determine the path of the shortest distance to the location of the hotel destination in the City of Palangka Raya and software development methodologies in used in this research is using the waterfall method consists of five steps namely requirement analysis, design, coding, implementation and testing, and maintenance. Testing conducted in this study, namely to use blackbox testing to test the functions on the website that made.*

**Keywords : Hotel, Waterfall, Dijkstra Algorithm**

## 1. PENDAHULUAN

Palangkaraya merupakan salah satu tujuan wisata di Indonesia yang memiliki jumlah hotel yang terus bertambah, berdasarkan data Dinas Pariwisata dan Ekonomi Kreatif Pemerintah Kota Palangka Raya, pada tahun 2015 terdapat 4 hotel berbintang dan 46 hotel berkelas melati. Sehingga, hal tersebut sering kali menimbulkan permasalahan bagi para wisatawan yaitu dalam mencari atau menentukan hotel berdasarkan jarak terpendek dengan tujuan untuk meminimalkan biaya transportasi maupun untuk mempercepat suatu perjalanan. Berdasarkan permasalahan tersebut, perlu dirancang suatu sistem yang dapat membantu para wisatawan untuk memberikan rekomendasi hotel berdasarkan jarak terpendek menuju lokasi hotel.

Pencarian jarak terpendek merupakan suatu masalah yang paling banyak dibahas dan dipelajari sejak akhir tahun 1950. Pencarian jarak terpendek ini telah diterapkan diberbagai bidang untuk mengoptimasi kinerja suatu sistem baik untuk meminimalkan biaya ataupun mempercepat jalannya suatu proses. Pencarian jarak terpendek termasuk dalam salah satu persoalan dalam teori *graf* yang berarti meminimalisasi bobot suatu lintasan dalam *graf*.

Metode yang dapat digunakan untuk menyelesaikan permasalahan pencarian jalur terpendek salah satunya adalah algoritma *Dijkstra*. Algoritma *Dijkstra* merupakan salah satu algoritma yang efektif dalam memberikan lintasan terpendek dari suatu lokasi ke lokasi yang lain. Prinsip dari algoritma *Dijkstra* adalah dengan pencarian dua lintasan yang paling kecil. Untuk menampilkan jalur tersebut, maka dibutuhkan sebuah aplikasi yang mampu menampilkan sebuah peta khususnya peta kota Palangkaraya dimana nantinya berguna untuk membimbing wisatawan menuju tempat tujuan [1].

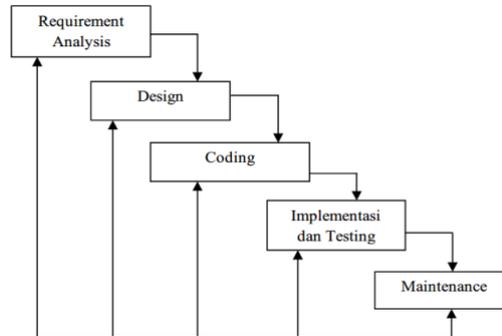
*Google maps* adalah layanan yang menampilkan peta secara *online* yang dapat diakses melalui *browser web* atau melalui perangkat *mobile*. Dengan *google maps* pengguna *mobile* dapat mengakses peta seluruh dunia secara *online* kapan saja asalkan perangkat *mobile* tersebut terkoneksi dengan internet [2]. Dengan *google map* ini diharapkan dapat membantu wisatawan dalam menampilkan peta lokasi hotel dan arah menuju lokasi hotel.

Disisi lain, perkembangan teknologi internet saat ini yang semakin pesat yang sangat membantu dalam kemudahan serta kecepatan pengiriman, penyampaian dan penerimaan informasi. Serta internet telah menjadi hal yang tidak terpisahkan dari kebutuhan sehari-hari. Sehingga, dengan adanya internet ini diharapkan mampu membantu meningkatkan jumlah reservasi kamar hotel.

Berdasarkan latar belakang tersebut maka dibuatlah sebuah website dengan judul "PEMBERIAN REKOMENDASI HOTEL BERDASARKAN JARAK DI KOTA PALANGKA RAYA BERBASIS WEB", dengan harapan website ini dapat membantu wisatawan agar lebih mudah mendapatkan rekomendasi hotel yang lokasinya dekat dengan wisatawan tanpa harus melihat satu persatu website hotel.

## 2. METODOLOGI PENELITIAN

Metodologi pengembangan yang digunakan dalam pembuatan website ini adalah metode *waterfall* [3]. Metode pengembangan system *waterfall* merupakan urutan kegiatan / aktivitas yang dilakukan dalam perancangan hingga pembuatan sistem. Fase-fase dalam *Waterfall* meliputi beberapa tahapan-tahapan yaitu :



Gambar 1. Diagram Alur Metodologi Waterfall  
Sumber : Sommerville,2011;30

Berikut ini langkah-langkah untuk mengembangkan sistem seperti yang ditentukan dalam Waterfall.

1. *Requirement Analysis* (analisis)

Proses penilaian sistem lama yang sedang berjalan dan studi kelayakan pengembangan system baru [4]. Untuk tahap analisis ini penulis menggunakan bisnis proses dan *flowchart* yang nantinya akan membantu penulis dalam menganalisis antara sistem lama dan sistem yang akan di usulkan.

2. *Design*

Pada tahapan ini akan di buat berbagai desain, yaitu :

- a. Desain Perancangan dengan DFD (*Data Flow Diagram*) [5].
- b. Desain Database dengan ERD (*EntityRelationship Diagram*) [6][7].
- c. *Site Map*
- d. Desain *UserInterface*.

3. *Coding*

Desain yang telah dibuatkan diubah kedalam bentuk yang dapat dimengerti oleh sistem komputer, yaitu kedalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap desain yang secara teknis akan dikerjakan oleh *programmer*.

4. *Implementasi Dan Testing*

Dalam melakukan proses *coding* juga dilakukan proses *testing* untuk menguji dan melihat kesalahan yang ada pada program maupun fungsi dari sistem, *testing* yang digunakan yaitu menggunakan *blackbox*.

5. *Maintenance*

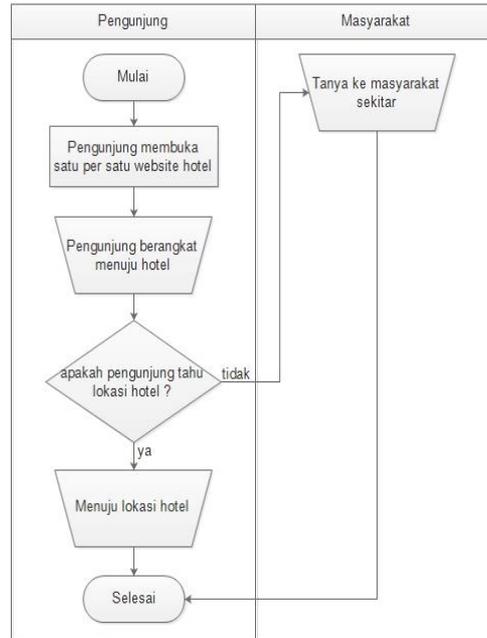
*Maintenance* (pemeliharaan) sangat diperlukan, termasuk di dalamnya adalah pengembangan, karena aplikasi sistem informasi yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada kesalahan kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada aplikasi tersebut.

**A. *Requirement Analysis***

Pada *website* Pemberian Rekomendasi Hotel Menggunakan Jarak Di Kota Palangka Raya, terdapat 2 pengguna yang terlibat didalam sistem yaitu admin dan pengunjung *web*. Kegiatan Admin di dalam *website* yaitu *Login* kedalam *website*, mengelola akun, mengelola data hotel, mengelola fasilitas, mengelola kelas kamar, mengelola data jalur gmap. Sedangkan kegiatan pengunjung didalam *website*, melihat daftar hotel, melihat daftar fasilitas, melihat informasi hotel, melihat lokasi hotel, melihat fasilitas hotel, dan melihat jalur menuju hotel yang akan dituju.

**B. *Bisnis proses sistem lama***

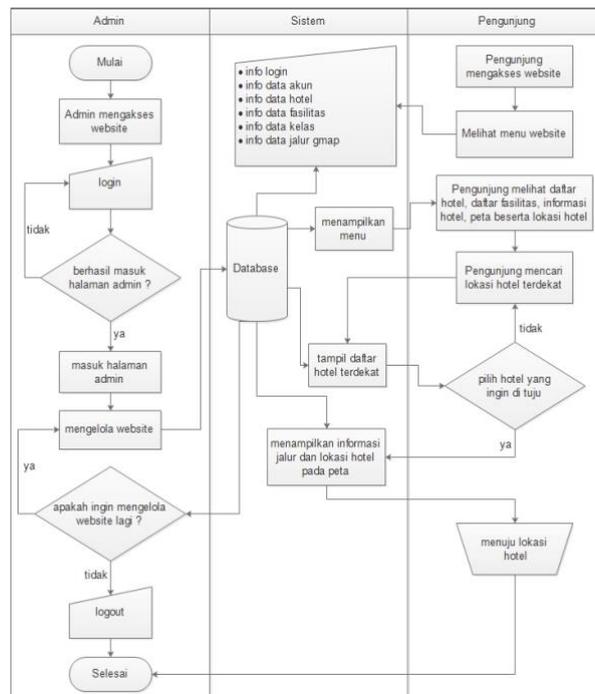
Berikut adalah deskripsi bisnis proses sistem lama serta gambaran *flowchart* bisnis proses sistem lama yang terdapat pada gambar 2 :



Gambar 2. Flowchart Bisnis Proses Sistem Lama

**C. Bisnis proses sistem baru**

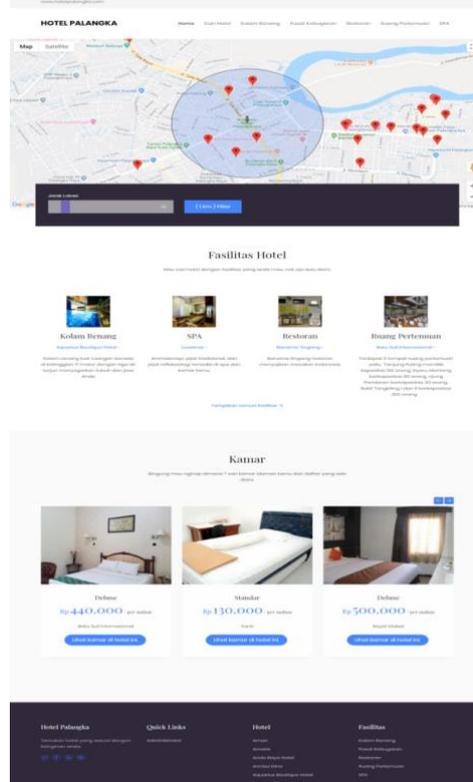
Dari kelemahan – kelemahan yang terdapat pada sistem lama diatas, didapatkan usulan rekomendasi sistem baru untuk merancang Pemberian Rekomendasi Hotel Menggunakan Jarak Di Kota Palangka Raya Berbasis Web. Munculnya ide untuk mengusulkan rekomendasi sistem baru ini adalah supaya informasi tidak hanya banyak diketahui oleh masyarakat yang tinggal di palangka raya melainkan juga harus diketahui oleh masyarakat luas atau wisatawan yang berkunjung ke palangka raya. Berikut adalah deskripsi bisnis proses sistem baru serta gambaran *flowchart* bisnis proses sistem baru yang terdapat pada gambar 3 :



Gambar 3. Flowchart Bisnis Proses Sistem Baru

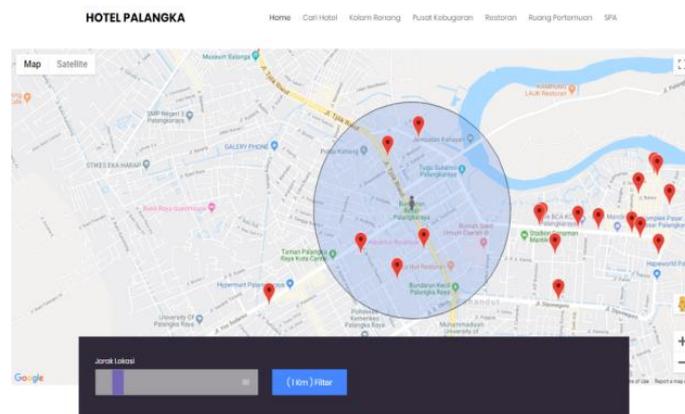
### 3. HASIL DAN PEMBAHASAN

Gambar 4 merupakan halaman utama yang berisi tentang aktivitas yang akan di lakukan pengunjung yang ada di website. Pada halaman ini terdapat fitur *google map* yang berisi informasi letak lokasi hotel yang ada di kota Palangka Raya. Di halaman utama ini juga terdapat daftar fasilitas dan juga kamar hotel. Berikut tampilan dari halaman utama.



Gambar 4. Halaman Utama

Sebelum masuk halaman rekomendasi, ada kegiatan yang harus di lakukan pengunjung di halaman utama yaitu menekan tombol filter dengan jarak radius yang dapat kita atur sesuai keinginan kita. Dimana jarak minimum dan maksimum radius telah di tetapkan oleh sistem yaitu 0,5 – 5 km.



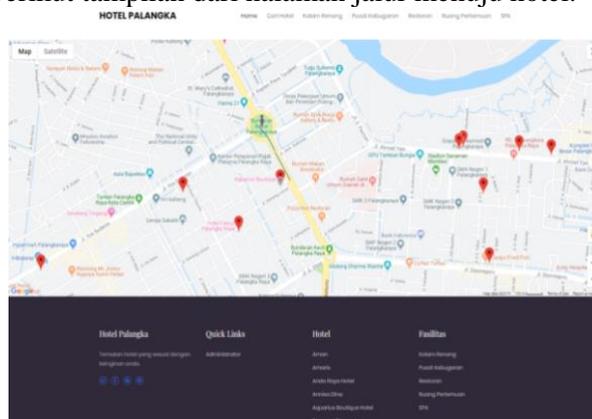
Gambar 5. Kegiatan Filter

Gambar 5 merupakan kegiatan filter. Setelah dilakukan filter, maka akan masuk kehalaman rekomendasi yang ada pada gambar 6. Halaman rekomendasi menampilkan informasi daftar hotel terdekat berdasarkan jarak radius. Berikut tampilan dari halaman rekomendasi.



Gambar 6. Halaman Rekomendasi

Gambar 7 merupakan halaman yang berisi tentang informasi jalur terdekat menuju hotel yang menggunakan algoritma dijkstra. Halaman ini merupakan tindak lanjut dari halaman rekomendasi, dimana pengunjung memilih salah satu hotel untuk menampilkan jalur terdekat menuju hotel tujuan. Berikut tampilan dari halaman jalur menuju hotel.



Gambar 7. Halaman Jalur Menuju Hotel Menggunakan Algoritma Dijkstra

### A. Algoritma Dijkstra

Algoritma *Dijkstra* merupakan salah satu varian dari algoritma *Greedy*, yaitu salah satu bentuk algoritma populer dalam pemecahan persoalan yang terkait dengan masalah optimasi. Sifatnya sederhana dan lempang (*straight forward*). Sesuai dengan artinya yang secara harafiah berarti tamak atau rakus (namun tidak dalam konteks negatif), algoritma *Greedy* ini hanya memikirkan solusi terbaik yang akan diambil pada setiap langkah tanpa memikirkan konsekuensi ke depan.

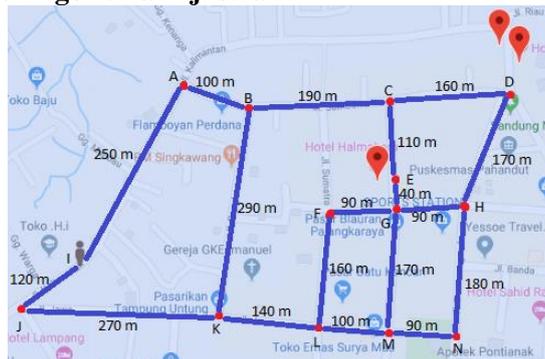
Intinya algoritma *Greedy* ini berupaya membuat pilihan nilai optimum lokal pada setiap langkah dan berharap agar nilai optimum lokal ini mengarah kepada nilai optimum global. Penggunaan strategi *Greedy* pada algoritma *Dijkstra* adalah pada setiap langkah, ambil sisi berbobot minimum yang menghubungkan sebuah simpul yang sudah terpilih dengan sebuah

simpul lain yang belum terpilih. Lintasan dari simpul asal ke simpul yang baru haruslah merupakan lintasan yang terpendek di antara semua lintasannya ke simpul simpul yang belum terpilih [1].

### B. Metode Graph

*Graf* adalah suatu kumpulan simpul (*nodes*) yang dihubungkan satu sama lain melalui sisi atau busur (*edges*) [8]. Secara informal, suatu *graf* adalah himpunan benda-benda yang disebut *verteks* atau *node* yang terhubung oleh *edge-edge*. Biasanya *graf* digambarkan sebagai kumpulan titik-titik (melambangkan *verteks*) yang dihubungkan oleh garis-garis (melambangkan *edge-edge*).

### C. Perhitungan Manual Algoritma Dijkstra



Gambar 8. Alur Graph

Gambar 8 diatas merupakan sektsa alur *graph*, yang dibutuhkan untuk menjadi acuan untuk perhitungan manual Dijkstra. Dimana akan dilakukan perjalanan dari node I (lokasi awal) ke node E (lokasi tujuan / Hotel Halmahera), berikut langkah-langkahnya :

1. Dari pemodelan graf diatas Dijkstra melakukan kalkulasi terhadap node tetangga yang terhubung langsung dengan node keberangkatan (node I). Berdasar kan gambar graf di atas, terdapat 2 jalur yang akan di bandingkan yaitu :

$$\text{Jalur I - A} = 0 + 250 = 250, \text{ dan}$$
$$\text{Jalur I - J} = 0 + 120 = 120$$

Nilai bobot diatas dibandingkan, dan berdasarkan perbandingan tersebut node J (I - J) memiliki nilai bobot yang paling kecil dari pada node A (I - A), nilai J (I - J) = 120. Sehingga, jalur yang akan di lakukan perhitungan terlebih dahulu yaitu jalur I - J

2. Node J (I - J) diset menjadi node keberangkatan dan ditandai sebagai node yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node terjamah yaitu node K.

$$\text{I - J - K} = 0 + 120 + 270 = 390$$

3. Sekarang node K (I - J - K) diset menjadi node keberangkatan dan ditandai sebagai node yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node terjamah.

$$\text{I - J - K - B} = 0 + 120 + 270 + 290 = 680$$
$$\text{I - J - K - L} = 0 + 120 + 270 + 140 = 530$$

Nilai bobot diatas dibandingkan, dan berdasarkan perbandingan tersebut node L (I - J - K - L) memiliki nilai bobot yang paling kecil dari pada node B (I - J - K - B), nilai L (I - J - K - L) = 530.

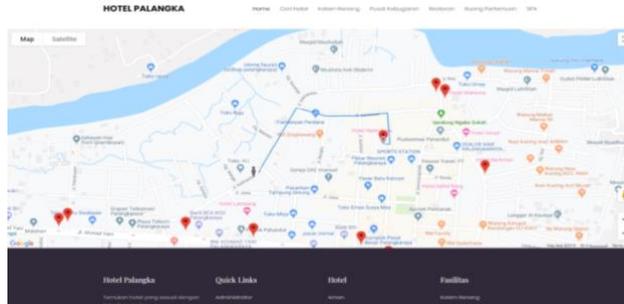
4. Node L (I - J - K - L) diset menjadi node keberangkatan dan ditandai sebagai node yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node terjamah.

$$\text{I - J - K - L - F} = 0 + 120 + 270 + 140 + 160 = 690$$
$$\text{I - J - K - L - M} = 0 + 120 + 270 + 140 + 100 = 630$$

- Nilai bobot diatas dibandingkan, dan berdasarkan perbandingan tersebut node M (I – J – K – L – M) memiliki nilai bobot yang paling kecil dari pada node F (I – J – K – L – F), nilai M (I – J – K – L – M) = 630.
5. Node M (I – J – K – L – M) diset menjadi node keberangkatan dan ditandai sebagai node yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node terjamah.  
 $I - J - K - L - M - G = 0 + 120 + 270 + 140 + 100 + 170 = 800$   
 $I - J - K - L - M - N = 0 + 120 + 270 + 140 + 100 + 90 = 720$   
Nilai bobot diatas dibandingkan, dan berdasarkan perbandingan tersebut node N (I – J – K – L – M – N) memiliki nilai bobot yang paling kecil dari pada node G (I – J – K – L – M – G), nilai N (I – J – K – L – M – N) = 720.
6. Node N (I – J – K – L – M – N) diset menjadi node keberangkatan dan ditandai sebagai node yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node terjamah yaitu node H.  
 $I - J - K - L - M - N - H = 0 + 120 + 270 + 140 + 100 + 90 + 180 = 900$
7. NodeH (I – J – K – L – M – N – H) diset menjadi node keberangkatan dan ditandai sebagai node yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node terjamah.  
 $I - J - K - L - M - N - H - G = 0 + 120 + 270 + 140 + 100 + 90 + 180 + 90 = 990$   
 $I - J - K - L - M - N - H - D = 0 + 120 + 270 + 140 + 100 + 90 + 180 + 170 = 1070$   
Nilai bobot diatas dibandingkan, dan berdasarkan perbandingan tersebut node G (I – J – K – L – M – N – H – G) memiliki nilai bobot yang paling kecil dari pada node D (I – J – K – L – M – N – H – D), nilai G (I – J – K – L – M – N – H – G) = 990.
8. NodeG (I – J – K – L – M – N – H – G) diset menjadi node keberangkatan dan ditandai sebagai node yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node terjamah. Karena Node E merupakan node tujuan, maka langsung menuju node E  
 $I - J - K - L - M - N - H - G - E = 0 + 120 + 270 + 140 + 100 + 90 + 180 + 90 + 40 = 1030$   
Bobot nilai dari jalur I – J – K – L – M – N – H – G – E yaitu 1030
9. Jalur selanjutnya yaitu node A (I – A = 250), NodeA (I – A) diset menjadi node keberangkatan dan ditandai sebagai node yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node terjamah yaitu node B.  
 $I - A - B = 0 + 250 + 100 = 350$
10. NodeB (I – A – B) diset menjadi node keberangkatan dan ditandai sebagai node yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node terjamah.  
 $I - A - B - C = 0 + 250 + 100 + 190 = 540$   
 $I - A - B - K = 0 + 250 + 100 + 290 = 640$   
Nilai bobot diatas dibandingkan, dan berdasarkan perbandingan tersebut node C (I – A – B – C) memiliki nilai bobot yang paling kecil dari pada node K (I – A – B – K), nilai C (I – A – B – C) = 540
11. NodeC (I – A – B – C) diset menjadi node keberangkatan dan ditandai sebagai node yang telah terjamah. Dijkstra melakukan kalkulasi kembali terhadap node-node tetangga yang terhubung langsung dengan node terjamah. Karena Node E merupakan node tujuan, maka langsung menuju node E  
 $I - A - B - C - E = 0 + 250 + 100 + 190 + 110 = 650$   
Bobot nilai dari jalur I – A – B – C – E yaitu 650
12. Setelah Kedua jalur tersebut di lakukan perhitungan, maka hasil dari nilai bobot kedua jalur tersebut di bandingkan :  
Jalur I – A – B – C – E = 0 + 250 + 100 + 190 + 110 = 650  
Jalur I – J – K – L – M – N – H – G – E = 0 + 120 + 270 + 140 + 100 + 90 + 180 + 90 + 40 = 1030

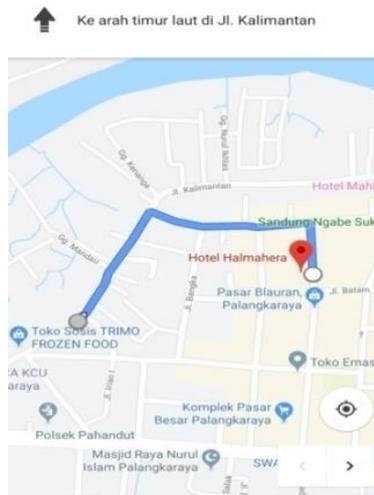
Berdasarkan hasil nilai bobot kedua jalur tersebut dapat dinyatakan bahwa jalur I – A – B – C – E memiliki nilai bobot yang paling kecil dari pada jalur I – J – K – L – M – N – H – G – E. Sehingga, berdasarkan perhitungan Dijkstra perjalanan dari node I (Lokasi awal) ke node E (lokasi tujuan / Hotel Halmahera) dinyatakan selesai dengan jalur terpendeknya adalah I – A – B – C – E dengan nilai bobot 650m

Gambar 9 di bawah ini merupakan hasil dari perhitungan Dijkstra dengan lokasi yang dituju yaitu Hotel Halmahera.



Gambar 9. Pengujian Perjalanan Dari Node I (lokasi awal) Ke Node E (lokasi tujuan / hotel halmahera) Dengan Algoritma Dijkstra

Berikut ini gambar 10 merupakan gambar hasil percobaan menggunakan *google map* dengan lokasi awal keberangkatan dan lokasi tujuan yang sama dengan perhitungan Dijkstra sebelumnya.



Gambar 10. Hasil Percobaan Menggunakan *Google Map*

Berdasarkan hasil dari kedua percobaan yang telah di lakukan dengan lokasi awal keberangkatan dan lokasi tujuan yang sama, ternyata memiliki hasil yang sama antara sistem atau perhitungan yang dibuat menggunakan algoritma Dijkstra dengan hasil yang di dapat dari *google map*.

#### 4. KESIMPULAN

Berdasarkan proses “Pemberian Rekomendasi Hotel Menggunakan Jarak Di Kota Palangka Raya Berbasis Web” maka dapat diambil kesimpulan bahwa dalam merancang sistem yang dapat memberikan rekomendasi hotel yang ada di Kota Palangka Raya berdasarkan jarak terpendek menuju lokasi hotel, ada beberapa tahapan yang harus dilakukan menyesuaikan dengan Metodologi *Waterfall* yaitu *Requirement analysis*, *Design*, *Coding*, *Implementasi* dan *testing*, dan

*maintenance*. Untuk menentukan jarak terpendek pada “Pemberian Rekomendasi Hotel Menggunakan Jarak Di Kota Palangka Raya Berbasis Web” menggunakan penerapan algoritma dijkstra.

#### DAFTAR PUSTAKA

- [1] Novandi, Raden Aprian Diaz. “Perbandingan Algoritma Dijkstra dan Algoritma Floyd-Warshall dalam Penentuan Lintasan Terpendek (Single Pair Shortest Path),” Bandung : Institut Teknologi Bandung, 2007.
- [2] Ren Peng, Zhong Hsiang Tsou, Ming. “Internet GIS: Distributed Geographic Information Service for the Internet and Wireless Networks,” John Wiley & Sons, 2003.
- [3] Sommerville, Ian. “Software Engineering (Rekayasa Perangkat Lunak),” Jakarta : Erlangga, 2011.
- [4] Soerherman, Bonnie, dan Pinontoan, Marion. “Merancang Sistem Informasi Konsep & Kasus dengan Visio,” Jakarta : PT Elex Media Komputindo, 2008.
- [5] Hanif Al Fatta. “Analisis dan Perancangan Sistem Informasi,” Yogyakarta : Andi, 2007.
- [6] Adi Nugroho. “Konsep Pengembangan Sistem Basis Data,” Cetakan pertama, Bandung, Penerbit : Informatika, 2004.
- [7] Simarmata, Janner dan Paryudi, Iman. “Basis Data,” Yogyakarta : Andi, 2010.
- [8] Zakaria. “Teknologi Informasi dan Komunikasi,” Jakarta : Arya Duta, 2006.