

## IMPLEMENTASI ALGORITMA DIJKSTRA PADA APLIKASI WISATA KUNINGAN BERBASIS ANDROID

Elda Candra Galih<sup>1</sup>, Rio Andriyat Krisdiawan<sup>2</sup>

Fakultas Ilmu Komputer Universitas Kuningan

Jl. Cut Nyak Dien No.36 A, Kel. Cijoho Kuningan 45513

[eldacandragalih95@gmail.com](mailto:eldacandragalih95@gmail.com)<sup>1</sup>, [rioandriyat@uniku.ac.id](mailto:rioandriyat@uniku.ac.id)<sup>2</sup>

Abstrak - Kabupaten Kuningan adalah salah satu tujuan wisata yang cukup populer di kalangan wisatawan. Kuningan memiliki banyak objek wisata yang potensial untuk wisata alam, sejarah, jiarah, maupun penelitian. Untuk menemukan destinasi pariwisata, wisatawan biasanya mencari aplikasi mesin pencari atau peta wisata sehingga memakan waktu yang lama. Untuk memudahkan wisatawan dalam mencari destinasi wisata, perlu sebuah aplikasi. Aplikasi yang akan dikembangkan berdasarkan penelitian yang diadakan di Dinas Pemuda Olahraga dan Pariwisata Kabupaten Kuningan. Aplikasi berbasis mobile diperlukan dalam menemukan lokasi objek wisata dan menyediakan deskripsi wisata, dan fasilitas yang ada di objek wisata. Untuk mencari rute terdekat, akan ditampilkan dalam bentuk peta. Wisatawan dapat memilih lokasi tempat wisata dan lokasi awal wisatawan, maka sistem akan menampilkan hasil rute yang ada di peta. Dan dalam aplikasi ini juga tersedia layanan untuk mencari lokasi wisatawan menggunakan teknologi GPS. Dan penulis menggunakan algoritma Dijkstra untuk menentukan rute terpendek dalam mencari rute dari lokasi awal ke lokasi yang diinginkan dari objek pariwisata. Penyimpanan data dalam aplikasi ini, penulis menggunakan database *MySQL* dan *Apache* sebagai web server dalam menyimpan data objek dan data node tujuan dan rute yang bersangkutan dengan mencari lokasi objek wisata. Dan pada pengembangan aplikasi ini berbasis *client server*.

**Kata Kunci** : *Algorithma Dijkstra*, Pencarian rute terdekat, Aplikasi berbasis *Mobile*, Kabupaten Kuningan, Objek Wisata.

*Abstract - Kuningan Regency is one of the tourist destinations that is quite popular among tourists. Kuningan has many potential tourism object whether it is natural tourism, history, jiarah, and artificial. To find tourism destinations, tourists are usually looking at search engine applications or tourist maps so it takes a long time. To facilitate tourists in finding tourist destinations, it needs an application. The application that will be developed based on preliminary research held at Dinas Pemuda Olahraga dan Pariwisata Kabupaten Kuningan. Mobile-based application is needed in finding the location of tourisism objects and it provides descriptions of attractions, and facilities that exist in the tourism object. It is created only to show the location of existing tourism object in Kuningan Regency. In finding the closest route, it will be displayed in the form of maps. Tourists can choose the location of tourist attractions and initial location of tourists, then the system will display the results of the existing route on the maps. And in this application is also available services to find the location of tourists using GPS technology. And the author uses Dijkstra algorithm to determine the shortest route in finding the route from the initial location to the desired location of the object of tourism. And in the data storage in this application, the author uses MySQL or Apache as a web server in storing data objects and data node,, destination and route concerned with searching the location of the tourist attraction. And on the development of this application he uses client server.*

**Keywords:** *Dijkstra Algorithma, Nearest Order, Mobile-based Application, Kuningan District, Tourism Object.*

### 1. PENDAHULUAN

#### 1.1. Latar Belakang

Wisata adalah suatu proses bepergian yang bersifat sementara yang dilakukan seseorang untuk menuju tempat lain diluar tempat tinggalnya. Wisata dapat dilakukan

dengan tujuan hiburan, *edukasi*, maupun yang lainnya. Wisata dapat dibagi dalam beberapa kategori, misalkan wisata alam, wisata pendidikan, dan wisata buatan seperti halnya kolam renang, waduk, situ, dan lain sebagainya. Setiap

daerah pasti memiliki tempat atau objek wisata.

Teknologi informasi juga dibutuhkan untuk memberikan informasi mengenai wisata yang ada di suatu daerah. Teknologi informasi digunakan untuk memberi informasi objek wisata yang ada di daerah tersebut kepada wisatawan yang akan berkunjung. Informasi wisata dapat berupa informasi teks ataupun informasi berupa visual atau gambar yang di publikasikan di media sosial maupun media lainnya.

Salah satu daerah yang dijadikan tujuan wisata yaitu Kabupaten Kuningan. Kabupaten Kuningan terletak di wilayah selatan Kabupaten Cirebon. Kabupaten Kuningan memiliki banyak objek wisata yang sudah banyak dikunjungi oleh wisatawan lokal maupun mancanegara. Tidak dipungkiri dengan semakin banyaknya wisatawan dari luar daerah maka informasi tentang tempat-tempat wisata yang ada di Kabupaten Kuningan banyak dibutuhkan oleh wisatawan yang akan berkunjung.

Masih kurangnya aplikasi yang memberikan informasi wisata yang ada di Kabupaten Kuningan, sehingga wisatawan sulit mendapatkan informasi tempat-tempat wisata yang akan dikunjungi. Selama ini informasi yang didapat wisatawan sebatas informasi gambar dan video saja melalui media sosial, sehingga belum terpusatnya informasi wisata yang mencakup lokasi, rute, gambar dan video.

Salah satu pemanfaatan Teknologi Informasi dalam mempermudah pencarian lokasi wisata yaitu dengan membuat sebuah aplikasi. Aplikasi yang dibangun dapat memberikan

informasi tentang tempat-tempat wisata yang ada di Kabupaten Kuningan. Dalam membangun sebuah aplikasi untuk mencari lokasi wisata salah satunya bisa menggunakan *Algoritma Dijkstra* dalam pencarian rute. *Algoritma Dijkstra* merupakan salah satu *algoritma* yang efektif dalam memberikan lintasan terpendek dari suatu lokasi ke lokasi lainnya. Prinsip dari *Algoritma Dijkstra* adalah dengan pencarian dua lintasan yang paling kecil. *Algoritma Dijkstra* memiliki iterasi untuk mencari titik yang jaraknya dari titik awal adalah paling pendek. Pada setiap iterasi, jarak titik memberikan jarak terpendek. Syarat *algoritma* ini adalah bobot sisinya yang harus negatif.

Berdasarkan uraian diatas maka penulis melakukan penelitian dengan judul “**Implementasi Algoritma Dijkstra Pada Aplikasi Wisata Kuningan Berbasis Android**”.

## 1.2. Identifikasi Masalah

1. Masih kurangnya aplikasi yang memberikan informasi tentang foto, *dekripsi* objek wisata, tarif objek wisata, dan lokasi dalam satu aplikasi.
2. Masih kurangnya penerapan teknologi *mapping* pada aplikasi wisata yang ada, sehingga wisatawan sulit mendapatkan informasi jarak terpendek dari satu lokasi ke lokasi wisata yang akan dikunjungi di Kabupaten Kuningan.
3. Belum terpusatnya informasi mengenai objek wisata yang ada di Kabupaten Kuningan, sehingga informasi yang didapat oleh wisatawan masih kurang tepat.
4. Masih kurangnya aplikasi wisata yang mengimplementasikan *algoritma dijkstra* untuk menentukan jarak terpendek dalam mencari lokasi objek wisata.

### 1.3. Batasan Masalah

1. Aplikasi yang dibangun difokuskan untuk menentukan jarak terpendek dari satu lokasi ke lokasi objek wisata yang akan dikunjungi wisatawan di Kabupaten Kuningan.
2. Aplikasi ini menggunakan *Algoritma Dijkstra* untuk menentukan jarak terpendek.
3. Bahasa pemrograman yang digunakan yaitu bahasa pemrograman *java* dan menggunakan *tools* eclipse juno, SDK dan ADT sebagai alat untuk membangun aplikasi android pada Aplikasi Wisata Kuningan.
4. Aplikasi dibuat berbasis *client server* dengan *database* MySQL, dengan *Back-End* PHP.
5. Untuk fitur-fitur yang dibangun oleh penulis, sebagai berikut:
  - a) Rute  
Untuk fitur rute penulis ingin menampilkan rute, seperti ; rute terdekat dari lokasi awal ke lokasi tujuan yang sudah di hitung menggunakan algoritma Dijkstra. Agar dapat mempermudah wisatawan memilih rute yang akan dilalui dengan menampilkan *image* atau *graph* sebagai *output*.
  - b) Deskripsi  
Penulis membuat atau memasukan deskripsi atau keterangan dan rincian tentang objek wisata yang ada di kabupaten kuningan.
6. Sedangkan untuk menentukan lokasi objek wisata penulis mengambil dari teknologi yang sudah ada yaitu *google maps*.
7. Aplikasi ini tidak terpengaruh oleh kemacetan dan keadaan rute yang akan dilalui karena aplikasi ini digunakan hanya untuk mencari rute terpendek saja.
8. Aplikasi ini hanya menampilkan objek wisata yang ada pada buku "Profil Pariwisata Kuningan" terbit tahun 2015.
9. Pengujian aplikasi menggunakan *smartphone* dengan *operating system lollipop* (API Level 21).

## 2. METODE PENELITIAN

### 2.1. Metode Pengumpulan Data

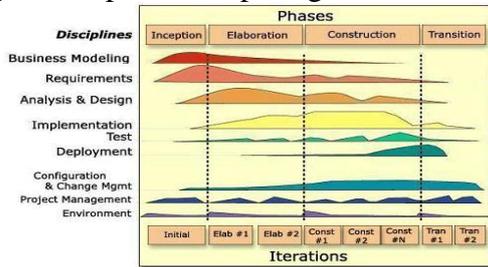
Dalam melakukan penelitian skripsi ini, penulis menggunakan beberapa tahapan, diantaranya:

1. Observasi  
Yaitu dengan melakukan observasi ke Dinas Pemuda, Olahraga dan Pariwisata selaku Dinas yang menaungi pariwisata Kabupaten Kuningan dan berkunjung ke tempat objek wisata yang akan dimasukan pada daftar objek wisata yang ada pada aplikasi, sehingga langkah-langkah yang dikerjakan dalam dalam membuat aplikasi tersebut bisa dipahami secara tepat dan sesuai dengan kinerja yang ada.
2. Wawancara  
Metode ini dilakukan dengan cara tanya jawab dengan pihak Dinas terkait yang akan dijadikan sumber informnsi mengenai objek wisata yang ada di Kabupaten Kuningan. Penulis melakukan wawancara dengan Bapak Agus Nur Vitriana M, SE. (Kasi Kabag. *Destination*).
3. Studi Pustaka  
Mencari referensi yang berkaitan dengan *Algoritma Dijkstra*, mulai dari buku-buku, jurnal maupun artikel dan sumber-sumber lain di internet.

### 2.2. Metode Pengembangan Sistem

*Relational Unified Process* (RUP) adalah pendekatan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang lebih baik (*well defined*) dan penstrukturan yang baik (*well structured*). RUP menyediakan

pendefinisian yang baik untuk alur hidup proyek perangkat lunak. Proses pengulangan/iterative pada RUP secara global dapat dilihat pada gambar berikut:



**Gambar 1.** RUP life cycle (Daniel Siahaan, 2012)

Penjelasan singkat empat tahapan dalam RUP adalah sebagai berikut:

1. **Inception** merupakan tahapan ini merupakan tahapan paling awal dimana pada tahapan ini dilakukan proses pengumpulan data yang akan dilakukan pada penelitian ini yaitu dengan cara observasi ke Dinas Pemuda, Olahraga dan Pariwisata, wawancara dengan Bapak Agus Nur Vitriana M, SE. (Kasi Kabag. *Destination*), dan studi pustaka (jurnal, buku, *e-book*, dll).
2. **Elaboraion** tahap ini lebih difokuskan pada perencanaan arsitektur sistem, serta dapat mendeteksi resiko yang mungkin terjadi pada proyek. Ditahap ini pula akan mulai dilakukan rancang bangun perangkat lunak secara iterative melalui aktifitas-aktifitas seperti *business modeling*, *requrement*s, *analysis* dan *design*. Sistem dirancang menggunakan UML, algoritma yang cocok digunakan adalah algoritma dijkstra yang diterapkan untuk mencari rute terpendek dari satu lokasi ke lokasi yang akan dituju.
3. **Construction** merupakan Pada tahap *contruction* ini merupakan tahap untuk membangun perangkat lunak sampai dengan saat perangkat lunak siap digunakan. Serta lebih difokuskan pada pengembangan komponen dan fitur-fitur sistem yang

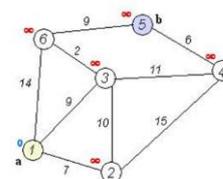
akan dibangun. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Bahasa pemrograman yang digunakan untuk pembuatan aplikasi ini menggunakan Bahasa pemrograman *java*. Tahap ini menghasilkan produk yang akan diserahkan kepada *user*.

4. **Transition** merupakan Tahap *transition* ini merupakan tahap terakhir dimana difokuskan pada bagaimana cara menyampaikan perangkat lunak yang sudah jadi kepada *user*. Aktifitas yang dilakukan pada tahap ini yaitu memberikan arahan atau *tutorial* kepada *user*, pemeliharaan dan pengujian apakah sistem sudah selesai dan sudah sesuai dengan harapan *user*

### 2.3. Algoritma Dijkstra

Algoritma *Dijkstra* merupakan salah satu algoritma yang efektif dalam memberikan lintasn terpendek dari suatu lokasi ke lokasi lainnya. Prinsip dari Algoritma *Dijkstra* adalah dengan pencarian dua lintasan yang paling kecil. Algoritma *Dijkstra* memiliki iterasi untuk untuk mencari titik yang jaraknya dari titik awal adalah paling pendek. Pada setiap iterasi, jarak titik memberikan jarak terpendek. Syarat algoritma ini adalah bobot sisinya yang harus negatif. (Adi Nugroho, 2009)

Algoritma ini bertujuan untuk menemukan jalur terpendek berdasarkan bobot terkecil dari satu titik kr titik lainnya. Misalkan titik menggambarkan gedung dan garis menggambarkan jalan, maka Algoritma *Dijkstra* melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik.



**Gambar 2.** Contoh keterhubungan antar titik dalam *Algoritma Dijkstra*

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Flowchart Algoritma dan Langkah-langkah Algoritma Dijkstra

1. Beri bobot (jarak) untuk setiap titik ke titik lainnya, lalu set nilai 0 pada *node* awal dan nilai tak hingga terhadap *node* lain (belum terisi).
2. Set semua *node* “belum terjamah” dan set *node* awal sebagai *node* keberangkatan”
3. Dari *node* keberangkatan, pertimbangkan *node* tetangga yang belum terjamah dan hitung jaraknya dari titik keberangkatan. Sebagai contoh, jika titik keberangkatan A ke B memiliki bobot jarak 6 dan dari B ke *node* C berjarak 2, maka jarak ke C melewati B menjadi  $6+2=8$ . Jika jarak ini lebih kecil dari jarak sebelumnya (yang telah terekam sebelumnya) hapus data lama, simpan ulang data jarak dengan jarak yang baru.
4. Saat kita selesai mempertimbangkan setiap jarak terhadap *node* tetangga, tandai *node* yang telah terjamah sebagai “*node* terjamah”. *Node* terjamah tidak akan pernah di cek kembali, jarak yang disimpan adalah jarak terakhir dan yang paling minimal bobotnya.
5. Set “*node* belum terjamah” dengan jarak terkecil (dari *node* keberangkatan sebagai “*node* keberangkatan” selanjutnya dan lanjutkan dengan kembali ke step 3.

Pseudocode dijkstra ( $w, a, z, L$ )

$L(a) := 0$

For semua verteks  $x \neq a$  do

$L(x) := \infty$

$T :=$  himpunan semua verteks

// T adalah himpunan verteks yang panjang terpendeknya dari a belum ditemukan

While  $z \in T$  do

Begin

Pilih  $v \in T$  dengan minimum  $L9v$ )

$T := T - \{v\}$

For setiap  $x \in T$  di samping  $v$  do

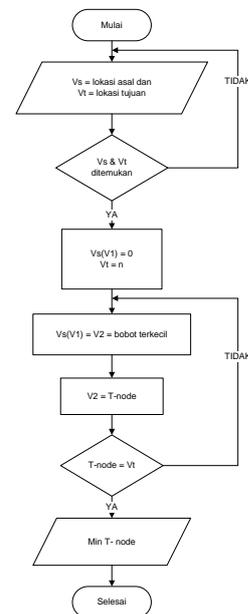
$L9x := \min\{L9x, L9v\} + w(v, x)$

end

end dijkstra

Jadi, pada dasarnya algoritma *Dijkstra* memberikan lintasan terpendek dari suatu lokasi ke lokasi lainnya.

Berikut adalah *flowchart* algoritma *Dijkstra* dapat dilihat pada gambar 3



**Gambar 3.** *Flowchart* Algoritma *Dijkstra*

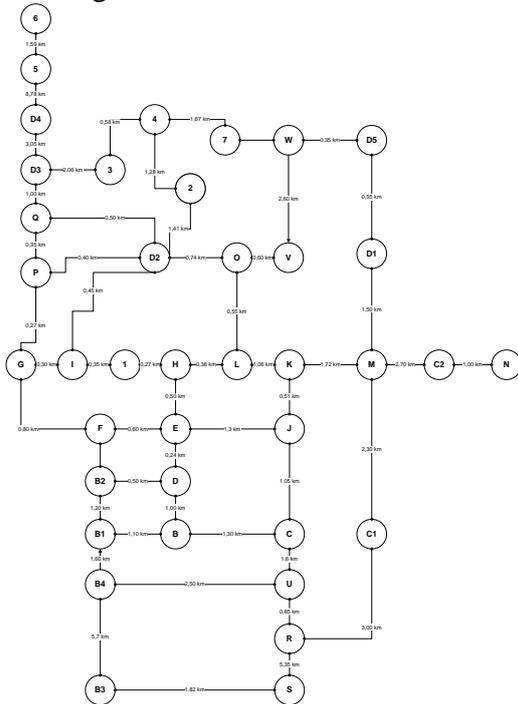
#### 3.2. Analisis Dijkstra

Analisis Pencarian lokasi objek wisata pada aplikasi wisata kuningan dengan menggunakan algoritma *Dijkstra* :

1. Menentukan *node-node* yang ada di wilayah Kabupaten Kuningan.
2. Mengambil Sampel untuk perhitungan yaitu sekitaran lokasi ala dan lokasi tujuan.
3. Memberikan inisialisasi pada setiap nodenya.
4. Menghitung setiap perpindahan dari satu *node* ke *node* lainnya.

Contoh kasus yang diambil oleh penulis dalam melakukan pencarian rute dari terminal kertawangunan menuju lokasi objek wisata kolam renang sanggariang.

Didapat map secara umum pada gambar 4, sebagai berikut



**Gambar 4.** Simple Graph Node

Kemudian setelah di dapatkan graf node secara umum maka kemudian dilakukan inisialisasi untuk setiap nodenya. Seperti pada tabel 1, berikut ini :

**Tabel 1.** Tabel keterangan node

No	Node	Nama Tempat
1	B	Simpang Windusengkahan
2	C	Simpang Ciporang
3	D	Simpang Jalan Baru
4	E	Simpang Yamsik
5	F	Simpang Pasar Baru
6	G	TAMKOT
7	H	Simpang Coloumbus Aruji
8	I	Citamba
9	J	Simpang Juanda
20	K	Lampu Merah Cijoho
11	L	Pemda Kuningan
12	M	Lampu Merah Cirendang
13	N	Lampu Merah Jalaksana
14	O	Simpang 4 pramuka
15	P	Lampu Merah Gotong Royong
16	Q	Simpang 4 Veteran
17	R	Terminal Kertawangunan
18	S	Lampu Merah Oleced
19	T	Jalan Baru Awirarangan
20	U	Pasar Ancaran
21	V	Bundaran Jalan Baru

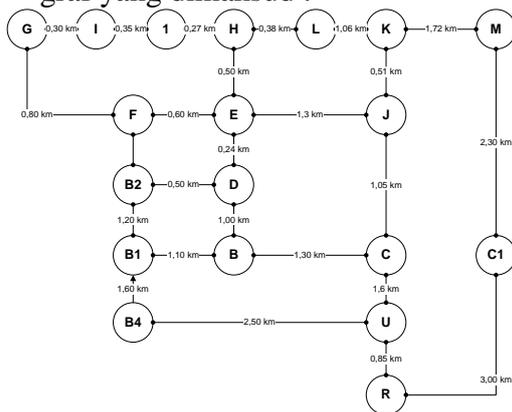
22	W	Simpang Jalan Baru Gunungkeling
23	B1	Winduhaji
24	B2	Awirarangan
25	B3	Tambakbaya
26	B4	Karangtawang
27	C1	Kedungarum
28	C2	Kramatmulya
29	D1	Cirendang
30	D2	Pramuka
31	D3	Sukamulya
32	D4	Kadugede
33	D5	Gunung Keling
34	D6	Ragawacana
35	1	Kolam Renang Sanggariang
36	2	Hutan Kota Mayasih
37	3	Paseban Tri Panca Tunggal
38	4	Kolam renang Cigugur
39	5	Waduk Darma
40	6	Darmaloka
41	7	Taman Purbakala Cipari

Dari uraian pemodelan graf diatas, dapat diambil *Graf* secara sederhana dari lokasi awal menuju lokasi objek wisata yang akan dikunjungi. Kemudian tentukan ada berapa jalur ang akan dilalui dari kertawangunan menuju kolam renang sanggariang. Didapatkan dari *graf* diatas rute dari terminal kertawangunan menuju kolam bernang sanggariang dapat dilalui melalui empat rute, yaitu :

1. Rute pertama yaitu dari Terminal Kertawangunan , ke Pasar Ancaran, ke Simpang Ciporang, ke Simpang Juanda, ke Lampu Merah Cijoho, ke Pemda Kuningan, ke Simpang Coloumbus Aruji, lalu terakhir ke lokasi objek wisata Sanggariang. Pada graf jalur tersebut dinotasikan  $R \rightarrow U \rightarrow C \rightarrow J \rightarrow K \rightarrow L \rightarrow H \rightarrow 1$
2. Rute kedua yaitu dari Terminal Kertawangunan , ke Pasar Ancaran, ke Karangtawang, ke Winduhaji, ke Awirarangan, ke Simpang Pasar Baru, ke Tamkot, ke Citamba, lalu terakhir ke lokasi objek wisata Sanggariang. Pada graf jalur tersebut dinotasikan  $R \rightarrow U \rightarrow B4 \rightarrow B1 \rightarrow B2 \rightarrow F \rightarrow G \rightarrow I \rightarrow H \rightarrow 1$

3. Rute ketiga yaitu dari Terminal Kertawangunan , ke Pasar Ancaran, ke Simpang Ciporang, ke Simpang Windusengkahan, ke Simpang Jalan Baru, ke Simpang Yamsik, ke Simpang Coloumbus Aruji, lalu terakhir ke lokasi objek wisata Sanggariang. Pada graf jalur tersebut dinotasikan  $R \rightarrow U \rightarrow C \rightarrow B \rightarrow D \rightarrow E \rightarrow H \rightarrow 1$
4. Rute keempat yaitu dari Terminal Kertawangunan , ke Kedungarum, ke Lampu Merah Cirendang, ke Lampu Merah Cijoho, ke Pemda Kuningan, ke Simpang Coloumbus Aruji, lalu terakhir ke lokasi objek wisata Sanggariang. Pada graf jalur tersebut dinotasikan  $R \rightarrow C1 \rightarrow M \rightarrow K \rightarrow L \rightarrow H \rightarrow 1$

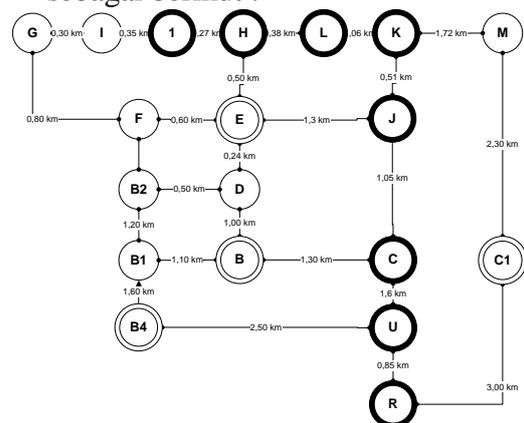
Dari keempat jalur rute diperoleh sampel node yang dapat dilalui dari Awirarangan ke Kolam Renang Sanggariang yaitu R, U, C, J, K, B4, B1, B2, F, G, I, B, D, E, C1, M, L, H dan 1, dari sempel node-node tersebut dapat dibuat sederhana yang hanya mencakup node-node dan jalur yang mengarah ke node 1. Berikut ini graf yang dimaksud :



**Gambar 5.** Graf sederhana sebagai sempel

Dari graf tersebut selanjutnya sistem akan melakukan perhitungan dengan algoritma dijkstra. Berikut ini uraian langkah-langkah pada algoritma dijkstra pada kasus tersebut.

1. Langkah pertama input berupa node awal, node tujuan dan graf. Dalam hal ini node awal yaitu node **R**, node tujuan **1** sementara input graf sederhana yang terdiri **R, U, C, J, K, B4, B1, B2, F, G, I, B, D, E, C1, M, L, H** dan **1**.
2. Langkah kedua yaitu menetapkan awal sebagai status ditemukan dan kemudian dikunjungi. Dalam hal ini node **R** berstatus ditemukan dan dikunjungi. Sementara node berikut berstatus belum ditemukan atau belum dikunjungi.
3. Langkah ketiga yaitu dilakukan pencarian terhadap setiap node yang dapat dicapai secara langsung yang dapat dikunjungi. Karena saat ini node yang sedang dikunjungi node **R** maka dilakukan pencarian terhadap semua node yang dapat dicapai langsung dari setiap node **R** dalam hal ini ditemukan dua node yaitu node **U** dan **C1** dengan memiliki akumulasi bobot saat ini yaitu  $U = 0,85$  km dan  $C1 = 3,00$  km. Bobot akumulasi adalah total bobot yang dicapai dari node awal ke node tersebut melalui rute yang sedang ditelusuri.
4. Langkah keempat, dan seterusnya, sehingga menghasilkan Node graf sebagai berikut :



**Gambar 6.** kondisi Graf pada saat Node H dikunjungi serta node 1 ditemukan dan dikunjungi sebagai node tujuan.

Karena node H dikunjungi serta node 1 ditemukan dan dikunjungi beagai node tujuan maka node **R, U, C, J, K, L, H → 1** memiliki bobot nilai : 5,45 km + 0,27 km = 5,72 km, seperti pada tabel 2, berikut ini :

**Tabel 2.** Kondisi node H dan node 1 Dikunjungi

Node	Status	Bobot	Rute
<b>R (Asal)</b>	Dikunjungi	0	R
<b>U</b>	Ditemukan	0,85 km	R - U
<b>C</b>	Ditemukan	1,60 km	R - U - C
<b>J</b>	Dikunjungi	1,05 km	R - U - C - J
<b>K</b>	Dikunjungi	0,51 km	R - U - C - J - K
<b>B4</b>	Ditemukan	2,50 km	R - U - B4
<b>B1</b>	Belum Ditemukan	-	-
<b>B2</b>	Belum Ditemukan	-	-
<b>F</b>	Belum Ditemukan	-	-
<b>G</b>	Belum Ditemukan	-	-
<b>I</b>	Belum Ditemukan	-	-
<b>B</b>	Ditemukan	1.30 km	R - U - C - B
<b>D</b>	Belum Ditemukan	-	-
<b>E</b>	Belum Ditemukan	1,30 km	R - U - C - J - E
<b>C1</b>	Ditemukan	3,00 km	R → C1
<b>M</b>	Belum Ditemukan	-	-
<b>L</b>	Dikunjungi	1,06 km	R - U - C - J - K - L
<b>H</b>	Dikunjungi	0,38 km	R - U - C - J - K - L - H
<b>1 (Tujuan)</b>	Dikunjungi	0,27	R - U - C - J - K - L - H - 1

Dari uraian diatas algoritma dijkstra tersebut dapat disimpulkan sebagai berikut :

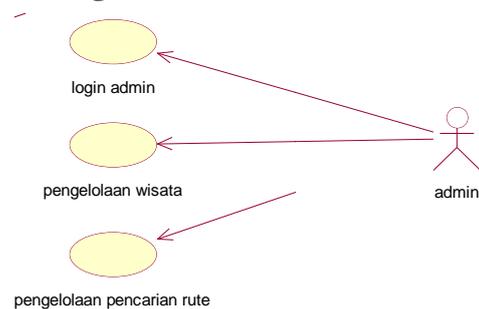
1. Dari uraian diatas didapatkan ;
  - a. Rute pertama ; **R → U → C → J → K → L → H → 1** dengan bobot jarak : 0,85 km + 1,60 km + 1,05 km + 0,51 km + 1,06 km + 0,38 km + 0,27 km = 5,72 km

- b. Rute kedua ; **R → U → B4 → B1 → B2 → F → G → I → H → 1** dengan bobot jarak : 0,85 km + 2,50 km + 1,60 km + 1,20 km + 0,35 km + 0,80 km + 0,30 km + 0,35 km = 7,95 km
- c. Rute ketiga ; **R → U → C → B → D → E → H → 1** dengan bobot jarak : 0,85 km + 1,60 km + 1,30 km + 1,00 km + 0,24 km + 0, 50 km + 0,27 km = 5,76 km
- d. Rute keempat ; **R → C1 → M → K → L → H → 1** dengan bobot jarak : 3,00 km + 2,30 km + 1,72 km + 1,06 km + 0,38 km + 0,27 km = 8,73 km.

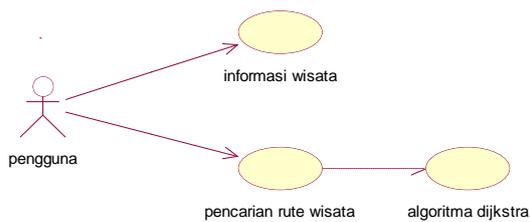
2. Dari keempat jalur yang dapat dilalui dari Terminal Type A Kertawangunan ke kolam renang sanggariang maka didapatkan jalur terpendek yaitu jalur pertama yaitu **R → U → C → J → K → L → H → 1** dengan keterangan rute sebagai berikut :

*Terminal Kertawangunan, ke Pasar Ancaran, ke Simpang Ciporang, ke Simpang Juanda, ke Lampu Merah Cijoho, ke Pemda Kuningan, ke Jln Aruji, lalu terakhir ke lokasi objek wisata Sanggariang. Dengan total jarak 5,72 Km.*

### 3.3.Diagram Use Case



**Gambar 7.** Use case system admin

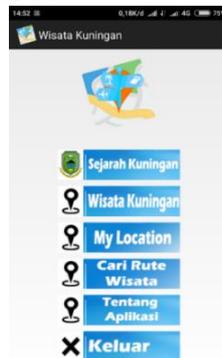


**Gambar 8.** Use case system pengguna

### 3.4 Interface Aplikasi

#### 3.4.1 Tampilan Beranda Aplikasi

Pada halaman penggunaini menampilkan *button* yang menuju halaman-halaman tertentu seperti menuju halaman sejarah, wisata, pencarian rute dan pencarian *my location*, berikut adalah tampilannya pada gambar 9.



**Gambar 9.** Tampilan menu beranda aplikasi

Terdapat lima *button* pada beranda aplikasi ini yaitu :

1. *Button* sejarah yaitu *button* yang menuju menu sejarah kuningan.
2. *Button* wisata kuningan yaitu *button* yang menuju menu daftar lokasi objek wisata yang ada di kabupaten kuningan.
3. *Button my location* yaitu *button* yang menuju menu dimana wisatawan dapat menempatkan lokasi dimana dia berada.
4. *Button* cari rute yaitu *button* yang menuju menu dimana digunakan untuk pencarian rute terdekat menuju lokasi objek wisata.
5. *Button* tentang yaitu *button* yang menuju menu dimana pengguna mengetahui tentang aplikasi.

#### 3.4.2 Tampilan Menu Kategori Wisata

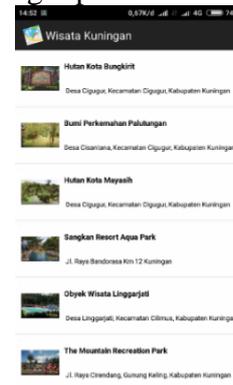
Tampilan menu kategori wisata merupakan halaman yang menampilkan beberapa kategori wisata yang ada di Kuningan.



**Gambar 10.** Tampilan menu kategori wisata

#### 3.4.3 Tampilan Menu Detail Kategori Wisata

Tampilan detail kategori wisata merupakan halaman yang menampilkan list wisata sesuai dengan kategori yang dipilih.



**Gambar 11.** Tampilan detail kategori wisata

#### 3.4.4 Tampilan Menu Detail Objek Wisata

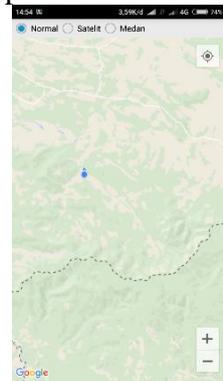
Tampilan detail objek wisata merupakan halaman yang menampilkan secara lengkap informasi dan deskripsi objek wisata.



**Gambar 12.** Tampilan detail objek wisata

### 3.4.5 Tampilan Menu *My Location*

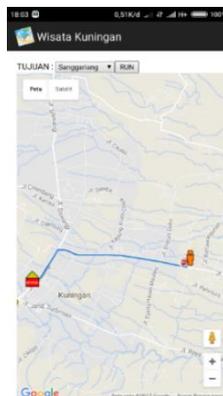
Tampilan pencarian *My Location* merupakan halaman yang menampilkan peta dan menampilkan lokasi si pengguna aplikasi.



**Gambar 13.** Tampilan pencarian *my location*

### 3.4.6 Tampilan Hasil Pencarian Rute Wisata

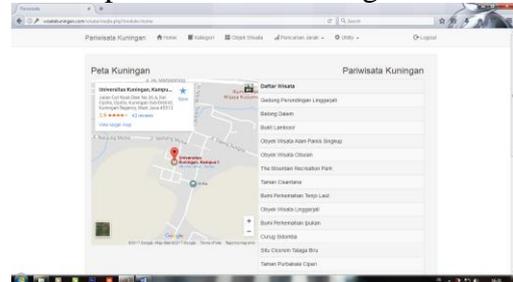
Tampilan hasil pencarian rute merupakan halaman yang menampilkan hasil pencarian rute yang diproses oleh algoritma.



**Gambar 14.** Tampilan hasil pencarian rute

### 3.4.7 Tampilan Halaman Admin

Tampilan home merupakan halaman utama ketika masuk ke halaman admin. Pada tampilan home terdapat detail apa yang akan ditampilkan di dalam aplikasi wisata kuningan.



**Gambar 15.** Tampilan halaman admin

## 4. KESIMPULAN

Berdasarkan hasil implementasi dan pengujian, maka kesimpulan dari topik ***“Implementasi Algoritma Dijkstra Pada Aplikasi Wisata Kuningan Berbasis Android (Studi Kasus: Dinas Pemuda Olahraga dan Pariwisata Kabupaten Kuningan)”*** yaitu sebagai berikut:

1. Aplikasi wisata kuningan diharapkan dapat membantu wisatawan yang ingin berwisata ke Kabupaten Kuningan, baik itu wisatawan lokal maupun wisatawan luar daerah.
2. Dengan adanya aplikasi ini, wisatawan dapat mencari informasi seputar wisata yang ada di Kabupaten Kuningan, karena aplikasi ini dapat memberikan informasi wisata yang ada di Kabupaten Kuningan saja.
3. Implementasi algoritma dijkstra dapat menemukan rute terpendek untuk mencari lokasi objek wisata.
4. Dengan adanya aplikasi ini informasi mengenai lokasi objek wisata di Kabupaten Kuningan bisa terpusat dan dapat mempermudah wisatawan.
5. Kelebihan aplikasi yang dibuat dari aplikasi lainnya yaitu aplikasi ini bukan hanya mencari rute tetapi juga ada deskripsi wisata dan daftar lokasi objek wisata yang ada di Kabupaten Kuningan.

## 5. SARAN

1. Penulis mengharapkan kedepannya apabila ada pengembangan ditambahkan sistem yang dapat mencakup data wisata yang ada di Wilayah 3 Cirebon.
2. Penulis mengharapkan pengembang kedepannya dapat menambahkan informasi lain seperti, informasi hotel dan fasilitas umum lainnya.

## DAFTAR PUSTAKA

- Santoso, Judhi S. (1993). Catatan Kuliah Teori Graph dan Aplikasinya. Teknik Informatika, ITB.
- Nazaruddin Safaat H. 2011. Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Informatika. Bandung.
- Jugiyanto, HM. (2015). Analisis dan Desain. Yogyakarta : Andi Offset
- Nazruddin Safaat H.(2012). Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android
- M. Aldous, Joan dan J. Wilson, Robin. 2000. Graphs and Application an Introductory Approach. Britania Raya: Universitas Terbuka.
- Pudjo Widodo, Prabowo.(2011).Menggunakan UML.Bandung: Informatika
- Rosa A.S dan M.Shalahuddin. (2010). Modul Pembelajaran Algoritma dan Pemrograman. Bandung : Modula.
- Utami, Ema.(2008).RDBMS menggunakan MySQL Server 2000. Yogyakarta : Graha Ilmu
- Hakim, Lukmanul.(2008). RaRahasia Inti Master PHP dan MySQLi(improved). Yogyakarta : Lokomedia
- Nugroho, Adi. 2005. Rational Rose untuk Pemodelan Berorientasi Objek. Bandung: Informatika
- Irawan, Koko. 2010. Potensi Objek Wisata Air Terjun Serdang Sebagai Daya Tarik Wisata Di Kota Labuan Batu Utara. Kertas Karya. Program Pendidikan Non Gelar Pariwisata. Universitas Sumatera.
- Sinaga, Supriono. 2010. Potensi dan Pengembangan Objek Wisata Di Kabupaten Tapanuli Tengah. Kertas Karya. Program DIII Pariwisata. Universitas Sumatera Utara.
- Nugroho, Adi (2009). Algoritma da Struktur Data dengan C#. Yogyakarta : Andi
- Hariningsih, SP. 2005. Teknologi Informasi. Jogjakarta: Graha Ilmu
- Sutabri, Tata . 2012. Analisis Sistem Informasi. Andi. Yogyakarta
- Yeni Kustiyahningsih, Devie Rosa Anamisa, 2011.Pemograman Basis Data Berbasis Web Menggunakan PHP & MySQL.Graha Ilmu : Yogyakarta.
- Cato Chandra,S.Kom dan Ir. Teddy Marcus Zakaria,M.T.2008.Be Smart Be Profesional With Microsoft Office 2007.Informatika.Bandung
- Pressman, R.S. (2010), Software Engineering : a practitioner's approach, McGraw-Hill, New York, 68.
- Rosa A.S dan Shalahuddin. (2011). Rekayasa Perangkat Lunak. Bandung : Informatika.
- Sinaga B.L, "Pemrograman Berorientasi Objek dengan Java", Gava Media, Yogyakarta, 2004.
- Beizer, B. (1990). Software Testing Techniques. Boston, International Thompson Computer Press
- Purnama, Rangsang. 2007. Pemrograman GUI Menggunakan Java. Jakarta : Prestasi Pustaka
- Buyens, Jim. 2001. Web Database Development. Elex Media Komputindo. Jakarta
- athiassen, L, Munk-Madsen, Andreas, N, Peter Axel, Stage, J. (2000). *Object Oriented Analysis & Design First Edition*. Aalborg, Denmark: Marko Publishing ApS.