

Penentuan Rute Evakuasi Bencana Kebakaran Menggunakan Algoritma Dijkstra berbasis *Web Framework* Vue.js

I Gede Surya Rahayuda¹, Ni Putu Linda Santiari²

Institut Teknologi dan Bisnis STIKOM Bali

e-mail: ¹surya_rahayuda@stikom-bali.ac.id, ²linda_santiari@stikom-bali.ac.id

Diajukan: 25 November 2019; Direvisi: 23 Desember 2019; Diterima: 27 April 2020

Abstrak

Penentuan rute terdekat diperlukan pada setiap perjalanan. Pada penelitian sebelumnya penulis telah menggunakan beberapa metode pencarian jalur yang diterapkan untuk menentukan jalur terdekat pengiriman produk. Pada penelitian saat ini penulis menerapkan metode pencarian jalur untuk penentuan rute terdekat perjalanan tim pemadam kebakaran. Diharapkan dengan digunakannya metode tersebut dapat mempercepat penentuan rute terdekat menuju lokasi kebakaran. Metode pencarian jalur yang digunakan adalah metode Dijkstra, metode tersebut diimplementasikan berbasis *web framework* vue.js. Berdasarkan penelitian yang dilakukan pada beberapa data tes, metode Dijkstra berhasil menentukan rute evakuasi terbaik. Penulis memilih menggunakan *framework* vue.js sebagai *front-end web* karena vue.js memiliki kelebihan dari desain tampilan yang baik dan reaktif. Berdasarkan pengujian sistem menggunakan *black-box testing* terhadap berbagai proses yang ada didapatkan hasil yang baik dan sesuai keinginan.

Kata kunci: Sistem Informasi Evakuasi Bencana, Dijkstra, Vue.js, Vuetify.js.

Abstract

Determination of the closest route is required on each trip. In previous studies the author has used several path search methods that are applied to determine the closest path of product delivery. In the current study the authors applied the path search method to determine the closest route to the fire fighter team. It is hoped that the use of these methods can accelerate the determination of the nearest route to the fire location. The path search method used is the Dijkstra method, the method is implemented based on the *web framework* vue.js. Based on research conducted on several test data, Dijkstra's method succeeded in determining the best evacuation route. The author chose to use the vue.js framework as a *web front-end* because vue.js has the advantages of good and reactive display design. Based on system testing using *black-box testing* on various existing processes obtained good results and as desired.

Keywords: Emergency Response System, Dijkstra, Vue.js, Vuetify.js.

1. Pendahuluan

Penentuan jalur terbaik atau sering disebut dengan *shortest pathfinding* merupakan suatu metode yang memiliki banyak manfaat dan dapat diterapkan pada berbagai kasus dan bidang ilmu. Pada penelitian sebelumnya penulis telah mengembangkan suatu program *desktop* menggunakan bahasa pemrograman *Visual Basic*, di mana program tersebut telah berhasil menentukan rute terbaik pada kasus pengiriman produk.

Pada tahun pertama program dikembangkan menggunakan dua metode, yaitu metode *Iterative Deepening Search* dan metode Held-Karp. Dari hasil penelitian yang dilakukan diketahui bahwa metode Held-Karp mendapatkan hasil yang lebih baik dibandingkan dengan metode *Iterative Deepening Search* [1]. Dalam beberapa pengiriman produk terdapat hasil yang sama dari kedua metode. Namun secara keseluruhan metode Held-Karp mendapatkan hasil yang lebih baik, selisih perbedaan antara kedua metode adalah sebesar 19,66% [2][3].

Pada tahun kedua, penulis menambahkan dua metode yaitu, *Bidirectional Search* dan *Depth Limited Search*. Metode tersebut berhasil diterapkan pada program dan digunakan untuk menentukan rute terpendek pada beberapa data tes. Dari perbandingan dengan metode lainnya, metode *Bidirectional Search* dan *Depth Limited Search* mendapatkan beberapa hasil yang lebih baik dari metode lainnya [4]. Menurut penulis metode *Bidirectional Search* akan lebih baik jika diterapkan pada kasus yang tidak memerlukan

perjalanan kembali ke titik awal dan akan mendapatkan hasil optimal jika digunakan pada kasus pengiriman yang memiliki data titik dengan jarak berdekatan di level awal dan akhir grap matrik, dan memiliki titik yang berjauhan di level tengah dari grap matrik. Metode *Depth Limited Search* baik diterapkan pada beberapa pengiriman produk yang memiliki poin atau tempat pengiriman yang banyak dan perlu dibatasi pada kegiatan pengiriman yang berbeda. Dari hasil yang didapatkan metode *Depth Limited Search* memiliki kesamaan dengan metode *Iterative Deepening Search*, hal ini disebabkan karena kedua metode tersebut merupakan pengembangan dari metode *Depth First Search*. Berdasarkan perbandingan hasil yang didapatkan, jika semua metode tersebut dibandingkan maka metode *Held-Karp* mendapatkan hasil yang terbaik [5].

Pada penelitian saat ini, atau penelitian tahun ketiga, penulis akan menggunakan metode Dijkstra, berdasarkan beberapa uji coba yang pernah dilakukan, metode Dijkstra memiliki kemungkinan untuk mendapatkan rute hasil yang lebih baik, dibandingkan dengan metode Held-Karp. Berbeda dengan penelitian sebelumnya, di mana metode *shortest pathfinding* diterapkan pada kasus pengiriman produk, pada penelitian tahun ini, penulis akan mencoba menerapkan metode Dijkstra pada suatu kasus penelitian mengenai penentuan rute terbaik tim pemadam kebakaran saat melakukan evakuasi korban pada bencana kebakaran. Sesuai dengan perencanaan pada penelitian sebelumnya [6][7], pada tahun ini penulis akan mengembangkan kedua metode tersebut menggunakan *web framework*. Dari beberapa *web framework* yang ada, penulis memilih menggunakan *web framework* Vue.js, karena *framework* Vue.js, memiliki kelebihan dari segi tampilan desain, Vue merupakan *frontend framework* yang menduduki peringkat atas dari beberapa situs *top rank framework web* saat ini. Vue memiliki kelebihan pada proses yang *reactive*, di mana hasil dari proses dilakukan secara langsung, seperti ketika berpindah halaman, *browser* tidak lagi *me-reload* seluruh elemen halaman, dengan begitu akan didapatkan *response* yang lebih cepat, *browser* juga tidak perlu mengunduh ulang elemen-elemen yang sudah ada, dengan begitu pengguna dapat lebih menghemat *bandwith* [8]. Fasilitas *two way data binding* dapat memberikan hasil langsung saat teks tersebut diubah, hal ini memungkinkan pengguna dapat memberikan pesan kepada pengguna lainnya tanpa jeda *reload* atau *save*. Vue juga menyediakan fasilitas *drag and drop* yang memungkinkan sebuah kelompok *post* bisa dipindahkan ke tempat lainnya secara *real time*. Ada lebih banyak lagi kelebihan dari Vue.js, dengan beberapa kelebihan tersebut, penulis akan menggunakan *framework* Vue.js pada penelitian tahun ini [9][10]. Penulis berharap nantinya akan dihasilkan suatu implementasi program yang baik, dan dapat memberikan beberapa pengetahuan baru bagi penulis, pembaca dan ilmu pengetahuan.

2. Metode Penelitian

Penulis menggunakan metode penelitian SDLC (*System Development Life Cycle*), SDLC merupakan siklus pengembangan perangkat lunak yang terdiri dari beberapa tahapan. Terdapat 4 metodologi dalam pengembangan perangkat lunak SDLC, yaitu: *Waterfall*, *Prototype*, *RAD (Rapid Application Development)*, dan *Agile Software Development*. Pada penelitian ini penulis menggunakan metode SDLC *Waterfall*. *Waterfall* atau *Classic Life Cycle* merupakan metode yang banyak digunakan pada *Software Engineering*, metode ini melakukan pendekatan secara sistematis dan terurut dari level kebutuhan sistem lalu menuju ke tahap analisis, desain, implementasi, dan pengujian sistem. Disebut *Waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Adapun tahapan-tahapan pada penelitian ini adalah:

a. Pengumpulan Data

Tahapan ini dilakukan untuk memperoleh data yang diperlukan dalam pengembangan sistem. Pengumpulan data dilakukan dengan mengambil data secara *online* dari situs Pemadam Kebakaran pada sebuah kota di negara Amerika Serikat, dengan nama Napperville. Data tersebut berupa data bencana kebakaran dari awal tahun 2019 sampai saat ini. Selain mengakses data secara *online*, penulis juga mengumpulkan data menggunakan studi literatur seperti buku, jurnal, prosiding, internet, dan beberapa sumber lainnya yang berkaitan dengan penelitian yang dilakukan.

b. Analisis

Tahapan analisis dilakukan untuk menganalisis permasalahan dan menentukan kebutuhan yang diperlukan dalam pembuatan sistem. Hasil analisis tersebut kemudian dijadikan dasar dalam membuat perancangan desain sistem.

c. Desain

Tahapan desain dilakukan untuk mengetahui alur data dan proses yang terjadi sistem. Perancangan desain sistem dilakukan menggunakan *Data Flow Diagram (DFD)*.

d. Implementasi Sistem

Tahapan implementasi dilakukan untuk menerjemahkan desain yang telah dibuat menggunakan bahasa pemrograman agar dapat dikembangkan menjadi sebuah sistem atau perangkat lunak. Sistem akan dikembangkan berbasis web menggunakan *framework* Vue.js. Untuk pencarian rute terdekat digunakan metode Dijkstra.

e. Pengujian Sistem

Tahapan pengujian sistem dilakukan untuk mengetahui apakah sistem yang dibuat telah sesuai dengan tujuan yang direncanakan. Pengujian sistem dilakukan menggunakan metode *black-box testing*.

Berikut merupakan gambar ilustrasi dari metode penelitian yang dilakukan:



Gambar 1. Metode penelitian.

3. Hasil dan Pembahasan

3.1. Pengumpulan Data

Berikut adalah beberapa data yang akan digunakan pada penelitian. Data tersebut merupakan *open data set* yang dibagikan oleh pemerintah Amerika Serikat, lebih khususnya pada sebuah kota dengan nama Napperville [13]. Berdasarkan sumbernya jenis data yang digunakan adalah data sekunder, karena data yang dikumpulkan diperoleh dari *open data set online*: <https://data.naperville.il.us>. Data tersebut merupakan data resmi yang dibagikan oleh departemen pemadam kebakaran pada sebuah kota di Amerika Serikat, yaitu Napperville *Fire Departemen* [14][12]. Sampai saat ini telah tercatat sebanyak 149.864 kejadian bencana kebakaran yang terjadi pada kota tersebut dari tahun 2010 – 2019. Penulis menggunakan sebanyak 80 data set terbaru dari keseluruhan data tersebut, berikut adalah beberapa data set yang akan digunakan pada penelitian, terlihat pada Tabel 1.

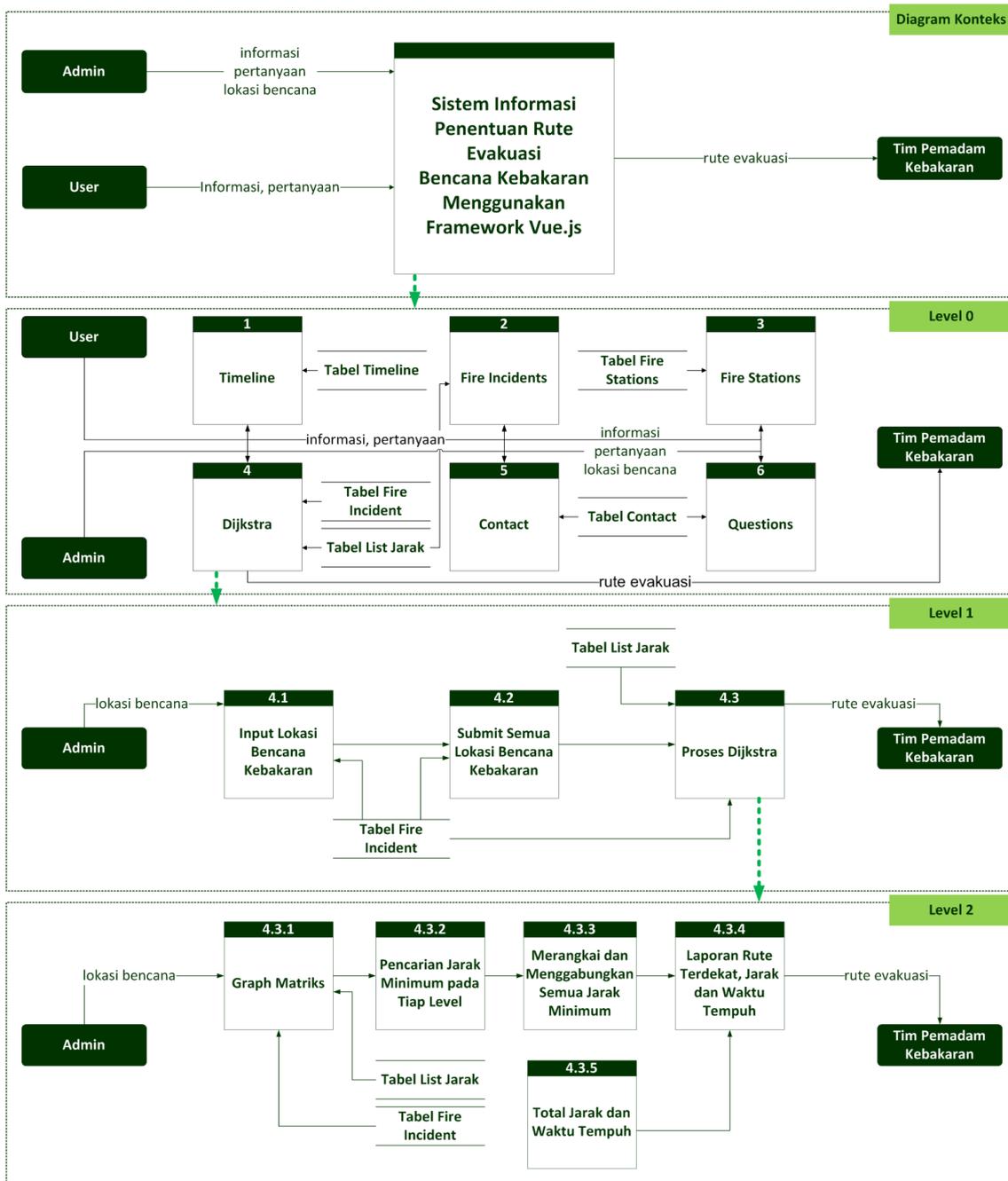
Tabel 1. Data bencana kebakaran dan koordinat pada *map*.

No	Tanggal	Alamat	Kota	Koordinat
1	22-Apr-2019	Normandy	Naperville	(41.711936, -88.222768)
2	22-Apr-2019	Audubon	Naperville	(41.811179, -88.200837)
3	22-Apr-2019	Rosecroft	Naperville	(41.711787, -88.210966)
4	22-Apr-2019	Naperville Wheaton	Naperville	(41.799269, -88.122684)
5	22-Apr-2019	Bradford	Naperville	(41.785456, -88.206206)
6	22-Apr-2019	Royal St George	Naperville	(41.781942, -88.168579)
7	22-Apr-2019	Water	Naperville	(41.770828, -88.150572)
8	22-Apr-2019	Hidden Spring	Naperville	(41.777714, -88.170752)
9	22-Apr-2019	Bradford	Naperville	(41.785456, -88.206206)
10	22-Apr-2019	Oswego	Naperville	(41.758973, -88.18478)
11	21-Apr-2019	Ogden	Naperville	(41.783567, -88.173161)
12	21-Apr-2019	Flat Rock	Naperville	(41.685055, -88.193387)
13	21-Apr-2019	Naper	Naperville	(41.803009, -88.119643)
14	21-Apr-2019	Timber Creek	Naperville	(41.704287, -88.155835)
15	21-Apr-2019	Grommon	Naperville	(41.699184, -88.178725)
16	21-Apr-2019	Fulham	Naperville	(41.727057, -88.203123)
17	21-Apr-2019	Diehl	Naperville	(41.798941, -88.22491)
18	21-Apr-2019	Iroquois	Naperville	(41.797784, -88.127581)
19	21-Apr-2019	Main	Naperville	(41.774669, -88.149323)
20	21-Apr-2019	Ogden	Naperville	(41.788615, -88.132897)
50
80	15-Apr-2019	Crystal	Naperville	(41.798796, -88.191144)

3.2. Analisis dan Desain Sistem

Untuk menjelaskan alur dan konsep dari penelitian, penulisan menggunakan *Data Flow Diagram* (DFD) untuk memodelkan alur penelitian yang dilakukan. DFD digambarkan mulai dari diagram konteks sampai dengan level 2. Pada DFD level 1 dan selanjutnya, penulis membatasi pengembangan DFD hanya

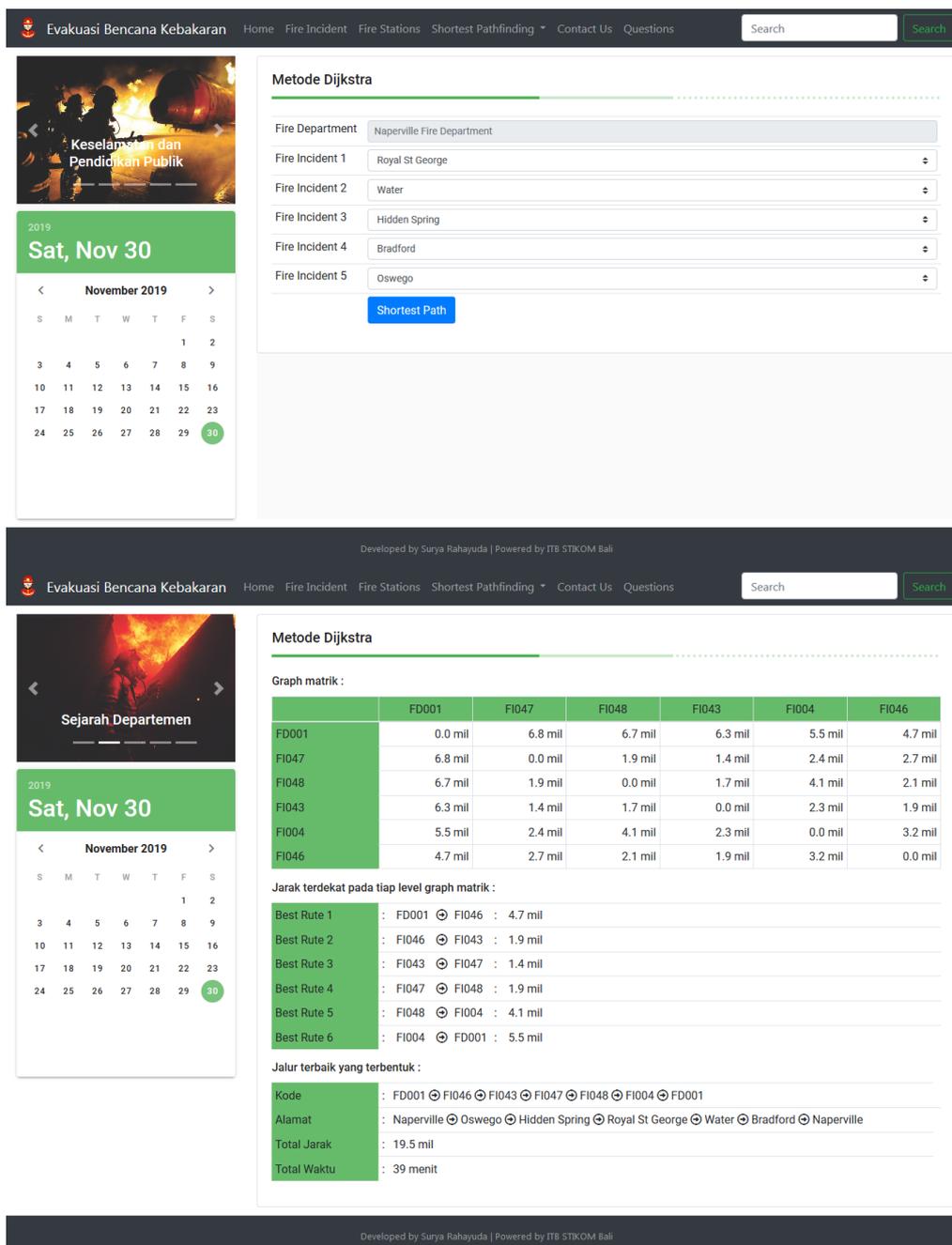
pada proses Dijkstra untuk mempersingkat penggambaran model DFD dan untuk memfokuskan pada inti dari penelitian yang dilakukan. DFD dari Sistem Informasi dapat dilihat pada gambar.



Gambar 2. Data flow diagram.

3.3. Implementasi

Metode Dijkstra telah berhasil dikembangkan berbasis web *framework* Vue.js. *Website* tersebut telah diunggah secara *online* pada salah satu *web hosting* dan dapat diakses melalui url: <http://surya-rahayuda.phpnet.us/rute-evakuasi>. Selain metode Dijkstra juga ditambahkan beberapa halaman lain sebagai pelengkap dari sistem informasi evakuasi bencana kebakaran. Beberapa halaman tersebut adalah: Home, Fire Incident, Fire Stations, Contact Us, dan Questions. Halaman Dijkstra dibangun untuk menerapkan metode Dijkstra. Pada halaman ini *user* atau *admin* dapat memilih beberapa lokasi kejadian kebakaran secara acak, dan kemudian menekan tombol Shortest Path. Sistem akan mengarahkan semua data yang di-*input*-kan menuju halaman *matrik.php* dan *dijkstra.php* di mana pada halaman tersebut data akan diproses menggunakan *graph* matrik dan metode *Dijkstra*. Akan terlihat *graph* matrik yang dihasilkan, jarak terdekat pada tiap level matrik dan rangkaian gabungan jalur terdekat yang terbentuk.



Gambar 3. Halaman Metode Dijkstra.

Metode Dijkstra mengawali pencarian dari level pertama grafik matrik, seperti terlihat pada grafik matrik level pertama di bawah ini:

1	FD001	FI047	FI048	FI043	FI004	FI046
FD001	0.0	6.8	6.7	6.3	5.5	4.7

dari keseluruhan data yang terdapat pada grafik level pertama, ditemukan bahwa rute terdekat adalah FD001 menuju FI046 dengan jarak 4.7 mil. Kemudian dilanjutkan dengan pencarian pada grafik matrik pada level FI046:

2	FD001	FI047	FI048	FI043	FI004	FI046
FD046	4.7	2.7	2.1	1.9	3.2	0.0

pada level grafik tersebut, ditemukan FI046 menuju FI043 memiliki rute terdekat sebesar 1.9 mil, pencarian dilanjutkan pada grafik level FI047:

3	FD001	FI047	FI048	FI043	FI004	FI046
FD043	6.3	1.4	1.7	0.0	2.3	1.9

pada level grafik di atas, ditemukan rute terdekat adalah FI043 menuju FI047 sebesar 1.4 mil, pencarian dilanjutkan pada grafik level FI047:

4	FD001	FI047	FI048	FI043	FI004	FI046
FD047	6.0	0.0	1.9	1.4	2.4	2.7

ditemukan rute terdekat adalah FI047 menuju FI048 sebesar 1.9 mil, pencarian dilanjutkan pada grafik level FI048:

5	FD001	FI047	FI048	FI043	FI004	FI046
FD048	6.7	1.9	0.0	1.7	4.1	2.1

rute terdekat ditemukan pada FI048 menuju FI004 sebesar 4.1 mil, pencarian dilanjutkan pada grafik level FI004:

6	FD001	FI047	FI048	FI043	FI004	FI046
FD004	5.5	2.4	4.1	2.3	0.0	3.2

grafik telah mencapai level akhir, pada level tersebut rute yang dipilih adalah rute kembali ke poin awal, yaitu FD004 menuju FD001, sebesar 5.5 mil. Semua rute tersebut kemudian digabungkan menjadi rangkaian rute perjalanan: FD001 → FI046 → FI043 → FI047 → FI048 → FI0204 → FD001, dengan total jarak tempuh sejauh 19.5 mil dan waktu tempuh selama 39 menit [14][13].

Semua data tes yang telah dikumpulkan dan disimpan kemudian diuji coba pada program berbasis web. Dari uji coba yang dilakukan dapat dihasilkan rute evakuasi pada tiap kasus bencana kebakaran, seperti terlihat pada Tabel 2.

Tabel 2. Hasil uji coba pada beberapa data tes.

No	Rute Hasil	Jarak	Waktu
1	Kasus 1, Kejadian Kebakaran pada : 22 April 2019		
	Naperville F.D. → Oswego	4.7 mil	9.4 menit
	Oswego → Hidden Spring	1.9 mil	3.8 menit
	Hidden Spring → Royal St George	1.4 mil	2.8 menit
	Royal St George → Water	1.9 mil	3.8 menit
	Water → Bradford	4.1 mil	8.2 menit
	Bradford → Naperville F.D.	5.5 mil	11 menit
	FD001 → FI046 → FI043 → FI047 → FI048 → FI0204 → FD001	19.5 mil	39 menit
2	Kasus 2, Kejadian Kebakaran pada : 22 April 2019		
	Naperville F.D. → Rosecroft	1 mil	2 menit
	Rosecroft → Normandy	1.1 mil	2.2 menit
	Normandy → Bradford	6.5 mil	13 menit
	Bradford → Audubon	2.3 mil	4.6 menit
	Audubon → Naperville Wheaton	6.1 mil	12.2 menit

No	Rute Hasil	Jarak	Waktu
	Naperville Wheaton → Naperville F.D.	9.6 mil	19.2 menit
	FD001 → FI031 → FI025 → FI004 → FI001 → FI024 → FD001	26.6 mil	53.2 menit
3	Kasus 3, Kejadian Kebakaran pada : 21 April 2019		
	Naperville F.D → Fulham	1.6 mil	3.2 menit
	Fulham → Main	6.3 mil	12.6 menit
	Main → Ogden	1.7 mil	3.4 menit
	Ogden → Iroquois	1.0 mil	2.0 menit
	Iroquois → Diehl	5.7 mil	11.4 menit
	Diehl → Naperville F.D	6.8 mil	13.6 menit
	FD001 → FI042 → FI045 → FI027 → FI044 → FI013 → FD001	23.1 mil	46.2 menit
4	Kasus 4, Kejadian Kebakaran pada : 21 April 2019		
	Naperville F. D. → Grommon	2.5 mil	5 menit
	Grommon → Flat Rock	1.6 mil	3.2 menit
	Flat Rock → Ogden	7.7 mil	15.4 menit
	Ogden → Cantera	4 mil	8 menit
	Cantera → Naper	3.7 mil	7.4 menit
	Naper → Naperville F. D.	11.8 mil	23.6 menit
	FD001 → FI019 → FI016 → FI027 → FI008 → FI023 → FD001	31.3 mil	62.6 menit
5	Kasus 5, Kejadian Kebakaran pada : 20 April 2019		
	Naperville F. D. → Willow Ridge	1.9 mil	3.8 menit
	Willow Ridge → Buttonwood	4 mil	8 menit
	Buttonwood → Candlenut	1.4 mil	2.8 menit
	Candlenut → Westminster	2.4 mil	4.8 menit
	Westminster → Diehl	1.2 mil	2.4 menit
	Diehl → Naperville F. D.	6.8 mil	13.6 menit
	FD001 → FI038 → FI006 → FI007 → FI036 → FI013 → FD001	17.7 mil	35.4 menit
6	Kasus 6, Kejadian Kebakaran pada : 19 April 2019		
	Naperville F. D. → Stockton	1.7 mil	3.4 menit
	Stockton → White Eagle	1.1 mil	2.2 menit
	White Eagle → Redstart	5.7 mil	11.4 menit
	Redstart → Centre Point	5.8 mil	11.6 menit
	Centre Point → Country Lakes	4.5 mil	9 menit
	Country Lakes → Naperville F. D.	6.8 mil	13.6 menit
	FD001 → FI034 → FI037 → FI030 → FI010 → FI011 → FD001	25.6 mil	51.2 menit
7	Kasus 7, Kejadian Kebakaran pada : 18 April 2019		
	Naperville F. D. → Aurora	4.4 mil	8.8 menit
	Aurora → Ellsworth	3.3 mil	6.6 menit
	Ellsworth → Hobson Mill	2.7 mil	5.4 menit
	Hobson Mill → Russet	2.2 mil	4.4 menit
	Russet → Woodview	2.2 mil	4.4 menit
	Woodview → Naperville F. D.	5 mil	10 menit
	FD001 → FI002 → FI015 → FI022 → FI032 → FI039 → FD001	19.8 mil	39.6 menit
8	Kasus 8, Kejadian Kebakaran pada : 17 April 2019		
	Naperville F. D. → Dorval	4.4 mil	8.8 menit
	Dorval → Washington	1.5 mil	3 menit
	Washington → North	5 mil	10 menit
	WNorth → Patriots	3.7 mil	7.4 menit
	Patriots → Genesee	1.4 mil	2.8 menit
	Genesee → Naperville F. D.	6.4 mil	12.8 menit
	FD001 → FI014 → FI035 → FI026 → FI028 → FI018 → FD001	22.4 mil	44.8 menit
9	Kasus 9, Kejadian Kebakaran pada : 16 April 2019		
	Naperville F. D. → Gypsum	1 mil	2 menit
	Gypsum → Garnette	3.7 mil	7.4 menit
	Garnette → Raintree	4.5 mil	9 menit
	Raintree → Brookdale	3.8 mil	7.6 menit
	Brookdale → Naperville Wheaton	5.5 mil	11 menit
	Naperville Wheaton → Naperville F. D.	9.6 mil	19.2 menit
	FD001 → FI020 → FI017 → FI029 → FI005 → FI024 → FD001	28.1 mil	56.2 menit
10	Kasus 10, Kejadian Kebakaran pada : 15 April 2019		
	Naperville F. D. → Spartina	3.3 mil	6.6 menit
	Spartina → Harbor	7.8 mil	15.6 menit
	Harbor → Center	3.4 mil	6.8 menit
	Center → Crystal	3.9 mil	7.8 menit
	Crystal → Bond	1.1 mil	2.2 menit
	Bond → Naperville F. D.	6.1 mil	12.2 menit
	FD001 → FI033 → FI021 → FI009 → FI012 → FI003 → FD001	25.6 mil	51.2 menit

Tabel 3. Perbandingan hasil Algoritma Dijkstra dan Held-Karp.

	Dijkstra	Held-Karp
Kasus 1	19.5 mil	17.8 mil
Kasus 2	26.6 mil	26.2 mil
Kasus 3	23.1 mil	23.1 mil
Kasus 4	28 mil	31.3 mil
Kasus 5	17.7 mil	17.7 mil
Kasus 6	25.6 mil	24.8 mil
Kasus 7	19.8 mil	19.8 mil
Kasus 8	22.4 mil	20.9 mil
Kasus 9	28.1 mil	25.5 mil
Kasus 10	25.6 mil	25.6 mil

Selain metode Dijkstra, penulis juga melengkapi halaman web dengan beberapa halaman, dimana pada tiap halaman menampilkan kelebihan dari *framework* Vue.js. Halaman Home merupakan halaman pertama yang diakses oleh *user*, pada halaman Home terdapat *sidebar* yang berisikan *carousel* dari gambar yang ada. Pada *main content* terdapat *timeline* yang berisikan berita atau *posting-an* yang dibuat oleh *admin*. *Timeline* dibuat menggunakan *Vuetify.js*, saat kursor diarahkan pada salah satu *posting-an* maka secara otomatis akan menutup gambar dan memfokuskan pada tombol detail dan tulisan. Pada halaman ini juga terdapat menu *navigation* yang mengarahkan pada semua halaman. Halaman Fire Incident berisikan semua informasi mengenai lokasi kejadian kebakaran yang pernah terjadi pada kota Napperville. Pada tabel terdapat tanggal, alamat, dan besar kerusakan yang terjadi menggunakan *progress bar*. Penyebab terjadinya kebakaran ditampilkan menggunakan *tooltip* yang akan muncul jika kursor diarahkan pada *progress bar*. Halaman Fire Station merupakan halaman yang berisi semua data mengenai *fire station* yang ada di kota Naperville, *list fire station* dibuat menggunakan *card deck*. Halaman Contact Us merupakan halaman yang dibangun agar *user* yang mengakses *website* dapat meng-*input*-kan data berupa biodata dan pertanyaan mengenai bencana kebakaran, departemen dan masalah atau keluhan lainnya. Data pertanyaan yang telah disimpan pada halaman Contact Us dapat diakses melalui halaman Question. Jika tombol View All ditekan, maka secara otomatis semua pertanyaan akan ditampilkan satu persatu sampai batas *queue* maksimal kemudian hilang atau *dequeue* dan digantikan oleh pertanyaan lainnya atau *enqueue* dan begitu seterusnya, sehingga pertanyaan tersebut terlihat terus bergerak. Proses tersebut atau disebut dengan *realtime logging* dibangun menggunakan *material* desain *vuetify.js*. Selain menggunakan media komputer atau laptop, tentunya *website* juga akan diakses menggunakan *device* lainnya yang lebih kecil seperti *tab* dan *handphone*. *Website* yang dibangun sudah menerapkan teknologi *responsive*, sehingga jika diakses menggunakan halaman kecil *website* akan tetap menyesuaikan dengan tampilan layar.

3.4. Pengujian Sistem

Pada penelitian ini digunakan metode pengujian *black box testing*. Pengujian dilakukan dengan menguji bagian antarmuka dari sistem informasi, setiap bagian dari antarmuka tersebut diuji agar dapat ditentukan apakah sistem sudah berjalan sesuai dengan fungsi yang diharapkan [8]. Tujuan dari pengujian ini adalah untuk mengetahui kesalahan pada sistem yang dibuat. Berikut adalah hasil pengujian menggunakan metode *black box testing* yang ditampilkan dalam bentuk tabel:

Tabel 4. Hasil pengujian sistem.

No	Kelas Uji	Skenario Pengujian	Hasil Pengujian
1	Halaman Home	Admin atau user dapat mengakses halaman home, pada halaman home akan menampilkan beberapa konten berita yang ditampilkan dalam bentuk <i>timeline</i> , setiap kolom berita dapat ditekan agar selanjutnya dapat diakses halaman yang menampilkan detail berita	Sesuai
2	Halaman Fire Incident	Halaman <i>fire incident</i> dapat diakses oleh <i>admin</i> dan <i>user</i> , halaman tersebut akan menampilkan <i>list</i> semua lokasi bencana kebakaran dalam bentuk tabel, terdapat kolom berupa persentase kerusakan dalam bentuk <i>barline</i> , dimana apabila kursor diarahkan pada <i>barline</i> akan menampilkan detail kerusakan dalam bentuk <i>tooltips</i>	Sesuai
3	Halaman Fire Stations	Halaman <i>fire stations</i> dapat diakses oleh <i>user</i> dan <i>admin</i> , halaman tersebut akan menampilkan semua stasiun pemadam yang ada dalam bentuk <i>card column</i> yang disertai dengan gambar	Sesuai
4	Halaman Dijkstra	Halaman <i>Dijkstra</i> merupakan halaman utama dari sistem, dimana <i>admin</i> dapat mengakses dan dapat menginputkan tempat kejadian bencana kebakaran, jika ditekan tombol submit maka sistem akan memproses graph matriks dan metode <i>Dijkstra</i> untuk mendapatkan solusi rute terbaik	Sesuai

No	Kelas Uji	Skenario Pengujian	Hasil Pengujian
5	Halaman <i>Contact Us</i>	Halaman <i>contact us</i> dapat diakses oleh <i>user</i> , <i>user</i> dapat menginputkan identitas dan dapat menuliskan saran atau pertanyaan kemudian data tersebut akan disimpan pada basis data sistem	Sesuai
6	Halaman <i>Questions</i>	Halaman <i>questions</i> dapat diakses oleh <i>user</i> dan <i>admin</i> , pada halaman ini digunakan teknologi <i>realtime logging</i> dari <i>vueify.js</i> , dimana jika tombol <i>view questions</i> ditekan, maka semua pertanyaan yang ada pada <i>database</i> akan ditampilkan dalam bentuk animasi <i>slide vertical transitions</i> secara bergiliran dan berganti warna	Sesuai

4. Kesimpulan

Sistem informasi penentuan rute evakuasi bencana kebakaran telah berhasil dikembangkan menggunakan *framework vue.js*. Alur dan desain sistem berhasil diilustrasikan menggunakan *data flow diagram*, dimana *data flow diagram* dikembangkan hingga level 2. Metode *Dijkstra* berhasil diterapkan pada sistem dan berhasil diujicobakan pada beberapa data tes. Namun terdapat beberapa kegagalan pada perbandingan hasil dari ujicoba algoritma *Dijkstra* dan *Held-Karp* terhadap data yang sama. Dari 10 kasus kejadian kebakaran, terdapat sebanyak 4 kasus evakuasi yang memiliki hasil sama antara metode *Dijkstra* dan *Held-Karp* dan terdapat 6 kasus evakuasi, dimana metode *Held-Karp* memiliki hasil yang lebih baik dibandingkan dengan metode *Dijkstra*. Hal ini mungkin disebabkan karena metode *Dijkstra* diterapkan pada kasus yang mengunjungi semua titik dan memungkinkan perjalanan kembali ke titik awal. Keterbatasan jumlah data uji juga dapat mempengaruhi hasil. Pengujian sistem dilakukan menggunakan metode *black box testing*, berdasarkan hasil dari pengujian didapatkan hasil yang baik dan sesuai dengan keinginan.

Daftar Pustaka

- [1] I. G. S. Rahayuda and N. P. L. Santiari, "Penerapan Pemrograman Dinamis Pada Manajemen Pengiriman Produk Menggunakan Metode Held-Karp," in *Konferensi Nasional Sistem & Informatika 2017*, 2017, pp. 513–518.
- [2] I. G. S. Rahayuda and N. P. L. Santiari, "Implementasi dan Perbandingan Metode Iterative Deepening Search dan Held-Karp pada Manajemen Pengiriman Produk," *J. Sisfo Inspirasi Prof. Sist. Inf. Institut Teknol. Sepuluh Nop.*, vol. 07, no. 02, 2017.
- [3] I. G. S. Rahayuda and N. P. L. Santiari, "Basis Path Testing of Iterative Deepening Search and Held-Karp on Pathfinding Algorithm," *J. Ilm. Kursor*, vol. 9, no. 2, 2018.
- [4] I. G. S. Rahayuda, N. P. L. Santiari, and N. Y. Arso, "Penerapan Bidirectional Search Dan Held-Karp Pada Penentuan Rute Pengiriman Produk," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 5, pp. 549–558, 2018.
- [5] I. G. S. Rahayuda and N. P. L. Santiari, "Penerapan Algoritma Depth Limited Search pada Penentuan Rute Pengiriman Produk," in *Seminar Nasional Telekomunikasi dan Informatika (Selisik) 2018*, 2018, pp. 184–189.
- [6] I. G. S. Rahayuda and N. P. L. Santiari, "Fire Incident Emergency Response Plan using Hybrid Fuzzy Dijkstra," *2019 1st Int. Conf. Cybern. Intell. Syst.*, 2019.
- [7] I. G. S. Rahayuda and N. P. L. Santiari, "Penentuan Rute Evakuasi Bencana Kebakaran Menggunakan Metode Dijkstra," in *Seminar Nasional Pendidikan Teknik Informatika (SENAPATI) ke-10 Tahun 2019*, 2019.
- [8] N. L. P. S. Aditya Herdinata Putra, Dian Pramana, "Sistem Manajemen Arsip Menggunakan Framework Laravel dan Vue.Js (Studi Kasus : BPKAD Provinsi Bali)," *J. Sist. dan Inform.*, vol. 13, no. 2, 2019.
- [9] H. X. Junhui Song, Min Zhang, "Design and Implementation of a Vue.js-Based College Teaching System," *Int. J. Emerg. Technol. Learn.*, vol. 14, no. 13, 2019.
- [10] Y. Quan, "Design and Implementation of E-commerce Platform based on Vue.js and MySQL," in *Proceedings of the 3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019)*, 2019.
- [11] M. Krumova, "Open Data Benchmarking for Higher Education: Management and Technology Perspectives," *Smart Cities Reg. Dev. J.*, 2017.
- [12] A. W. Sejati, "Pemanfaatan Open Data untuk Mencari Jurnal dan Referensi Ilmiah," *Res. Gate*, 2019.
- [13] C. Kure, "Open Data in a Big Data World," *Sci. Int.*, 2015.
- [14] A. Z. Marijn Janssen, Yannis Charalabidis, "Benefits, Adoption Barriers and Myths of Open Data and Open Government," *J. Inf. Syst. Manag.*, vol. 29, 2012.