**SISTEM INFORMASI RUTE WISATA DI KOTA SURABAYA MENGGUNAKAN ALGORITMA DIJKSTRA DENGAN GRAF REDUKSI**

Alfatara Nurus Sa’adah

Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Surabaya

e-mail: [alfatarasaadah@mhs.unesa.ac.id](mailto:alfatarasaadah@mhs.unesa.ac.id)

Elly Matul Imah

Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Surabaya

e-mail: [ellymatul@mhs.unesa.ac.id](mailto:ellymatul@mhs.unesa.ac.id)

Abstrak

Salah satu manfaat dari perkembangan teknologi adalah penggunaan *Google Maps*, yang membuat tempat wisata di Kota Surabaya sudah banyak dieksplor oleh para wisatawan. Namun, kemacetan yang terjadi di Kota Surabaya menjadi masalah para wisatawan. Pencarian rute wisata terpendek untuk mengunjungi suatu tempat wisata adalah solusi yang sangat tepat. Masalah ini dapat diselesaikan dengan algoritma *Dijkstra* dengan graf reduksi. Dalam proses pencarian rute wisata terpendek, sebuah graf yang merepresentasikan 25 tempat wisata di Surabaya harus direduksi menggunakan aturan reduksi *fork closed rule* terlebih dahulu untuk mempercepat proses pencarian menggunakan algoritma Dijkstra. Dengan adanya sistem informasi rute wisata terpendek di kota Surabaya, pencarian rute wisata lebih cepat dan efisien dengan kompleksitas waktu sebesar dan dapat memudahkan masyarakat umum yang membutuhkan lintasan terpendek untuk menuju tempat – tempat wisata di Kota Surabaya.

Kata kunci : *Dijkstra*, Graf Reduksi, Sistem Informasi

Abstract

One of the benefits of technological development is the use of Google Maps, which makes the tourist attractions in the city of Surabaya has been much explored by tourists. However, the congestion that occurs in the city of Surabaya to be a problem for tourists. Searching for the shortest tourist route to visit a tourist attraction is a very appropriate solution. This problem can be solved by Dijkstra's algorithm with reduction graph. In the shortest tourist route search process, a graph which represents 25 sites in Surabaya should be reduced using the reduction rules fork closed rule in advance to speed up the search process using Dijkstra's algorithm. With the shortest tourist route information system in the city of Surabaya, tourist route search faster and more efficiently with time complexity of , and can facilitate the public who need the shortest path to get to places - attractions in Surabaya.

Keywords : Dijkstra, Graph Reduction, Information System

# **PENDAHULUAN**

Kota Surabaya merupakan ibu kota Provinsi Jawa Timur, Surabaya juga merupakan kota metropolitan terbesar di Provinsi Jawa Timur (Surabaya.go.id). Selain menjadi pusat penopang perekonomian wilayah Indonesia bagian tengah dan timur, Surabaya juga memiliki banyak destinasi wisata yang menarik. Dari wisata alam, wisata sejarah, wisata religi, wisata kuliner nmaupun wisata keluarga, ada di kota Surabaya (Wikipedia, 2016).

Salah satu manfaat dari perkembangan teknologi yang semakin pesat adalah tempat – tempat wisata di Surabaya sudah banyak dieksplor oleh para wisatawan terutama wisatawan luar daerah dan wisatawan asing. Namun, masih ada beberapa masalah saat wisatawan akan berwisata di Surabaya, kota metropolitan yang sering dilanda kemacetan hampir di setiap jamnya. Efisiensi waktu saat akan mengunjungi beberapa tempat wisata menjadi salah satu masalah yang saat ini sedang terjadi. Pencarian lintasan terpendek untuk mengunjungi suatu tempat wisata adalah solusi yang sangat tepat. Masalah pencarian lintasan terpendek dapat diselesaikan dengan konsep teori graf. Metode yang dapat digunakan untuk mencari lintasan terpendek salah satunya adalah algoritma *Dijkstra* dengan graf reduksi.

Algoritma *Dijkstra* adalah sebuah algoritma yang dipakai untuk mencari panjang lintasan terpendek dari sebuah titik a ke sebuah titik z di graf bobot G, dengan bobot setiap sisi G adalah bilangan positif (I Ketut Budayasa, 2007). Graph-bobot G yaitu graph yang setiap sisinya dikaitkan dengan bilangan real. Misal e adalah sisi pada graph-bobot G, maka w(e) adalah bobot sisi e yang berupa bilangan real. Bobot pada graf-bobot G merepresentasikan jarak antara titik a ke titik z. Namun, untuk menyelesaikan masalah efisiensi waktu saat akan mengunjungi beberapa tempat wisata, maka bobot pada graf-bobot G akan ditambahkan dengan variabel tingkat kepadatan.

Pencarian lintasan terpendek menggunakan algoritma *Dijkstra* dengan kompleksitas sebesar, dengan adalah banyak titik dan adalah banyak sisi. Dengan menggunakan graf reduksi sebagai metode tambahannya akan mempercepat pencarian Dijkstra lebih cepat karena dapat menemukan lintasan terpendek tanpa melalui lintasan – lintasan yang tidak perlu (Rafael Rodríguez-Puente dan Manuel S Lazo-Cortés, 2013).

**METODE PENELITIAN**

**Lintasan Terpendek**

Dalam graf berbobot, lintasan terpendek yakni, sebuah lintasan yang mempunyai panjang minimum antara dua titik yang diketahui (Richard Johnsonbaugh, 2002).



Gambar 1. Lintasan terpendek dari v1 ke v3 pada graf G

Lintasan terpendek dari titik v1 ke v3 pada graf G pada Gambar 1, yaitu (v1, v4, v5, v3), dengan panjang .

**Algoritma Dijkstra**

Algoritma *Dijkstra* adalah sebuah algoritma yang dipakai untuk mencari panjang lintasan terpendek dari sebuah titik a ke sebuah titik z di graf bobot G, dengan bobot setiap sisi G adalah bilangan positif (I Ketut Budayasa, 2007).

Algoritma dijkstra akan diurakan sebagai berikut:

Input : Sebuah graf G berbobot terhubung dengan vertex sampai .

Output : , panjang lintasan terpendek dari sampai .

1. : Inisialisasi V, dimana V adalah himpunan titik, V. Dimana adalah tempat–tempat wisata yang akan dikunjungi. Pilih sebagai titik awal rute wisata dan .
2. : Inisialisasi adalah panjang lintasan terpendek dari ke , . Untuk setiap , maka , dimana adalahpanjang lintasan terpendek dari ke , dengan panjang lintasan , dan .

T adalah himpunan titik pada graf G yang panjang lintasan terpendeknya dari belum ditentukan.

1. : Pilih , dengan minimum , sehingga .
2. : Untuk setiap disamping , maka .
3. : Jika , kembali ke langkah 3. Jika , maka didapat dengan panjang lintasan terpendek dari ke .

(Richard Johnsonbaugh, 2002)

**Graf Reduksi**

*Definisi 1*

Sebuah graf reduksi adalah sebuah tupel Gr = (), dimana:

* adalah himpunan titik.
* adalah himpunan sisi.
* : →(), adalah fungsi dimana untuk setiap mengembalikan biaya pergi dari ke melalui dengan mendekati dan mendekati . Fungsi f juga ditetapkan untuk kasus-kasus dimana dan/atau . Dalam kasus trivial f()=0.
* adalah himpunan penulisan ulang melalui (), dimana didefinisikan sebagai

Algoritma graf reduksiditunjukkan sebagai berikut :

Input : – Sebuah graf , dimana

adalah himpunan dari penulisan ulang.

– Sebuah partisi P pada V.

Output : Sebuah graf reduksi

Langkah 1 : Tentukan , dimana ],

Langkah 2 : – Diperoleh titik yang akan direduksi

* Diperoleh sisi dari titik yang direduksi .
* Didapat aturan penulisan ulang .

Langkah 3 : Buat graf reduksi .

(Rafael Rodríguez-Puente dan Manuel S Lazo-Cortés, 2013)

**Aturan Reduksi**

Proses reduksi yang menggunakan lima aturan, yaituterminal, sequential, adjacent, closed, and overlapped. Aturan pengurangan ini diterapkan dengan melewati semua node dari graf dan memeriksa jika aturan pengurangan tersebut dapat diterapkan. Ketika sebuah node dihapus dari graf, maka semua transisi yang melekat padanya secara otomatis dihapus.

*Closed Reduction Rule*

Aturan reduksi *closed* mengurangi beberapa transisi antara dua node dari jenis yang sama. Dalam kasus reduksi *closed choice* dan gabungan, seperti gambar 2 yang dilakukan adalah penjumlahan bukan perkalian. Jika panjang sebuah graf dipilih dari salah satu panjang subgraf dibawah transisi yang terkait dengan , tidak ada panjang subgraf dibawah transisi yang terkait dengan yang akan dipilih karena strukturnya *choice*. Oleh karena itu, untuk reduksi *closed choice* dan gabungan, jumlah total panjang subgraf dari ke menjadi (Wasim Sadiq dan Maria E Orlowska, 2000).



Gambar 2. Prinsip Perhitungan Untuk Aturan Reduksi *Closed*

**Algoritma *Dijkstra yang Dimodifikasi***

Algoritma Dijkstra yang dimodifikasi adalah sebuah algoritma dijkstra yang dimodifikasi dengan menggunakan graf reduksi.

Berikut adalah penjelasan dari algoritma dijkstra yang telah dimodifikasi:

Input : Sebuah graf berbobot terhubung dengan vertex sampai .

Output : L(z), panjang lintasan terpendek dari a sampai z.

Langkah 1 : Inisialisasi , dimana adalah himpunan titik, . Dimana adalah tempat–tempat wisata yang tersedia. Pilih sebagai titik awal rute wisata dan .

Langkah 2 : Inisialisasi adalah panjang lintasan terpendek dari ke , . Untuk setiap , maka , dimana adalah panjang lintasan terpendek dari ke , dengan panjang lintasan

dan jarak dari ke .

adalah himpunan titik pada graf yang panjang lintasan terpendeknya dari belum ditentukan.

Langkah 3 : Pilih , dengan minimum , sehingga .

Langkah 4 : Untuk setiap disamping , maka .

Langkah 5 : Jika , kembali ke langkah 3. Jika , maka didapat dengan panjang lintasan terpendek dari ke .

(Rafael Rodríguez-Puente dan Manuel S Lazo-Cortés, 2013)



Gambar 3. Alur Algoritma Dijkstra yang Dimodifikasi

**Tahapan Membangun Sistem Informasi**

Agar sebuah sistem informasi valid dan berjalan efisien sesuai kebutuhan user, ada beberapa tahapan yang harus dilakukan oleh *developer*:

1. Perencanaan
2. Analisa
3. Rancangan Desain

Tahapan ini merupakan fondasi sebuah sistem informasi. Ada 3 jenis desain yang dibuat di langkah ini, yaitu:

1. Desain Data Flow Diagram
2. Desain Entity Relationship Diagram
3. Desain User Interface
4. Pengembangan
5. Testing

**Tempat Wisata Kota Surabaya**

Tempat wisata di kora Surabaya ada 5 jenis, yaitu wisata religi, wisata alam, wisata sejarah, wisata kuliner, dan wisata keluarga. Berikut tempat – tempat wisata di kota Surabaya yang akan ditampilkan dalam Tabel 1.

Tabel 1. Data Tempat Wisata di Surabaya

|  |  |  |
| --- | --- | --- |
| **Jenis Wisata** | **Tempat Wisata** | **Node** |
| Wisata Alam | * Ekowisata Mangrove |  |
| * Pantai Ria Kenjeran |  |
| Wisata Religi | * Gereja Katolik Santa Maria Tak Bercela |  |
| * Makam Sunan Ampel |  |
| * Masjid Muhammad Cheng Ho |  |
| * Pura Agung Jagat Karana |  |
| Wisata Sejarah | * Monumen Kapal Selam |  |
| * Monumen Jalesveva Jayamahe |  |
| * Museum TNI - AL Loka Jala Crana |  |
| * Museum WR. Soepratman |  |
| * Monumen Tugu Pahlawan |  |
| Wisata Keluarga | * Ciputra Waterpark |  |
| * Kebun Bibit Wonorejo Surabaya |  |
| * Kebun Binatang Surabaya |  |
| * Patung Joko Dolog |  |
| * Taman Harmoni |  |
| * Taman Bungkul |  |
| * Suroboyo Carnival Park |  |
| Wisata Kuliner | * Pakuwon Food Festival |  |
| * Rawon Pak Pangat Wonokromo |  |
| * Sate Klopo Ondomohen Ny. Asih |  |
| * Rujak Cingur Ahmad Jaiz |  |
| * Zangrandi Ice Cream |  |
| * Lontong Balap Garuda Pak Gendut |  |
| * Semanggi Dempo Surabaya |  |

**HASIL DAN ANALISA**

**Entity Relationship Diagram (ERD)**

Entity Relationship Diagram dari Sistem Informasi Rute Wisata Terpendek Di Kota Surabaya akan dijelaskan pada Gambar 4.



Gambar 4. Entity Relationship Diagram dari Sistem Informasi Rute Perjalanan Wisata di Kota Surabaya

***Flowchart***

*Flowchart* analisis pencarian rute wisata terpendek di Kota Surabaya menggunakan algoritma Dijkstra dengan graf reduksi akan dijelaskan seperti Gambar 5.





Gambar 5. Flowchart Analisis Pencarian Rute Wisata Terpendek Di Kota Surabaya Menggunakan Algoritma Dijkstra Dengan Graf Reduksi

**Data Flow Diagram (DFD)**

1. Data Flow Diagram (DFD) Level 0

Data Flow Diagram (DFD) Level 0 dari Sistem Informasi Rute Wisata Terpendek Di Kota Surabaya akan dijelaskan pada Gambar 6.



Gambar 6. Data Flow Diagram Level 0 dari Sistem Informasi Rute Perjalanan Wisata di Kota Surabaya

1. Data Flow Diagram (DFD) Level 1

Data Flow Diagram (DFD) Level 1 dari Sistem Informasi Rute Wisata Terpendek Di Kota Surabaya akan dijelaskan pada Gambar 7.



Gambar 7. Data Flow Diagram Level 1 dari Sistem Informasi Rute Perjalanan Wisata di Kota Surabaya

**User Interface**

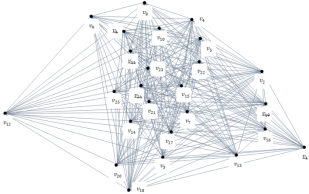
User Interface dari Sistem Informasi Rute Wisata Terpendek Di Kota Surabaya akan dijelaskan pada Gambar 8 dan Gambar 9.

Gambar 8. User Interface – memilih tempat wisata dan jam keberangkatan

Gambar 9. User Interface – menampilkan hasil pencarian rute wisata terpendek

**Data**

Data nama tempat – tempat wisata di Surabaya diperoleh dari Dinas Kebudayaan dan Pariwisata Kota Surabaya sebanyak 25 tempat wisata. Sedangkan data koordinat setiap tempat wisata, data jarak antar tempat wisata, dan data tingkat kepadatan rute antar tempat wisata diperoleh dari google maps. Titik koordinat digunakan untuk menentukan lokasi setiap tempat wisata dalam program, detail lihat Gambar 10.



Gambar 10. Graf berbobot G dengan 25 titik dan titik dengan node bergaris bawah adalah titik tujuan.

Berikut adalah keterangan dari titik tujuan wisata (titik dengan node bergaris bawah) :

: Ekowisata Mangrove

: Pura Agung Jagat Karana

: Monumen Tugu Pahlawan

: Pakuwon Food Festival

: Lontong Balap Garuda Pak Gendut

Berikut ini adalah langkah – langkah penyelesaian masalah pencarian rute wisata menggunakan algoritma dijkstra dengan graf reduksi:

* 1. Memilih tempat wisata yang dikunjungi

Ada 5 tempat wisata yang dipilih, yaitu Ekowisata Mangrove, Pura Agung Jagat Karana, Monumen Tugu Pahlawan, Pakuwon Food Festival, dan Lontong Balap Garuda Pak Gendut.

* 1. Penyelesaian manual menggunakan algoritma dijkstra dengan graf reduksi

Berikut ini adalah langkah-langkah pengolahan data menggunakan algoritma dijkstra dengan graf reduksi:

* + 1. Menentukan titik tereduksi ()

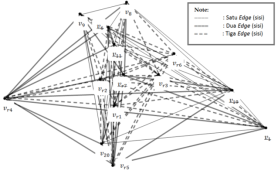
Untuk menentukan titik mana yang akan direduksi, maka akan dicari . Setelah didapat titik tereduksi (), maka akan ditulis ulang dengan rumus . Hasil dari proses iterasi untuk menentukan titik tereduksi lihat pada Tabel 2.

Tabel 2. Proses Iterasi untuk Menentukan Titik Tereduksi

|  |  |  |
| --- | --- | --- |
| Iterasi ke - | Titik Reduksi | Bobot |
| 1 | – | Null |
| 2 |  | 3 |
| 3 |  | 12 |
| 4 | – | Null |
| 5 |  | 13 |
| 6 | – | Null |
| 7 | – | Null |
| 8 | – | Null |
| 9 | – | Null |
| 10 |  | 14 |
| 11 |  | 17 |
| 12 |  | 16 |
| 13 | – | Null |
| 14 | 16 | 20 |
| 15 | – | Null |
| 16 | – | Null |
| 17 |  | 14 |
| 18 |  | 18 |
| 19 |  | 22 |
| 20 |  | 32 |
| 21 | – | Null |
| 22 | – | Null |
| 23 | – | Null |
| 24 | (Titik Tujuan) | 14 |
| 25 | (Titik Tujuan) | 12 |
| 26 | (Titik Tujuan) | 32 |
| 27 | (Titik Tujuan) | 29 |
| 28 | (Titik Tujuan) | 27 |
| 29 | (Titik Tujuan) | 32 |
| 30 | – | Null |
| 31 | (Titik Tujuan) | 24 |
| 32 | (Titik Tujuan) | 26 |
| 33 | – | Null |
| 34 | – | Null |

* + 1. Mereduksi graf

Berdasarkan algoritma graf reduksi, graf G (Gambar 10) direduksi sehingga menjadi graf Gr (Gambar 11).



Gambar 11. Graf Gr

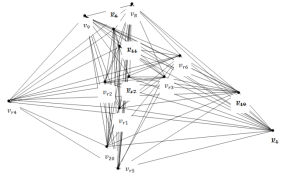
* + 1. Pencarian rute wisata menggunakan algoritma dijkstra dengan graf tereduksi

Permasalahan:

Seorang wisatawan ingin mengunjungi tempat wisata di Surabaya dari → → → → . Wisatawan tersebut ingin mencari lintasan terpendek agar dapat meminimumkan waktu tempuh untuk tiba di tujuan. Berikut ini adalah graf bobot dengan 25 titik dimana titik tersebut merupakan 25 tempat wisata dan setiap sisi diasumsikan jalan dua arah (simetris).

Penyelesaian:

Masalah ini akan diselesaikan menggunakan algoritma Dijkstra dengan graf reduksi. Dengan menggunakan aturan *closed reduction rule*, graf menjadi seperti Gambar 11.



Gambar 12. Graf Gr setelah diberi aturan closed reduction

Berikut adalah langkah penyelesaian masalah:

* + - 1. Mencari lintasan terpendek dari ke .

Dengan menggunakan algoritma Dijkstra, didapat lintasan terpendek dari ke , yaitu {} dengan panjang lintasan .

* + - 1. Mencari lintasan terpendek dari ke .

Dengan menggunakan algoritma Dijkstra, didapat lintasan terpendek dari ke , yaitu {} dengan panjang lintasan .

* + - 1. Mencari lintasan terpendek dari ke .

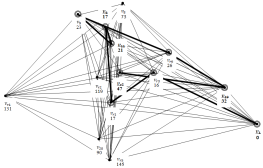
Dengan menggunakan algoritma Dijkstra, didapat lintasan terpendek dari ke , yaitu {} dengan panjang lintasan .

* + - 1. Mencari lintasan terpendek dari ke .

Dengan menggunakan algoritma Dijkstra, didapat lintasan terpendek dari ke , yaitu {} dengan panjang lintasan .

* + - 1. Mencari lintasan terpendek dari ke .

Dengan menggunakan algoritma Dijkstra, didapat lintasan terpendek → → → → , yaitu {} dengan panjang lintasan . Detail lihat Gambar 13.



Gambar 13. Hasil pencarian lintasan terpendek → → → → menggunakan algoritma Dijkstra dengan graf reduksi

* 1. Penyelesaian komputerisasi menggunakan algoritma dijkstra dengan graf reduksi

Berikut ini adalah langkah – langkah penyelesaian masalah pencarian rute wisata menggunakan algoritma dijkstra dengan graf reduksi dengan berbasis komputerisasi.

* + - 1. Tampilan utama dari sistem informasi pencarian rute wisata menggunakan algoritma dijkstra dengan graf reduksi. Detail lihat Gambar 14.

Gambar 14. Step 1

* + - 1. Klik tombol bertuliskan “Rute Tempat – Tempat Wisata Di Surabaya” untuk mencari rute wisata terpendek dengan 5 titik tujuan wisata. Detail lihat Gambar 15.

Gambar 15. Step 2

* + - 1. Pilih jam keberangkatan yang akan mempengaruhi tingkat kepadatan dan bobot tempat wisata itu. Setelah itu, centang 5 kotak yang merupakan titik tujuan wisata, yaitu Ekowisata Mangrove, Pura Agung Jagat Karana, Monumen Tugu Pahlawan, Pakuwon Food Festival, dan Lontong Balap Garuda Pak Gendut. Lalu klik Next pada pojok kanan bawah tampilan *interface*. Detail lihat Gambar 16.

Gambar 16. Step 3

* + - 1. Maka akan muncul tampilan *interface* yang menampilakan hasil pencarian rute wisata terpendek dengan 5 tujuan wisata. Detail lihat Gambar 17.

Gambar 17. Step 4

* 1. Perhitungan Kompleksitas Algoritma Dijkstra dan Algoritma Dijkstra dengan Graf Reduksi
     + - 1. Perhitungan Kompleksitas Algoritma Dijkstra

Kompleksitas waktu algoritma Dijkstra adalah , dengan adalah banyak titik dan adalah banyak sisi. Banyak titik dalam perhitungan algoritma Dijkstra adalah 25 titik dan karena graf G adalah graf komplit maka banyak sisi dalam perhitungan algoritma Dijkstra adalah . Maka kompleksitas waktu algoritma Dijkstra adalah .

* + - * 1. Perhitungan Kompleksitas Algoritma Dijkstra dengan Graf Reduksi

Kompleksitas waktu algoritma Dijkstra dengan graf reduksi adalah , dengan adalah banyak titik dan adalah banyak sisi. Banyak titik dalam perhitungan algoritma Dijkstra adalah 25 titik, namun sebelum proses pencarian dijkstra graf G direduksi dan mereduksi beberapa titik, sehingga titik pada graf berbobot G adalah 13 titik dan karena graf G adalah graf komplit maka banyak sisi dalam perhitungan algoritma Dijkstra adalah . Maka kompleksitas waktu algoritma Dijkstra dengan graf reduksi adalah .

**KESIMPULAN**

Pencarian rute perjalanan wisata terpendek dan terefisien di Kota Surabaya menggunakan algoritma *Dijkstra* dengan graf reduksi menggunakan sistem informasi ini membuat proses pencarian lintasan lebih cepat dan efisien dengan kompleksitas waktu sebesar .

Sistem informasi rute wisata terpendek di Kota Surabaya menggunakan algoritma Dijkstra dengan graf reduksi memudahkan masyarakat umum yang membutuhkan lintasan terpendek untuk menuju tempat – tempat wisata di Kota Surabaya.

**DAFTAR PUSTAKA**

Rodríguez-Puente, R. dan Lazo-Cortés, M. S. 2013. *Algorithm for shortest path search in Geographic Information Systems by using reduced graphs*. Springer Plus. Vol. 2 (291): hal 1-13.

Johson, Richard. 2002. *Matematika Diskrit (Discrete Mathemathics 4th Ed.)*. Jakarta: PT. Prenhallindo.

Budayasa, I Ketut. 2007. Teori Graph dan Aplikasinya. Surabaya: Unesa University Press.

Dinas Kebudayaan dan Pariwisata Kota Surabaya. 2017. *About Surabaya* (https://sparkling.surabaya. go.id/about-surabaya/the-history-of-surabaya, diakses 2 Oktober 2016, pukul 08.15 WIB)

BPS Kota Surabaya. *Jumlah dan Distribusi Penduduk* (http://sp2010.bps.go.id/index.php/site/tabel?tid=337&wid=3500000000, diakses 2 Oktober 2016, pukul 09.00 WIB)

Sadiq, Wasim dan Orlowska, Maria E. 2000. “Analyzing Process Models Using Graph Reduction Techniques”. *Elsevier: Information Systems*. Vol. 25 (2): hal. 117 – 134.

Mohring, Rolf H., Schilling, Heiko., Schutz, Birk., Wagner, Dorothea., dan Willhalm, Thomas. 2006. “Partitioning Graphs to Speed Up Dijkstra’s Algorithm”. *ACM Digital Library: Journal of Experimental Algorithmics*. Vol. 11.

Sadiq, Wasim dan Orlowska, Maria E. 1999. “Applying Graph Reduction Techniques for Identifying Structural Conflicts in Process Models”. *SpringerLink*.

Delling, Daniel., Sanders, Peter., Schultes, Dominik., Wagner, Dorothea. 2009. “Algorithmics of Large and Complex Networks, chapter Engineering Route Planning Algorithms”. *SpringerLink, part of Lecture Notes in Computer Science*. Vol. 5515: hal. 117 – 139.

Muhammad Tajuddin, Abdul Manan. "Rancang Bangun Sistem Informasi Monitoring Satuan Kerja Dalam Rangka *Good Governance*", Jurnal Matrik, 2017

Delling, Daniel., Wagner, Dorothea. 2009. “*Time-Dependent Route Planning*”. *SpringerLink, part of Lecture Notes in Computer Science*. Vol. 5868: hal 207 – 230.

Wagner, D., Willhalm, T. 2005. “*Drawing Graphs to Speed Up Shortest-Path Computations. In: Proceedings of the 7th Workshop on Algorithm Engineering and Experiments*”. SIAM, Philadelphia: hal 15 – 24.

Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L., Stein, Clifford. 2001. “*Introduction to Algorithms, Second Edition*”. London, England: The MIT Press.

Noshita, Kohei. 1985. “*A theorem on the expected complexity of dijkstra's shortest path algorithm*”. Elsevier: Journal of Algorithms. Vol. 6 (2): hal. 400 – 408.