

PENERAPAN ALGORITMA DIJKSTRA UNTUK MENENTUKAN RUTE TERPENDEK DARI PUSAT KOTA SURABAYA KE TEMPAT BERSEJARAH

ORIGINALITY REPORT

11%	11%	0%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

12%
★ jurnal.unidha.ac.id
Internet Source

Exclude quotes	Off	Exclude matches	< 1%
Exclude bibliography	On		

PENERAPAN ALGORITMA DIJKSTRA UNTUK MENENTUKAN RUTE TERPENDEK DARI PUSAT KOTA SURABAYA KE TEMPAT BERSEJARAH

by Yosefina Finsensia Riti

Submission date: 07-Mar-2022 02:02PM (UTC+0700)

Submission ID: 1778349838

File name: nentukan_Rute_Terpendek_Dari_Pusat_Kota_Ke_Tempat_Bersejarah.pdf (860.2K)

Word count: 4195

Character count: 24585

PENERAPAN ALGORITMA DIJKSTRA UNTUK MENENTUKAN RUTE TERPENDEK DARI PUSAT KOTA SURABAYA KE TEMPAT BERSEJARAH

Maria Chatrin Bunaen¹⁾, Hanna Pratiwi²⁾, Yosefina Finsensia Riti³⁾

^{1,2,3}Program Studi Ilmu Informatika, Universitas Katolik Darma Cendika

Jalan Dr. Ir. H. Soekarno No.201, Surabaya - Indonesia

email: ¹maria.chatrin@student.ukdc.ac.id, ²hannapратиwi@student.ukdc.ac.id, ³yosefina.riti@ukdc.ac.id

Abstract

Tempat bersejarah merupakan tempat yang sering dikunjungi warga sekitar bahkan wisatawan dari luar kota maupun mancanegara. Tempat bersejarah merupakan tempat yang wajib dikunjungi oleh wisatawan dari luar daerah tetapi terdapat permasalahan yang sering terjadi pada saat ingin mengunjungi destinasi tempat bersejarah ini. Permasalahannya adalah rute dari masing-masing lokasi tempat bersejarahnya sendiri dan juga ketidaktahuan akan rute-rute terpendek dan tercepat yang lebih efisien. Penelitian ini bertujuan untuk menentukan rute terpendek yang akan menggunakan Algoritma Dijkstra sebagai metode penelitiannya. Penelitian ini akan disertai dengan penentuan titik mana saja yang akan dilalui sehingga bisa mendapatkan rute terpendek sehingga bisa mengurangi waktu berkendara. Akan disertai juga cara penggunaan Algoritma Dijkstra dalam pencarian rute terpendek untuk lokasi tempat bersejarah ini. Hasil yang diperoleh dari penelitian ini terdapat 5 rute terpendek yang bisa dilalui untuk menuju tempat bersejarah melalui titik awal yaitu Stasiun Gubeng.

Keywords: Tempat Bersejarah, Rute Terpendek, Algoritma Dijkstra, Surabaya

Abstract

Historic places are places that are often visited by local residents and even tourists from outside the city and abroad. Historic places are places that must be visited by tourists from outside the area, but there are problems that often occur when they want to visit the destinations of this historic place. The problem is the route from each historical site location itself and also the ignorance of the shortest and fastest routes that are more efficient. This study aims to determine the shortest route that will use Dijkstra's Algorithm as a research method. This research will be accompanied by determining which points will be passed so that they can get the shortest route so as to reduce driving time. It will also be accompanied by how to use Dijkstra's Algorithm in finding the shortest route for the location of this historic place. The results obtained from this study there are 4 shortest routes that can be taken to get to historical places through the starting point Gubeng Station.

Keywords— Historic Sites, Shortest Route, Dijkstra's Algorithm, Surabaya

PENDAHULUAN

Tempat bersejarah merupakan lokasi yang sering dikunjungi orang banyak terkhusus orang-orang dari luar kota. Tempat bersejarah merupakan lokasi dimana bagian sejarah, militer, dan budaya yang dirawat dan dilestarikan karena nilai warisan budaya yang ada. Tempat

bersejarah biasanya dilindungi oleh hukum bahkan berbagai tempat bersejarah telah diakui sebagai situs bersejarah nasional. Tempat bersejarah ini bisa berupa bangunan atau struktur yang memiliki makna lokal, regional, atau nasional. Tempat bersejarah akan selalu dipelihara supaya pengunjung bisa menikmati lokasi

serta bisa belajar tentang warisan budaya atau belajar mengenai sejarah yang terjadi. Banyak sekali manfaat yang didapatkan dari tempat bersejarah, contohnya sebagai bukti nyata peristiwa sejarah yang bisa diamati saat ini, menjadi wawasan dan pengetahuan, membantu dalam bidang pendidikan dan ilmu pengetahuan, dan lain-lain. Salah satu kota yang memiliki banyak tempat bersejarah adalah Surabaya. Surabaya memiliki banyak macam tempat bersejarah yang bisa dikunjungi masyarakat Surabaya atau orang-orang dari luar kota Surabaya, diantaranya adalah Tugu Pahlawan, Monumen Kapal Selam, Kawasan Jembatan Merah, Gedung Internatio, Gedung Cerutu, Gedung Siola (Museum Surabaya), dan lain-lain. Namun permasalahan yang sering dialami para pengunjung adalah kesulitan dalam memilih rute terpendek dari titik awal (lokasi berangkat) menuju lokasi bersejarah yang ingin dikunjungi. Hal ini mengakibatkan perjalanan ke lokasi bersejarah tersebut memakan banyak waktu dijalan ditambah lagi padatnya pengendara jalan di kota Surabaya, sehingga ini berdampak juga pada jumlah lokasi bersejarah yang ingin dikunjungi. Bisa saja seorang atau sekelompok pengunjung ingin mendatangi banyak lokasi bersejarah sekaligus dalam sehari, tetapi karena salah memilih rute terpendek pengunjung tersebut hanya bisa berkunjung ke satu atau dua lokasi saja. Itulah mengapa pencarian rute terpendek sangat dibutuhkan dalam hal ini, dan penelitian yang akan dilakukan ini sangat berguna serta bermanfaat untuk kedepannya. Sehingga penelitian ini akan merujuk pada mencari rute terpendek dari pusat kota Surabaya ke sejumlah tempat bersejarah menggunakan Algoritma Dijkstra.

Tidak mudah memang memilih langsung rute terpendek melalui maps keadaan bisa saja berubah drastis saat

dilihat waktu tempuhnya sebentar, rute tidak berwarna orange atau merah yang menandakan kepadatan atau kemacetan, tetapi dilokasi ternyata situasi yang berbanding terbalik terjadi kepadatan lalu lintas sehingga mengakibatkan lamanya waktu perjalanan pengunjung. Terlebih lagi jika pengunjung tersebut merupakan wisatawan dari luar kota yang tidak familiar dengan keadaan padatnya pengguna jalan di kota Surabaya dan pastinya tidak begitu mengerti rute-rute jalan yang ada di Surabaya. Dengan adanya jalur terpendek yang bisa dilalui maka perjalanan menuju lokasi tempat bersejarah tidak akan memakan banyak waktu. Dengan adanya permasalahan tersebut penelitian yang ingin dilakukan ini akan sangat berguna serta berdampak positif dalam menentukan rute terpendek dari pusat kota Surabaya ke tempat bersejarah yang ada di Surabaya menggunakan Algoritma Dijkstra.

Penelitian terkait penentuan rute terpendek sudah banyak dilakukan dengan berbagai Algoritma diantaranya adalah algoritma A*(Ferdiansyah & Ahmad, 2013), Floyd (Nasional, Informasi, & Apriliani, 2018), Dynamic Programming(Di, n.d.). Khusus pada penelitian ini akan menggunakan Algoritma Dijkstra. Algoritma Dijkstra merupakan salah satu algoritma yang sangat populer dari teori graf untuk menentukan jalur terpendek. Algoritma Dijkstra mencari jalur terpendek dengan membandingkan bobot terkecil dari node awal sampai node terakhir atau tujuan, untuk menemukan jalur paling efektif dan efisien untuk ditempuh. Pada tahun 1950, menentukan jalur terpendek merupakan masalah yang telah banyak dibahas dan dipelajari. Pencarian jalur terpendek ini telah diterapkan di berbagai bidang untuk mengoptimalkan kinerja transmisi otomatis. Salah satu pencarian jalur terpendek yang paling menarik untuk dibahas adalah terkait transportasi.

Algoritma Dijkstra dipilih untuk penentuan rute terpendek karena dapat menentukan rute terpendek dari graf berbobot dimana bobotnya bernilai lebih besar dari nol (positif), dari titik awal dari semua titik yang diinginkan, sehingga dapat ditemukan rute terpendek dari titik awal sampai dengan titik tujuan. Tujuannya adalah menentukan representasi graf berarah untuk seluruh rute jalan sekaligus menentukan rute terpendeknya dari pusat kota menuju tempat – tempat bersejarah di Surabaya menggunakan Algoritma Dijkstra.

Penelitian tentang rute terpendek sudah banyak dilakukan, diantaranya dilakukan oleh Masri, dkk(Masri, Kiswanto, & Kusuma, 2019), menggunakan Algoritma Dijkstra. Sistem ini sangat berguna untuk pengunjung karena wisatawan bisa mengetahui rute terpendek yang bisa ditempuh dari dan menuju suatu tempat wisata yang diinginkan di daerah objek pariwisata Danau Toba dan Sekitarnya dengan menggunakan sistem ini. Dengan adanya penelitian rute terpendek dengan menggunakan Algoritma Dijkstra ini para wisatawan diharapkan dapat menambah informasi mengenai rute tercepat dan akhirnya para wisatawan memilih rute yang terbaik yang dapat mengefisienkan semua biaya dan mengetahui semua informasi objek wisata yang ada di Danau Toba.

Penelitian yang dilakukan oleh Harahap, dkk(Harahap, 2019), merupakan persoalan dalam menemukan jalur terpendek seiring dengan penghematan waktu yang tersingkat. Hal ini menjadi penting dalam kedinamisan masyarakat perkotaan. Jumlah rute yang ditempuh juga menjadi persoalan tersendiri untuk mencapai tempat tujuannya. Kita akan menentukan titik-titik manakah yang harus dilalui sehingga mendapatkan tempat tujuan dengan jarak terpendek dan penggunaan waktu yang tersingkat dengan

menggunakan algoritma Dijkstra. Pencarian lintasan terpendek merupakan persoalan optimasi. Nilai pada sisi *graph* bisa dinyatakan sebagai jarak antar kota. Lintasan terpendek bisa diartikan sebagai proses minimalisasi bobot pada lintasan.

Penelitian selanjutnya oleh Ferdiansyah, dkk(Ferdiansyah & Ahmad, 2013), tentang Penerapan Algoritma Dijkstra untuk Menentukan Rute Terpendek Pembacaan Water Meter Induk PDAM Tirta Kerta Raharja Kabupaten Tangerang. Data-data pendukung perhitungan algoritma Dijkstra ini menggunakan data yang berasal dari informasi bagian Pelanggan mengenai nama dan lokasi pelanggan dan data koordinat lokasi dari GoogleMaps. Penelitian ini menghasilkan aplikasi website untuk mencari rute terpendek ke beberapa lokasi pelanggan, sehingga diharapkan para petugas pembaca meter dapat dengan mudah dan cepat untuk mencapai lokasi tujuan dengan mengaksesnya melalui komputer desktop, handphone, smartphone atau iPad.

Berikutnya penelitian yang dilakukan oleh Gautama, dkk(Hermanto, 2020), tentang Penentuan Rute Terpendek dengan Menggunakan Algoritma Dijkstra pada Jalur Bus Sekolah. Salah satu optimalisasi yang bisa dilakukan adalah dengan menentukan jarak terpendek dari jalur bus sekolah. Semakin pendek jarak yang ditempuh tentunya berdampak pada biaya dan waktu. Hasil dari penelitian ini berupa total rute Bus Sekolah Denpasar untuk shift pagi sepanjang 95,5 km. Bus Sekolah tersebut menggunakan bahan bakar pertadex, dimana 1 liter dengan harga Rp 9.400 dapat menempuh 12,8 km. Jadi total biaya bahan bakar bus sekolah tersebut adalah Rp 70.132,-. Hasil ini dapat memberikan gambaran untuk Dinas Perhubungan kota Denpasar mengenai terapan matematika dalam menentukan rute yang dapat mengoptimalkan pengeluaran biaya bahan bakar.

Penelitian selanjutnya oleh Juniawan, dkk(Juniawan et al., 2020), tentang Penentuan Rute Terpendek Tujuan Wisata di Kota Toboali Menggunakan Algoritme Dijkstra Berbasis Web. Kesulitan yang dialami para wisatawan ini adalah dalam hal mengetahui dan menentukan jalan yang akan dilalui menuju ke tujuan wisatanya. Untuk itu dibuatlah sistem informasi geografis yang berbasis web dan dilengkapi dengan fungsi penentuan jarak terpendek menggunakan Algoritme Dijkstra agar dapat lebih cepat sampai pada tujuan wisata. Sistem dirancang berbasis web agar lebih banyak orang yang menggunakannya. Hasil dari penelitian berupa sistem informasi pariwisata Kota Toboali yang dapat menentukan jarak terpendek menuju lokasi wisata. Dari hasil pengujian algoritme dapat dibuktikan bahwa sistem mampu menentukan jarak terpendek dari titik awal yang ditentukan pengguna menuju titik tujuan wisatanya. Dari pengujian blackbox didapat hasil bahwa fungsional sistem memiliki kinerja yang baik.

Selanjutnya oleh Setiawan, dkk(Setiawan et al., 2019), mengenai Penentuan Rute Terpendek Menuju Pusat Perbelanjaan di Jakarta Menggunakan Algoritma Dijkstra. Permasalahan untuk memilih pusat perbelanjaan itu yang mendorong peneliti melakukan penelitian ini. Hasil yang didapatkan adalah sebuah aplikasi untuk mencari pusat perbelanjaan terdekat dengan menggunakan algoritma Dijkstra yang dapat digunakan pada *smartphone* agar memudahkan untuk mengetahui dan menemukan pusat perbelanjaan yang ada di sekitar pengguna.

Penelitian berikutnya oleh Syahputra (Syahputra, 2017), tentang Penentuan Rute Terpendek Pendistribusian Naskah Ujian Nasional Menggunakan Algoritma Dijkstra (Dinas Pendidikan Dan Pengajaran Kota Binjai). Pemilihan rute saat pendistribusian

Naskah Ujian Nasional menuju sekolah tujuan dilakukan dengan cara konvensional atau berdasarkan kesepakatan oleh petugas pendistribusi naskah tersebut, proses ini tidak dilakukan berdasarkan data yang akurat. Masalah rute terpendek dapat diselesaikan dengan sistem informasi geografis berbasis web menggunakan algoritma Dijkstra. Penelitian ini menghasilkan sebuah aplikasi untuk pencarian rutenya. Prototype perangkat lunak pencarian rute terpendek dapat menyimpan dan menampilkan hasil pencarian rute terpendek dari lokasi tujuan awal hingga lokasi tujuan akhir.

Dalam hal ini akan ditentukan jalan yang harus dilalui dari pusat kota menuju tempat bersejarah agar dapat menemukan tujuan yang diinginkan dengan jarak terpendek. Oleh karena itu, jarak terpendek dapat diartikan sebagai berat minimum jalur yaitu penjumlahan dari bobot semua busur membentuk jalan. Menentukan jalur terpendek merupakan salah satu permasalahan untuk menentukan sebuah jalur antara dua node dengan jumlah bobot minimal. Pada kasus pencarian jalur terpendek antara dua lokasi pada peta untuk melakukan perjalanan antara dua lokasi, Algoritma Dijkstra ini cocok untuk menentukan rute terpendek sehingga peneliti menggunakan “Penerapan Algoritma Dijkstra Untuk Menentukan Rute Terpendek Dari Pusat Kota Surabaya Ke Tempat Bersejarah.

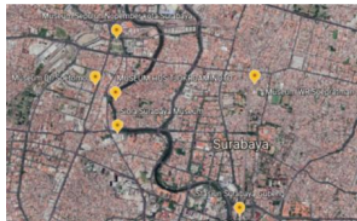
METODE PENELITIAN

2.1.Tahap Penelitian

Penelitian ini menggunakan metode yaitu dengan studi literature dan pengumpulan data. Studi literature dilakukan untuk memperoleh data yang diperlukan melalui referensi seperti buku, jurnal, artikel, maupun pencarian disitus internet. Pengumpulan data dilakukan

untuk mengumpulkan dan menganalisa data seperti informasi tempat bersejarah yang ada di Surabaya. Adapun lima tempat bersejarah menurut humas.surabaya.go.id yang saat ini telah menjadi museum dan akan menjadi *sample* dalam penelitian ini dengan titik awal Stasiun Gubeng Surabaya. Tempat-tempat tersebut adalah Museum Sepuluh Nopember, Museum Hos Tjokroaminoto, Museum WR. Soepratman, Museum Dr. Soetomo dan Museum Surabaya.

Berikut ini adalah titik – titik yang ditandai dengan warna kuning dengan menggunakan *google earth* yang akan menjadi bagian dari penelitian serta data yang merupakan data sekunder yang digunakan untuk menentukan jarak setiap titik yang berasal dari *google maps*:



Gambar 1. Titik – titik yang digunakan untuk penelitian

2.2.Path Terpendek

Pencari lintasan terpendek di dalam graf merupakan persoalan untuk mencari lintasan antara dua atau lebih simpul pada. Graf yang digunakan adalah graf berbobot (*weighted graph*), yaitu graf yang setiap sisinya diberikan suatu nilai atau bobot. Bobot pada sisi graf dapat menyatakan jarak antar kota. Dalam hal ini bobot harus bernilai positif, walau dalam hal lain dapat bernilai negatif. Lintasan terpendek dengan verteks awal dan verteks tujuan didefinisikan sebagai lintasan terpendek dengan bobot minimum dan berupa lintasan sederhana (*simple path*). Kata "terpendek" pada persoalan lintasan

terpendek berarti minimalisasi dari bobot pada suatu lintasan di dalam graf.

Terdapat beberapa jenis persoalan lintasan terpendek, antara lain:

- Lintasan terpendek antara dua buah simpul tertentu.
- Lintasan terpendek antara semua pasangan simpul.
- Lintasan terpendek dari simpul tertentu ke semua simpul yang lain.
- Lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu.

2.3.Algoritma Dijkstra

Algoritma Dijkstra dikstra ditemukan oleh *Edsger.Wybe Dijkstra* pada tahun 1959. Algoritma ini merupakan algoritma yang dapat memecahkan masalah pencarian jalur terpendek dari suatu graf pada setiap simpul yang bernilai tidak negatif. Dijkstra merupakan algoritma yang termasuk dalam algoritma *greedy*, yaitu algoritma yang sering digunakan untuk memecahkan masalah yang berhubungan dengan suatu optimasi. Dalam pencarian jalur terpendeknya algoritma dijkstra bekerja dengan mencari bobot yang paling minimal dari suatu graf berbobot, jarak terpendek akan diperoleh dari dua atau lebih titik dari suatu graf dan nilai total yang didapat adalah yang bernilai paling kecil. Misalkan G adalah graf berarah berlabel dengan titik-titik $V(G) = \{v_1, v_2, \dots, v_n\}$ dan path terpendek yang dicari adalah dari v_1 ke v_n . Algoritma Dijkstra dimulai dari titik v_1 . Dalam iterasinya, algoritma akan mencari satu titik yang jumlah bobotnya dari titik 1 terkecil. Titik-titik yang terpilih dipisahkan, dan titik-titik tersebut tidak diperhatikan lagi dalam iterasi berikutnya. Langkah-langkah dalam menentukan lintasan terpendek pada algoritma Dijkstra yaitu:

1. Pada awalnya pilih *node* sumber sebagai *node* awal, diinisialisasikan dengan „1“.
2. Bentuk tabel yang terdiri dari *node*, status, bobot, dan *predecessor*. Lengkapi kolom bobot yang diperoleh dari jarak *node* sumber ke semua *node* yang langsung terhubung dengan *node* sumber tersebut.
3. Jika *node* sumber ditemukan maka tetapkan sebagai *node* terpilih.
4. Tetapkan *node* terpilih dengan label permanen dan perbaharui *node* yang langsung terhubung.
5. Tentukan *node* sementara yang

Berikut adalah algoritma dijkstra yang dimisalkan dalam sebuah graf berbobot dengan n buah simpul dinyatakan dengan matriks $M=[m_{ij}]$, yang dalam hal ini:

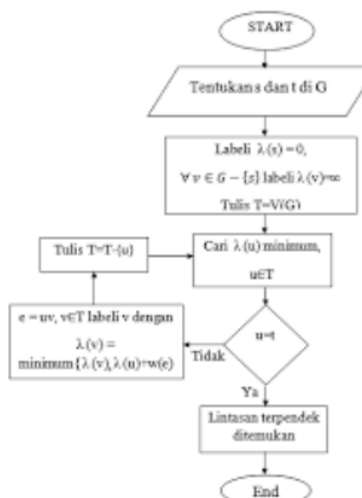
m_{ij} = bobot sisi (i,j) (pada graf tak berarah
 $m_{ij} = m_{ji}$)
 $m_{ii} = 0$
 $m_{ij} = \infty$, jika tidak ada sisi dari simpul i ke simpul j
 Selain matriks M, juga menggunakan tabel $S=[s_i]$, yang dalam hal ini:
 $s_i = 1$, jika simpul i termasuk ke dalam lintasan terpendek
 $s_i = 0$, jika simpul i tidak termasuk ke dalam lintasan terpendek
 Dan tabel $D=[d_i]$, yang dalam hal ini
 d_i = panjang lintasan dari simpul awal a ke simpul i

Berikut merupakan pseudocode untuk menghitung untuk menghitung jarak :

```
start
Langkah 0:
    float cost[N][N];
    int i, j;
    float w;
```

```
int source, tujuan, x, y;
Langkah 1:
    for (i=1;i<N;i++)
        for (j=1;j<N;j++)
            cost[i][j] = IN;
Langkah 2:
    for (x=1;x<N; x++) {
        for (y=x+1;y<N;y++) {
            printf("Masukan nilai dari simpul %d ke %d: ", x,y);
            scanf("%f", &w);
            cost [x][y] = cost[y][x] = w;
        }
    }
Langkah 3:
    printf("\n");
end
```

Berikut adalah flowchart dari algoritma dijkstra:



Gambar 1. Flowchart Algoritma Dijkstra (Sumber: Unnes Journal)

HASIL DAN PEMBAHASAN

Pada penelitian ini diperoleh graf dari dari Gambar 1. Titik – titik yang digunakan untuk penelitian yang mana titik akan diasumsikan sebagai titik awal serta tempat bersejarah dan sisi merupakan jarak dari setiap titik. Pada penelitian ini kami memperoleh beberapa hasil, yaitu:

1. Graf yang digunakan adalah graf berbobot yang dimana data jarak dari satu titik ke titik yang lain diambil dari *Google Maps*, jarak dari satu titik ke titik lainnya dinyatakan dalam km. Nilai pada jarak tersebut merupakan bobot dari setiap sisi sehingga graf tersebut dinyatakan graf berbobot.



Gambar 2. Graf Berbobot

2. Berikut kelima tempat bersejarah beserta titik awal dan keterangan tempat pada penelitian ini, yang dimana data tempat bersejarah ini diambil dari humas.surabaya.go.id.

No	Tempat	Keterangan
0	Stasiun Gubeng	Titik Awal
1	Siola Surabaya Museum (Museum Surabaya)	Tempat Bersejarah
2	Museum Hos Tjokroaminoto	Tempat Bersejarah
3	Museum Dr. Soetomo	Tempat Bersejarah
4	Museum Sepuluh Nopember	Tempat Bersejarah
5	Museum WR. Soepratman	Tempat Bersejarah

Table 1. Keterangan tempat dari titik yang ditandai dengan angka.

3. Tabel jarak dari graf yang saling terhubung yang dimana jarak dari satu titik ke titik lainnya dinyatakan dalam

km, serta tanda ∞ menyatakan jika tidak ada sisi dari antar titik.

	0	1	2	3	4	5
0	0	3,9	∞	∞	∞	4,3
1	3,9	0	1,7	∞	∞	3
2	∞	1,7	0	1,4	∞	∞
3	∞	∞	1,4	0	1,2	4
4	∞	∞	∞	1,2	0	4
5	4,3	3	∞	4	4	0

Table 2. Jarak yang saling terhubung.

4. Menentukan jarak terpendek ke titik menuju titik lainnya dengan perhitungan jarak terdekat dapat dimulai dari simpul dengan bobot terkecil.

- a. Jarak-jarak untuk menentukan jarak terpendek Stasiun Gubeng menuju Museum Dr. Soetomo maupun sebaliknya.

$$0 \rightarrow 1 = 3,9$$

$$0 \rightarrow 5 = 4,3$$

$$1 \rightarrow 2 = 1,7$$

$$1 \rightarrow 5 = 3$$

$$2 \rightarrow 3 = 1,4$$

$$5 \rightarrow 3 = 4$$

Dari jarak-jarak diatas ditentukan jarak yang memiliki bobot terkecil, kemudian dijumlahkan.

$$- 0 \rightarrow 3 = 0 \rightarrow 1 + 1 \rightarrow 2 + 2 \rightarrow 3 = 7$$

$$- 0 \rightarrow 3 = 0 \rightarrow 1 + 1 \rightarrow 5 + 5 \rightarrow 3 = 10,9$$

$$- 0 \rightarrow 3 = 0 \rightarrow 5 + 5 \rightarrow 3 = 8,3$$

Dari ketiga perhitungan diatas, perhitungan yang digunakan adalah yang menghasilkan bobot terkecil sehingga akan digunakan sebagai jarak terpendek. Jarak yang diperoleh dari perhitungan tersebut adalah $0 \rightarrow 3 = 0 \rightarrow 1 + 1 \rightarrow 2 + 2 \rightarrow 3 = 7$

- b. Lakukan pencarian perhitungan seperti yang ada diatas kemudian

buat kembali tabel jarak seperti Table 3. Jarak yang saling terhubung dengan mengisi setiap hasil perhitungan untuk memperoleh jarak terpendek dari satu titik ke titik lainnya.

	0	1	2	3	4	5
0	0	3,9	5,6	7	8,3	4,3
1	3,9	0	1,7	2,1	3,3	3
2	5,6	1,7	0	1,4	2,6	5,4
3	7	2,1	1,4	0	1,2	4
4	8,3	3,3	2,6	1,2	0	4
5	4,3	3	5,4	4	4	0

Table 4. Jarak antar titik.

- c. Setelah melakukan perhitungan dari setiap jarak dalam menentukan rute terpendek, maka akan dibuat tabel penentuan rute terpendek dari titik awalnya yaitu Stasiun Gubeng.

No	Titik	Tujuan	Rute	Jarak
1	Stasiun Gubeng (0)	Siola Surabaya Museum (Museum Surabaya)	Stasiun Gubeng – Siola Surabaya (Museum Surabaya)	3,9 km
2	Stasiun Gubeng (0)	Museum Hos Tjokroaminoto	Stasiun Gubeng – Siola Surabaya Museum (Museum Surabaya – Museum Hos Tjokroaminoto)	5,6 km
3	Stasiun Gubeng (0)	Museum Dr. Soetomo	Stasiun Gubeng – Siola Surabaya Museum (Museum Surabaya – Museum)	7 km

			Hos Tjokroaminoto – Museum Dr. Soetomo	
4	Stasiun Gubeng (0)	Museum Sepuluh Nopember	Stasiun Gubeng – Museum WR. Soepratman – Museum Sepuluh Nopember	8,3 km
5	Stasiun Gubeng (0)	Museum WR. Soepratman	Stasiun Gubeng – Museum WR. Soepratman	4,3 km

Table 4. Rute yang dilalui dari titik awal menuju tempat bersejarah

Berdasarkan hasil penelitian dalam pencarian rute terpendek yang telah dilakukan, didapatkan 5 rute terpendek yang dapat dilalui saat menuju ke tempat bersejarah dengan titik awal adalah Stasiun Gubeng. Setelah hasil perhitungan diatas maka selanjutnya akan dilakukan implementasi hasil perancangan, dalam hal ini penelitian akan menggunakan bahasa pemrograman. Pengimplementasian system bertujuan untuk mengetahui sampai sejauh mana hasil yang dikembangkan berlangsung. Penelitian ini menggunakan bahasa c++ yang dimana bahasanya terstruktur serta dapat digunakan dengan berbagai platform. Berikut merupakan code c++ yang diimplementasikan dari Algoritma Dijkstra:

```
#include <iostream>
#include<stdio.h>
#include<conio.h>
#include<ctype.h>
#include<string.h>
#include<math.h>
#define IN 99
#define N 6
```

```
using namespace std;

void dijkstra(float cost[][N], int source, int
tujuan) {
    float dist[N],d;
    int prev[N], selected[N]={0},i, m, min,
start, j;
    char path[N];
    //float total=0;
    for(i=1;i<N;i++) {
        dist[i] = IN;
        prev[i] = -1;
    }

    start = source;
    selected[start]=1;
    dist[start]=0;
    while(selected[tujuan] == 0) {
        min = IN;
        m = 0;
        for(i=1;i<N;i++) {
            d = dist[start] + cost[start][i];
            if (d< dist[i]&&selected[i]== 0) {
                dist[i] = d;
                prev[i] = start;
            }
            if (min> dist[i]&&selected[i]== 0) {
                min = dist[i];
                m = i;
            }
        }
        start = m;
        selected[start] = 1;
    }
    start = tujuan;
    j = 0;
    while(start != -1) {
        path[j++]
start+64;

        start = prev[start];
    }
    path[j] = '\0';

    printf("\nPath : %s", path);

    cout << "\nBobot yang dilalui: "
<<dist[tujuan];
```

```
}

int main() {
    float cost[N][N];
    int i, j;
    float w;
    int source, tujuan, x, y;

    for (i=1;i< N;i++)
    for (j=1;j< N;j++)
    cost[i][j] = IN;
    for (x=1;x< N; x++) {
        for (y=x+1;y<N;y++) {
            printf("Masukan nilai dari simpul
%d ke %d: ", x,y);
            scanf("%f", &w);
            cost [x][y] = cost[y][x] = w;
        }
        print ("\n");
    }
    printf("\n Masukan asal simpul : ");
    scanf("%d", &source);
    printf("\n Masukan tujuan simpul : ");
    scanf("%d", &tujuan);
    dijkstra(cost, source, tujuan);

    return 0;
}
```

Hasil berikut ini merupakan code diatas yang telah dijalankan:

Program akan meminta untuk memasukan bobot dari masing masing simpul serta asal simpul atau titik awal ke titik tujuan. *Gambar 4* menunjukkan hasil dari pencarian rute terpendek dari Stasiun Gubeng menuju Museum Dr. Soetomo, melewati dua simpul dan memiliki bobot 8.20 km.

```
Masukan nilai dari simpul 1 ke 3: 5.6
Masukan nilai dari simpul 1 ke 4: 7.0
Masukan nilai dari simpul 1 ke 5: 8.2

Masukan nilai dari simpul 2 ke 3: 1.7
Masukan nilai dari simpul 2 ke 4: 3.1
Masukan nilai dari simpul 2 ke 5: 4.3

Masukan nilai dari simpul 3 ke 4: 1.4
Masukan nilai dari simpul 3 ke 5: 2.6

Masukan nilai dari simpul 4 ke 5: 1.2

Masukan asal simpul : 1
Masukan tujuan simpul : 5

Path : EA
Jalur yang dilalui: 8.2
```

Gambar 4. Hasil percobaan dari program dijkstra

SIMPULAN

Dari penelitian dan hasil implementasi Algoritma Dijkstra berdasarkan lintasannya maka dapat diambil kesimpulan bahwa:

1. Didapatkan 5 rute terpendek yang dapat dilalui saat menuju ke tempat bersejarah dari awalnya adalah Stasiun Gubeng.
2. Permasalahan pencarian rute terpendek untuk mengunjungi tempat bersejarah dapat diselesaikan dengan Algoritma Dijkstra.
3. Dengan menggunakan metode Dijkstra ini, dapat ditentukan jalur terpendek dari sebuah jalur perjalanan dengan menentukan vertex awal dan vertex tujuan serta membandingkan nilai dari masing-masing vertex.

DAFTAR PUSTAKA

- Di, Abstract. (n.d.). *Pemilihan jalur terpendek dengan dynamic programming berbasis web*.
- Ferdiansyah, & Ahmad, Rizal. (2013). Penerapan Algoritma Dijkstra untuk Menentukan Rute Terpendek

- Pembacaan Water Meter Induk PDAM Tirta Kerta Raharja Kabupaten Tangerang. *Jurnal TICOM*, 2(1), 51–57.
- Harahap, Muhammad Khoiruddin. (2019). *Pencarian Jalur Terpendek dengan Algoritma Dijkstra*. 2, 18–23.
- Hermanto, Koko. (2020). *Penentuan Rute Terpendek dengan Menggunakan Algoritma Dijkstra pada Jalur Bus Sekolah*. 10(2), 116–123.
<https://doi.org/10.24843/JMAT.2020.v10.i02.p128>
- Juniawan, Fransiskus Panca, Sylfania, Dwi Yuny, Studi, Program, Informatika, Teknik, Korespondensi, Penulis, Terpendek, Rute, & Toboali, Pariwisata Kota. (2020). *SHORTEST PATH DETERMINATION OF THE TOURIST DESTINATION IN TOBOALI USING WEB-BASED DIJKSTRA ALGORITHM*. 7(1), 211–218.
<https://doi.org/10.25126/jtiik.202071954>
- Masri, Mahrizal, Kiswanto, Andika Pratomo, & Kusuma, Budhi Santri. (2019). *IMPLEMENTASI ALGORITMA DIJKSTRA DALAM PERANCANGAN PARIWISATA DANAU TOBA DAN SEKITARNYA*. 221–225.
- Nasional, Jurnal, Informasi, Sistem, & Apriliani, Vera. (2018). *Penerapan Algoritma Floyd Warshall dalam Aplikasi Penentuan Rute Terpendek Mencari Lokasi BTS (Base Tower Station) pada PT . GCI Palembang*. 02, 81–88.
- Setiawan, Jodi, Prakoso, Rezy S., Suryaningrum, Kristien Margi, Universitas, Mahasiswa, Mulia, Bunda, Universitas, Dosen, Mulia, Bunda, Raya, Jalan Lodan, Rw, R. T., Pademangan, Kec, & Utara, Kota Jakarta. (2019). *PERBELANJAAN DI JAKARTA MENGGUNAKAN ALGORITMA DIJKSTRA*. 21(3),

156–165.
Syahputra, Siswan. (2017). Penentuan
Rute Terpendek Pendistribusian
Naskah Ujian Nasional
Menggunakan Algoritma Dijkstra
(Dinas Pendidikan Dan Pengajaran

Kota Binjai). *Jurnal Teknik
Informatika Kaputama (JTik)*, 1(1).
Retrieved from
[http://jurnal.kaputama.ac.id/index.ph
p/JTIK/article/viewFile/38/34](http://jurnal.kaputama.ac.id/index.php/JTIK/article/viewFile/38/34)