

PENERAPAN ALGORITMA DIJKSTRA DALAM PENENTUAN LINTASAN TERPENDEK MENUJU UPT. PUSKESMAS CILODONG KOTA DEPOK

Cicih Sri Rahayu¹, Windu Gata², Sri Rahayu³, Agus Salim⁴, Arif Budiarto⁵

^{1,2,4,5} Program Studi Ilmu Komputer,

Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri

³ Program Studi Teknik Informatika,

Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri

¹14002466@nusamandiri.ac.id, ²windu@nusamandiri.ac.id, ³sriahayu.rry@nusamandiri.ac.id,

⁴14002462@nusamandiri.ac.id, ⁵14002437@nusamandiri.ac.id

ABSTRACT

Artikel:

Diterima: 17 Desember 2020

Direvisi: 01 September 2021

Diterbitkan: 06 September 2021

*Alamat Korespondensi:

14002466@nusamandiri.ac.id

One of the government's efforts in providing health to the community is the construction of health centers in each sub-district, and the community is expected to be able to take advantage of the health facilities provided by the government. One of the problems that exist in the community is determining the shortest distance to the puskesmas. In Depok City, there are 26 routes that can be passed from the 38 nodes or vertices to the Cilodong Health Center with the starting point of the Depok mayor's office. This study uses a survey research method to calculate the actual distance at each node or vertex, the purpose of this study is to determine the shortest path taken by the starting point from the Depok mayor's office to get to the Cilodong Health Center by applying the dijkstra algorithm. This dijkstra algorithm works by visiting all existing points and making a route if there are 2 routes to the same 1 point then the route that has the lowest weight is chosen so that all points have an optimal route. This quest continues until the final destination point. After doing this research and testing using a simple application to calculate the distance by applying the dijkstra algorithm, it was found that the shortest path taken to the destination is through the GDC Main Gate or on the test results in Iteration 26. From the results of this study, people can choose this closest route to save time when viewed from the distance of the existing track. For further research, it is expected to be able to compare two other algorithms and other parameters so that the closest route with the fastest time is obtained.

Keywords: *Dijkstra's Algorithm, Trajectory, Puskesmas*

ABSTRAK

Salah satu upaya pemerintah dalam menyelenggarakan kesehatan kepada masyarakat adalah dengan dibangunnya puskesmas disetiap kecamatan, dan masyarakat diharapkan bisa memanfaatkan fasilitas kesehatan yang telah diberikan pemerintah. Salah satu permasalahan yang ada pada masyarakat adalah menentukan jarak terpendek menuju puskesmas. Di Kota Depok untuk menuju puskesmas Cilodong ada 26 lintasan yang dapat dilalui dari 38 node atau vertex dengan titik awal kantor wali kota Depok. Penelitian ini menggunakan metode penelitian survei untuk menghitung jarak sebenarnya pada masing-masing node atau vertex, tujuan penelitian ini adalah untuk menentukan lintasan terpendek yang ditempuh dengan titik awal dari kantor wali kota Depok untuk sampai ke puskesmas Cilodong dengan menerapkan algoritma *dijkstra*. Algoritma *dijkstra* ini bekerja dengan mengunjungi semua titik yang ada dan membuat rutenya jika ada 2 rute menuju 1 titik yang sama maka rute yang memiliki bobot terendahlah yang dipilih sehingga semua titik mempunyai rute yang optimal. Pencarian ini berlangsung sampai titik tujuan terakhir. Setelah melakukan penelitian dan pengujian ini menggunakan aplikasi sederhana untuk menghitung jarak dengan menerapkan algoritma *dijkstra* didapatkan hasil bahwa telah ditemukan berupa lintasan terpendek yang ditempuh ke tempat tujuan yaitu melalui Gerbang Utama GDC atau pada hasil uji coba pada Iterasi 26 yaitu 5820 meter. Dari hasil penelitian ini diharapkan masyarakat bisa memilih rute terdekat untuk menghemat waktu jika dilihat dari jarak lintasan yang ada. Untuk penelitian selanjutnya diharapkan mampu membandingkan dua algoritma lain dan parameter yang lain sehingga didapatkan rute terdekat dengan waktu tercepat.

Kata Kunci: *Algoritma Dijkstra, Lintasan, Puskesmas*

I. PENDAHULUAN

Secara geografis Kota Depok berbatasan langsung dengan Kota Jakarta atau berada dalam lingkungan wilayah Jabotabek. Kota Depok sebagai wilayah termuda di Jawa Barat, mempunyai luas wilayah sekitar 200,29 km². Kota Depok terdiri dari 11 kecamatan salah satunya adalah kecamatan Cilodong, Salah satu bentuk upaya pemerintah pusat maupun daerah dalam menyelenggarakan kesehatan kepada masyarakat maka disetiap kecamatan dibangun instansi pemerintah sebagai unit penyelenggara pelayanan kesehatan masyarakat, yakni Pusat Kesehatan Masyarakat atau yang biasa disebut Puskesmas. Peraturan Menteri Kesehatan Republik Indonesia Nomor 75 Tahun 2014 tentang Puskesmas adalah fasilitas pelayanan kesehatan yang menyelenggarakan upaya

kesehatan masyarakat dan upaya kesehatan perseorangan tingkat pertama, dengan lebih mengutamakan upaya promotif dan preventif, untuk mencapai derajat kesehatan masyarakat yang setinggi-tingginya di wilayah kerjanya. Puskesmas merupakan unit pelaksana teknis kesehatan di bawah supervisi Dinas Kesehatan Kabupaten/Kota. Secara umum, mereka harus memberikan pelayanan preventif, promotif, kuratif sampai dengan rehabilitatif baik melalui Upaya Kesehatan Perorangan (UKP) atau Upaya Kesehatan Masyarakat (UKM) [1]. Kota Depok mendirikan fasilitas kesehatan UPT. Puskesmas yang terletak di Perum GDC, Jl. Boulevard Grand Depok City, Jatimulya, Kec. Cilodong, Kota Depok, Jawa Barat, sebagai salah satu Upaya Kesehatan masyarakat (UKM) warga Kota Depok di sekitar kecamatan Cilodong. Untuk sampai di Unit Pelayanan

Terpadu (UPT) Puskesmas Cilodong warga dihadapkan dengan pilihan lintasan yang banyak dan ini membuat bingung warga dalam menggunakan lintasan mana yang terpendek untuk sampai di tujuan. Penentuan sebuah jalur terpendek merupakan hal yang penting dan dibutuhkan sehubungan dengan optimasi waktu yang digunakan serta beberapa penghematan dibidang lainnya. Dengan jalur terpendek yang dilalui, membuat pekerjaan lebih efektif, cepat dan dapat tentunya terjadi penghematan biaya. Jalur terpendek dapat diartikan sebagai nilai minimal dari suatu lintasan, yaitu jumlah nilai dari keseluruhan bentuk lintasan. Untuk membantu menentukan lintasan terpendek dapat memilih jalur yang terpendek dari tempat asal ke tujuan. Hal ini terkadang tidak dapat membantu secara maksimal dikarenakan banyaknya jumlah jalan yang harus dipilih dan tidak dapat diperkirakan jarak tempuh pada jalur itu [2], namun dalam optimasi pemilihan jalur atau lintasan terpendek yang umum digunakan adalah algoritma *Dijkstra* yang juga merupakan algoritma *Greedy* bisa menentukan lintasan terpendek. Algoritma *Dijkstra* telah banyak digunakan dalam penelitian untuk memecahkan masalah jalur terpendek, termasuk menggunakan model grafik dan *Dijkstra* algoritma dalam menentukan rute kendaraan di jalan tol [3], dan menggunakan kombinasi node berdasarkan algoritma *Dijkstra* untuk menentukan jarak terpendek antar kota di Pulau Jawa [4]. Pada penelitian sebelumnya, algoritma *Dijkstra* digunakan pada penerapan algoritma *dijkstra* untuk penentuan jalur terbaik evakuasi tsunami – studi kasus: kelurahan Sanur Bali [5]. Penerapan algoritma *dijkstra* juga digunakan oleh penelitian sebelumnya dalam penentuan rute terpendek menuju masjid di perumahan citra indah city [6].

Berdasarkan permasalahan tersebut, maka diperlukan menentukan lintasan terpendek yang ditempuh warga untuk sampai ke puskesmas Cilodong. Tujuan dari penelitian ini adalah untuk menentukan lintasan terpendek yang ditempuh warga untuk sampai ke Puskesmas Cilodong dengan metode yang digunakan menerapkan algoritma *dijkstra* yang mampu menghasilkan rute terpendek bagi warga untuk menuju ke puskesmas Cilodong.

Kelebihan dan kelemahan implementasi menggunakan algoritma *dijkstra*, diantaranya adalah:

- a. Algoritma *dijkstra* dapat digunakan untuk menentukan jalur alternatif, apabila jalur utama mengalami hambatan.
- b. Algoritma *dijkstra* dapat digunakan untuk memberikan solusi permasalahan rute terpendek dan rute terpanjang, elemen-elemen (bobot) dari rute tersebut berupa jarak tempuh, biaya, maupun hal lainnya.
- c. Algoritma *Dijkstra* tidak dapat memprediksi kemacetan suatu rute perjalanan menuju suatu tempat.

Dalam menentukan lintasan terpendek yang ditempuh bisa diilustrasikan menggunakan aplikasi pencarian rute terdekat menuju puskesmas Cilodong sederhana yang dibuat dengan bahasa pemrograman html dan css.

Isi naskah ini adalah hasil asli dari penelitian saya sendiri dengan menggunakan referensi beberapa jurnal yang telah saya cantumkan.

II. METODOLOGI

Metode yang digunakan pada penelitian ini adalah dengan metode survei untuk menghitung jarak sebenarnya pada masing-masing node atau vertex dengan bantuan google maps melalui link: <https://www.google.co.id/maps> maka di dapatkan data seperti pada tabel 2. Algoritma *dijkstra* menyelesaikan masalah pencarian jalur/lintasan terpendek, sebuah masalah untuk mencari jalur/lintasan antara dua simpul dalam sebuah graf berbobot dengan jumlah total bobot (misal berupa jarak) terkecil [5]. Pencarian rute terpendek menggunakan algoritma *dijkstra* dapat digambarkan menggunakan *flowchart* [6]. Algoritma ini menyelesaikan masalah mencari sebuah lintasan terpendek dari *vertex* a ke *vertex* z dalam *graph* berbobot, bobot tersebut adalah bilangan positif jadi tidak dapat dilalui oleh node negatif, namun jika terjadi demikian, maka penyelesaian yang diberikan adalah *infiniti* atau jumlah tak terbatas [7]. Node dalam peta yang sudah membentuk *graph* berarah, dapat diketahui hubungan antar node. Hubungan antar node secara lengkap yang membentuk *graph* kemudian dibuat *matrix adjacency* yaitu hubungan antar node dan bobot [8]. Algoritma *dijkstra* untuk menentukan alternatif jalur yang dapat dipilih untuk menghindari kemacetan lalu lintas [9].

algoritma *Dijkstra* melakukan kalkulasi terhadap semua kemungkinan bobot terkecil dari setiap titik [10].

Dijkstra adalah algoritma yang digunakan untuk mencari lintasan terpendek pada sebuah graf berarah. Contoh penerapan algoritma ini adalah lintasan terpendek yang menghubungkan antara dua kota berlainan tertentu. Kasus ini sering disebut *Single-source Single Destination Shortest Path Problems* [11].

Algoritma *Dijkstra* merupakan algoritma yang digunakan untuk menentukan jarak terpendek dari satu vertex ke vertex yang lainnya pada suatu graph berbobot, jarak antar vertex adalah nilai bobot dari setiap edge pada graph. Suatu bobot harus bernilai positif (bobot ≥ 0). Algoritma *Dijkstra* ditemukan oleh Edger Wybe *Dijkstra*. Algoritma *Dijkstra* dikenal juga sebagai algoritma greedy yaitu algoritma yang penyelesaian masalah dengan mencari nilai maksimum. Shortest path merupakan persoalan untuk mencari lintasan terpendek antara dua buah vertex pada graph berbobot yang memiliki gabungan nilai jumlah bobot pada edge graph yang dilalui dengan jumlah yang paling minimum atau dapat dinyatakan juga sebagai berikut :

diberikan sebuah graph berbobot (dengan himpunan vertex V , himpunan edge E , dan fungsi bobot bernilai bilangan real yang dapat ditulis menggunakan rumus (1) berikut ini :

$$f: E \rightarrow \mathbb{R} \quad (1)$$

dan diberikan elemen v' dari V , sehingga dapat dicari sebuah lintasan P dari v ke setiap v dari V [2].



algoritma *Dijkstra* pada penelitian ini dapat diilustrasikan dengan diagram berikut [12]:

Gambar 1. Langkah-langkah penelitian

Tahapan penelitian dan skenario pengujian yang dilakukan peneliti melalui beberapa tahapan sebagai berikut :

2.1 Tentukan Peta Dasar

Langkah pertama adalah menentukan peta dasar sebagai bahan penelitian. Dalam penelitian ini peta kecamatan Cilodong dan sekitarnya.

2.2 Menetapkan Node Dan Label

Langkah kedua adalah menetapkan node dan memberi label pada setiap node. Berdasarkan peta dasar yang telah ditentukan, yang dianggap sebagai simpul adalah setiap persimpangan jalan menuju puskesmas Cilodong, dan diberi label pada setiap simpul.

2.3 Ukur Jarak Antar Node

Proses selanjutnya adalah mengukur jarak antar node, dengan menggunakan algoritma *Dijkstra*. Algoritma *Dijkstra* menyelesaikan masalah pencarian lintasan terpendek, sebuah masalah untuk mencari jarak antara dua simpul dalam sebuah graf berbobot dengan jumlah total bobot (Misal berupa jarak) terkecil, dengan cara mencari jarak terpendek antara simpul awal dengan simpul-simpul lainnya sehingga lintasan yang terbentuk dari simpul awal sampai simpul tujuan memiliki jumlah bobot terkecil.

2.4 Pengujian Algoritma Dijkstra

Langkah-langkah pengujian menggunakan aplikasi sederhana untuk menghitung jarak dengan menerapkan algoritma *Dijkstra* untuk menuju puskesmas Cilodong.

2.5 Analisis Hasil Tes

Setelah semua proses telah dilakukan, *Dijkstra* terakhir adalah menganalisis hasil pengujian pengukuran rute terpendek menggunakan algoritma *Dijkstra* dan menarik kesimpulan dari hasil analisis tersebut.

III. HASIL DAN PEMBAHASAN

Tahapan penelitian dan skenario pengujian yang dilakukan peneliti melalui beberapa tahapan sebagai berikut :

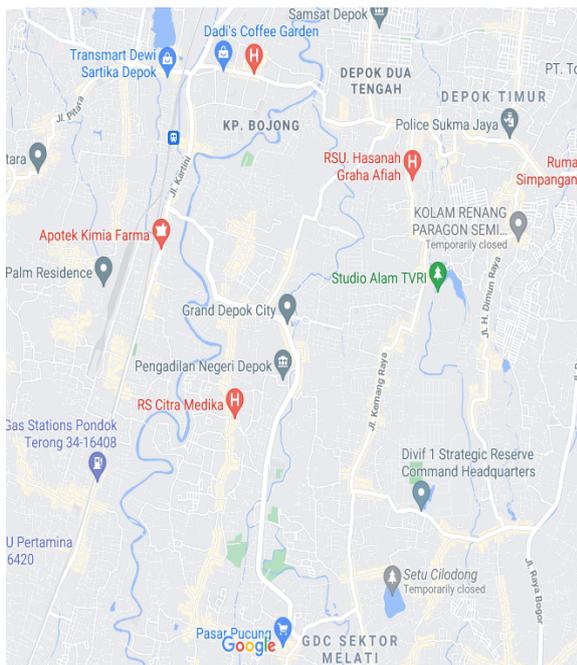
3.1 Peta Dasar

Studi kasus pada penelitian ini dilakukan di kota Depok, Jawa Barat dengan longitude and latitude yang terletak pada -6.394558, 106.822618.

Data yang digunakan adalah berupa data jalan-jalan yang dapat di lalui untuk menuju puskesmas Cilodong dengan titik awal kantor wali kota Depok. Jalan-jalan yang dapat dilalui ada 26 lintasan yang diwakili dengan istilah iterasi.

Dalam penelitian ini menggunakan Peta dasar kota Depok yang diambil dari <https://www.google.co.id/maps>. Untuk menghitung jarak antara node satu dengan node lainnya menggunakan istilah inialisasi antara node dan jaraknya menggunakan google maps.

Proses penghitungan jumlah dari masing-masing 26 lintasan atau iterasi dengan cara menambahkan masing-masing inialisasi antara node atau vertex.

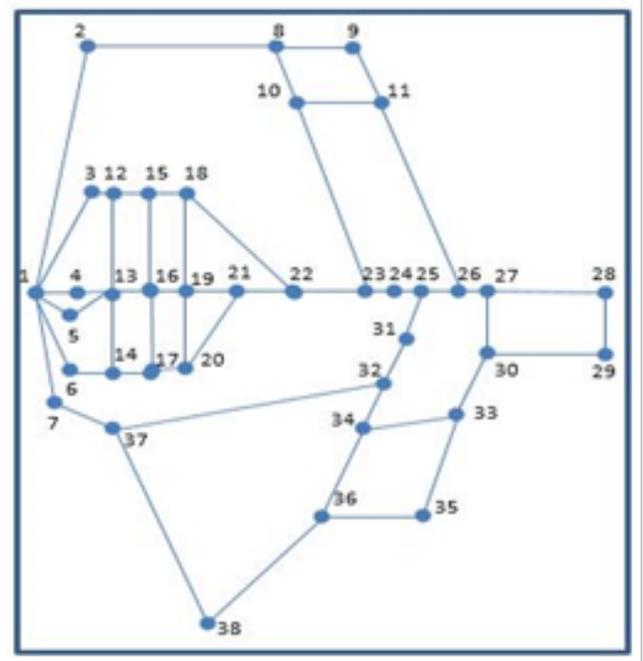


Gambar 2. Peta kecamatan cilodong dan sekitarnya

3.2 Proses Identifikasi Node / Vertex Dan Label

Berdasarkan Peta dasar kota Depok yang diambil dari <https://www.google.co.id/maps>, Ukuran setiap tepi dihitung menggunakan google maps. Dari proses Identifikasi diperoleh 38 node. Proses

identifikasi nodes / vertex pada penelitian ini dapat diilustrasikan dengan gambar berikut [12]:



Gambar 3. Identifikasi nodes / vertex

Tabel 1. Keterangan node / vertex

No. Node	Keterangan
1	Kantor wali kota Depok
2	Jl. Ir. H. Juanda
3	Jl. Dahlia
4	Jl. Siliwangi
5	Jl. Kemuning
6	Jl. Pemuda
7	Gerbang Utama GDC
8	Jl. Sentosa
9	Gema Insani
10	Jl. Proklamasi
11	RM. Takana 88
12	Jl. Bungur
13	Warkop Barokah
14	Villa Hijau
15	Jl. Cempaka
16	Mediafarma
17	Elmay Studio photo
18	Mie ayam berkat
19	RS. Hermina
20	Depok Data RC
21	Wr. Padang Bungo Talago
22	Dacler Sumber Mas
23	Masjid Alhuda
24	Jl. KSU
25	Jl. Raden Saleh
26	Polsek Sukmajaya

No. Node	Keterangan
27	Jl. H. Dimun.
28	Simpangan Depok
29	Simpangan Cilodong
30	Masjid Jami At-taqwa
31	Kemang Swatama
32	Jl. Madrasah
33	Pertigaan H. Musa
34	Toko Kue Kanzah
35	Klinik Melati
36	Soto Bogor H. Jojon
37	Jl. Mandor Samin
38	Puskesmas Cilodong

3.3 Proses Pengukuran Jarak Antar Node
 Pengukuran jarak ke setiap node dilakukan dengan menggunakan Google Maps sehingga diperoleh data dari jarak node sebagai berikut:

Tabel 2. Data jarak antar node

Node		Inisialisasi Antara Node	Jarak (meter)
a	b		
1	2	J1	2200
1	3	J2	220
1	4	J3	600
1	5	J4	700
1	6	J5	1000
1	7	J6	2100
2	8	J7	1100
8	9	J8	1000
3	12	J9	110
12	15	J10	170
15	18	J11	130
12	13	J12	300
15	16	J13	250
18	19	J14	260
18	22	J15	500
8	10	J16	1500
9	11	J17	1200
10	11	J18	1700
10	23	J19	1300
11	26	J20	1600
4	13	J21	190
13	16	J22	190
16	19	J23	200
19	21	J24	190
21	22	J25	100
22	23	J26	1300
23	24	J27	180
24	25	J28	260
25	26	J29	1400
26	27	J30	270
27	28	J31	1200
5	13	J32	270
13	14	J33	350

Node		Inisialisasi Antara Node	Jarak (meter)
a	b		
16	17	J34	350
19	20	J35	350
20	21	J36	450
6	14	J37	190
14	17	J38	190
17	20	J39	200
25	31	J40	2500
31	32	J41	800
27	30	J42	3800
28	29	J43	3000
29	30	J44	350
30	34	J45	550
34	33	J46	1300
32	33	J47	1300
34	36	J48	550
33	35	J49	800
35	36	J50	1000
7	37	J51	800
37	38	J52	2500
37	32	J53	1300
36	38	J54	900

3.4 Pengujian Algoritma Dijkstra

Langkah-langkah pengujian menggunakan aplikasi sederhana untuk menghitung jarak dengan menerapkan algoritma *dijkstra* untuk menuju puskesmas Cilodong yaitu dengan cara sebagai berikut:

1. Menentukan node atau Vertex yang dapat dilalui dengan menekan Menu 1



No Node	Keterangan	No Node	Keterangan
1	Kantor wali kota Depok	20	Depok Data RC
2	Jl. Ir. H. Juanda	21	Wr. Padang Bungo Talago
3	Jl. Dahlia	22	Daeler Sumber Mas
4	Jl. Siliwangi	23	Masjid Alhuda
5	Jl. Kemuning	24	Jl. KSU
6	Jl. Pemuda	25	Jl. Raden Saleh
7	Gerbang Utama GDC	26	Polsek Sukmajaya
8	Jl. Sentosa	27	Jl. H. Dimun.
9	Gema Insani	28	Simpangan Depok
10	Jl. Proklamasi	29	Simpangan Cilodong
11	RM. Takana 88	30	Masjid Jami At-taqwa
12	Jl. Bungur	31	Kemang Swatama
13	Warkop Barokah	32	Jl. Madrasah
14	Villa Hijau	33	Pertigaan H. Musa
15	Jl. Cempaka	34	Toko Kue Kanzah
16	Mediafarma	35	Klinik Melati
17	Elmay Studio photo	36	Soto Bogor H. Jojon
18	Mie ayam berkat	37	Jl. Mandor Samin
19	RS. Hermina	38	Puskesmas Cilodong

Gambar 4. Menentukan node atau vertex

2. Inisialisasi antara node atau vertex dengan menekan Menu 2



Node A	Node B	Inisialisasi Antara Node	Jarak (meter)	Node A	Node B	Inisialisasi Antara Node	Jarak (meter)
1	2	J1	2200	24	25	J28	260
1	3	J2	220	25	26	J29	1400
1	4	J3	600	26	27	J30	270
1	5	J4	700	27	28	J31	1200
1	6	J5	1000	5	13	J32	270
1	7	J6	2100	13	14	J33	350
2	8	J7	1100	16	17	J34	350
8	9	J8	1000	19	20	J35	350
3	12	J9	110	20	21	J36	450
12	15	J10	170	6	14	J37	190
15	18	J11	130	14	17	J38	190
12	13	J12	300	17	20	J39	200
15	16	J13	250	25	31	J40	2500
18	19	J14	260	31	32	J41	800
18	22	J15	500	27	30	J42	3800
8	10	J16	1500	28	29	J43	3000
9	11	J17	1200	29	30	J44	350
10	11	J18	1700	30	34	J45	550
10	23	J19	1300	34	33	J46	1300
11	26	J20	1600	32	33	J47	1300
4	13	J21	190	33	34	J48	1300
13	16	J22	190	34	35	J49	550
16	19	J23	200	33	36	J50	800
19	21	J24	190	35	36	J51	1000
21	22	J25	100	37	38	J52	2500
22	23	J26	1300	37	32	J53	1300
23	24	J27	180	36	38	J54	900

Gambar 5. Inisialisasi antara node atau vertex

3. Menentukan lintasan yang dapat dilalui atau disebut iterasi dengan menekan Menu 3

Aplikasi Sederhana Menghitung Jarak		Menu 1 Menu 2 Menu 3 Menu 4 Menu 5	
Iterasi	Inisialisasi Antar Node/Vertex Yang Di Lalui	Iterasi	Inisialisasi Antar Node/Vertex Yang Di Lalui
1	J1,J7,J16,J19,J27,J28,J40,J41,J47,J48,J54	14	J2,J9,J10,J11,J14,J24,J25,J26,J27,J28,J40,J41,J53,J52
2	J1,J7,J16,J19,J27,J28,J40,J41,J53,J52	15	J3,J21,J22,J23,J24,J25,J26,J27,J28,J40,J41,J47,J48,J54
3	J1,J7,J16,J18,J20,J30,J42,J45,J49,J50,J54	16	J3,J21,J22,J23,J24,J25,J26,J27,J28,J40,J41,J53,J52
4	J1,J7,J16,J18,J20,J30,J42,J45,J46,J48,J54	17	J4,J32,J22,J23,J24,J25,J26,J27,J28,J40,J41,J47,J48,J54
5	J1,J7,J8,J17,J20,J30,J31,J43,J44,J45,J49,J50,J54	18	J4,J32,J22,J23,J24,J25,J26,J27,J28,J40,J41,J53,J52
6	J1,J7,J8,J17,J20,J30,J31,J43,J44,J45,J46,J48,J54	19	J5,J37,J33,J22,J23,J24,J25,J26,J27,J28,J40,J41,J47,J48,J54
7	J2,J9,J10,J11,J15,J26,J27,J28,J40,J41,J47,J48,J54	20	J5,J37,J33,J22,J23,J24,J25,J26,J27,J28,J40,J41,J53,J52
8	J2,J9,J10,J11,J15,J26,J27,J28,J40,J41,J53,J52	21	J5,J37,J38,J34,J23,J24,J25,J26,J27,J28,J40,J41,J47,J48,J54
9	J2,J9,J12,J22,J23,J24,J25,J26,J27,J28,J40,J41,J47,J48,J54	22	J5,J37,J38,J34,J23,J24,J25,J26,J27,J28,J40,J41,J53,J52
10	J2,J9,J12,J22,J23,J24,J25,J26,J27,J28,J40,J41,J53,J52	23	J5,J37,J38,J39,J36,J25,J26,J27,J28,J40,J41,J47,J48,J54
11	J2,J9,J10,J13,J23,J24,J25,J26,J27,J28,J40,J41,J47,J48,J54	24	J5,J37,J38,J39,J36,J25,J26,J27,J28,J40,J41,J53,J52
12	J2,J9,J10,J13,J23,J24,J25,J26,J27,J28,J40,J41,J53,J52	25	J6,J51,J53,J47,J48,J54
13	J2,J9,J10,J11,J14,J24,J25,J26,J27,J28,J40,J41,J47,J48,J54	26	J6,J51,J52

Gambar 6. Menentukan iterasi

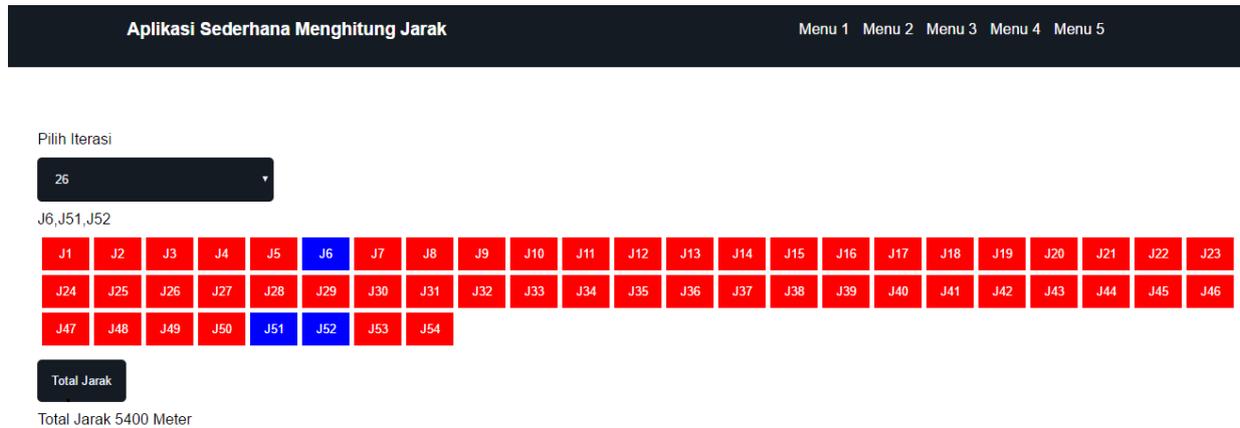
4. Menguji lintasan yang dapat dilalui atau disebut iterasi dengan menekan Menu 4

Gambar 7. Menguji iterasi dengan menghasilkan jarak

Dalam proses pengujian ini dengan menekan tombol hijau yang dimana masing-masing tombol mewakili nilai masing-masing jarak antara node atau vertex.

Gambar 8. Hasil Uji Iterasi 1

Hasil penghitungan pada iterasi 1 adalah 7150 meter, untuk perhitungan iterasi 2 sampai iterasi 26 prosesnya sama



Gambar 9. Hasil uji iterasi 26

- Untuk hasil seluruh uji iterasi 1 sampai dengan 26 dengan menekan Menu 5
 Dengan melakukan pengujian pada penelusuran jarak terpendek dari algoritma *dijkstra* dengan melewati lintasan dari 6 jalur utama dari Kantor wali kota Depok menuju puskesmas Cilodong yaitu Jl. Ir. H. Juanda, Jl. Dahlia, Jl. Siliwangi, Jl. Kemuning, Jl. Pemuda, Gerbang Utama GDC dengan 38 node/vertex. Dari penelusuran jarak terpendek dengan algoritma *dijkstra*, maka di dapat hasil seperti pada contoh tabel 3 dan 4, sebagai berikut:

Pengujian akan berhenti karena semua node/vertex sudah terpilih, sehingga akan menghasilkan lintasan terpendek yang dapat dilalui dari node /vertex 1 kesetiap node yang ada.



Nama Iterasi	Jarak Total	Nama Iterasi	Jarak Total
Iterasi 1	12590	Iterasi 14	10020
Iterasi 2	13640	Iterasi 15	9260
Iterasi 3	15420	Iterasi 16	10310
Iterasi 4	15470	Iterasi 17	9440
Iterasi 5	15170	Iterasi 18	10490
Iterasi 6	15220	Iterasi 19	10010
Iterasi 7	8920	Iterasi 20	11060
Iterasi 8	9970	Iterasi 21	10010
Iterasi 9	9100	Iterasi 22	11060
Iterasi 10	10150	Iterasi 23	9920
Iterasi 11	9030	Iterasi 24	10970
Iterasi 12	10080	Iterasi 25	6950
Iterasi 13	8970	Iterasi 26	5400

Jarak Terjauh : Iterasi 4 dengan jarak total 15470

Jarak Terdekat : Iterasi 26 dengan jarak total 5400

Gambar 10. Hasil uji iterasi 26

3.5 Analisis Hasil Tes

Hasil pengujian algoritma *dijkstra* dengan melewati lintasan dari 6 jalur utama dari kantor wali kota Depok menuju puskesmas Cilodong. Dari 6 jalur utama yang dapat dilalui dari kantor wali kota Depok menuju UPT Puskesmas Cilodong adalah :

Tabel 3. Enam jalur utama menuju UPT puskesmas cilodong

No	Nama Jalur Utama
1.	Jl. Ir. H. Juanda
2.	Jl. Dahlia
3.	Jl. Siliwangi
4.	Jl. Kemuning
5.	Jl. Pemuda
6.	Gerbang Utama GDC

Dengan 38 node/vertex menghasilkan 26 lintasan yang dapat di lalui menuju UPT Puskesmas Cilodong adalah sebagai berikut :

Tabel 4. Node /vertex yang dapat di lalui menuju UPT puskesmas cilodong

Iterasi	Inisialisasi Antar Node/Vertex Yang Di Lalui
1	J1, J7, J16, J19, J27, J28, J40, J41, J47, J48, J54
2	J1, J7, J16, J19, J27, J28, J40, J41, J53, J52
3	J1, J7, J16, J18, J20, J30, J42, J45, J49, J50, J54
4	J1, J7, J16, J18, J20, J30, J42, J45, J46, J48, J54
5	J1, J7, J8, J17, J20, J30, J31, J43, J44, J45, J49, J50, J54
6	J1, J7, J8, J17, J20, J30, J31, J43, J44, J45, J46, J48, J54
7	J2, J9, J10, J11, J15, J26, J27, J28, J40, J41, J47, J48, J54
8	J2, J9, J10, J11, J15, J26, J27, J28, J40, J41, J53, J52
9	J2, J9, J12, J22, J23, J24, J25, J26, J27, J28, J40, J41, J47, J48, J54
10	J2, J9, J12, J22, J23, J24, J25, J26, J27, J28, J40, J41, J53, J52
11	J2, J9, J10, J13, J23, J24, J25, J26, J27, J28, J40, J41, J47, J48, J54
12	J2, J9, J10, J13, J23, J24, J25, J26, J27, J28, J40, J41, J53, J52
13	J2, J9, J10, J11, J14, J24, J25, J26, J27, J28, J40, J41, J47, J48, J54
14	J2, J9, J10, J11, J14, J24, J25, J26, J27, J28, J40, J41, J53, J52

15	J3, J21, J22, J23, J24, J25, J26, J27, J28, J40, J41, J47, J48, J54
16	J3, J21, J22, J23, J24, J25, J26, J27, J28, J40, J41, J53, J52
17	J4, J32, J22, J23, J24, J25, J26, J27, J28, J40, J41, J47, J48, J54
18	J4, J32, J22, J23, J24, J25, J26, J27, J28, J40, J41, J53, J52
19	J5, J37, J33, J22, J23, J24, J25, J26, J27, J28, J40, J41, J47, J48, J54
20	J5, J37, J33, J22, J23, J24, J25, J26, J27, J28, J40, J41, J53, J52
21	J5, J37, J38, J34, J23, J24, J25, J26, J27, J28, J40, J41, J47, J48, J54
22	J5, J37, J38, J34, J23, J24, J25, J26, J27, J28, J40, J41, J53, J52
23	J5, J37, J38, J39, J36, J25, J26, J27, J28, J40, J41, J47, J48, J54
24	J5, J37, J38, J39, J36, J25, J26, J27, J28, J40, J41, J53, J52
25	J6, J51, J53, J47, J48, J54
26	J6, J51, J52

Tabel 5. Contoh hasil iterasi ke 1 sampai dengan 26 algoritma *dijkstra*

Nama Iterasi	Jarak (meter)
Jumlah Iterasi 1	12590
Jumlah Iterasi 2	13640
Jumlah Iterasi 3	15420
Jumlah Iterasi 4	15470
Jumlah Iterasi 5	15170
Jumlah Iterasi 6	15220
Jumlah Iterasi 7	8920
Jumlah Iterasi 8	9970
Jumlah Iterasi 9	9100
Jumlah Iterasi 10	10150
Jumlah Iterasi 11	9030
Jumlah Iterasi 12	10080
Jumlah Iterasi 13	8970
Jumlah Iterasi 15	10020
Jumlah Iterasi 16	9260
Jumlah Iterasi 17	10310
Jumlah Iterasi 18	9440
Jumlah Iterasi 19	10490
Jumlah Iterasi 20	10010
Jumlah Iterasi 21	10010
Jumlah Iterasi 22	11060
Jumlah Iterasi 23	9920
Jumlah Iterasi 24	10970
Jumlah Iterasi 25	6950

Nama Iterasi	Jarak (meter)
Jumlah Iterasi 26	5400

Dari data diatas sekarang sudah dapat ditentukan jalur alternatif kendaraan dengan jarak terdekat adalah pada iterasi ke 26 yaitu dengan berawal dari kantor wali kota Depok lalu menuju Gerbang Utama GDC lalu melewati Jl. Mandor Samin setelah itu akan tiba di tempat tujuan puskesmas Cilodong.

Unsur kebaruan atau temuan dari sebuah penelitian ini adalah di uji dengan membangun sebuah aplikasi sederhana menghitung jarak dimana aplikasi sederhana tersebut di bangun dengan bahasa pemrograman HTML, CSS, Javascript, sebagai tool untuk pembuktian menghitung jarak dengan sistem sederhana, selain itu penggunaan algoritma *dijkstra* untuk dengan tingkat akurasi yang baik serta menghasilkan solusi rute terpendek yang cukup akurat membuat algoritma *dijkstra* lebih unggul digunakan untuk penyelesaian masalah lebih dari dua masalah jarak yang bersamaan.

Perlu penelitian lebih lanjut untuk mengetahui jarak terdekat dengan parameter waktu, biaya, moda transportasi dan kemacetan lalu lintas di jam-jam sibuk serta menggunakan algoritma lainnya sebagai pembanding hasil dari algoritma *dijkstra* untuk berbagai keperluan penyelesaian masalah jarak terdekat.

IV. PENUTUP

Dari hasil penelitian dengan menggunakan pemrograman sederhana yang menggunakan algoritma *dijkstra* disimpulkan bahwa Algoritma *dijkstra* dapat menentukan jarak terdekat dengan akurat, penelitian ini menguji lintasan yang dapat dilalui atau disebut iterasi sebanyak 26 lintasan yang dapat di lalui menuju puskesmas Cilodong dengan titik awal kantor wali kota Depok, penelitian ini menghitung jarak objek dari node mulai dan node tujuan yang sama. Hasil penelitian ini akan berkorelasi antara jarak dan waktu tempuh, jika jarak yang dilampaui dengan lintasan bebas hambatan maka dengan jarak terdekat bisa menghasilkan waktu tempuh yang cepat.

Hasil penelitian ini tetap dapat dikembangkan dengan membangun tool atau alat hitung jarak dengan aplikasi *android* atau *web base* yang *responsive* lainnya. Dengan menggunakan algoritma lainnya sebagai pembanding untuk algoritma *dijkstra*. Dengan menambahkan parameter waktu, biaya, moda

transportasi dan kemacetan lalu lintas di jam-jam sibuk.

DAFTAR PUSTAKA

- [1] S. Irmawati, "KUALITAS PELAYANAN KESEHATAN DI PUSKESMAS SANGURARA KECAMATAN TATANGA KOTA PALU," *J. Katalogis*, vol. 5, no. 1, pp. 188–197, 2017.
- [2] M. K. Harahap, "Pencarian Jalur Terpendek dengan Algoritma Dijkstra," *J. Penelit. Tek. Inform.*, vol. 2, pp. 18–23, 2019.
- [3] S. Kirono, M. I. Arifianto, R. E. Putra, A. Musoleh, and R. Setiadi, "Graph-based modeling and dijkstra algorithm for searching vehicle routes on highways," *Int. J. Mech. Eng. Technol.*, vol. 9, no. 8, pp. 1273–1280, 2018.
- [4] B. Amaliah, C. Fatichah, and O. Riptianingdyah, "Finding the shortest paths among cities in Java Island using node combination based on Dijkstra algorithm," *Int. J. Smart Sens. Intell. Syst.*, vol. 9, no. 4, pp. 2219–2236, 2016, doi: 10.21307/ijssis-2017-961.
- [5] E. Ismantohadi and I. Iryanto, "Penerapan Algoritma Dijkstra untuk Penentuan Jalur Terbaik Evakuasi Tsunami – Studi Kasus: Kelurahan Sanur Bali," *JTT (Jurnal Teknologi Terapan)*, vol. 4, no. 2. 2018, doi: 10.31884/jtt.v4i2.79.
- [6] S. Ardyan, A. Suyitno, and Mulyono, "Implementasi Algoritma Dijkstra Dalam Pencarian Rute Terpendek Tempat Wisata Di Kabupaten," *UNNES J. Math.*, vol. 6, no. 2, pp. 108–116, 2017.
- [7] Ferdiansyah and R. Ahmad, "Penerapan Kabupaten Tangerang," *J. TICOM*, vol. 2, no. 1, pp. 51–57, 2013.
- [8] S. Nandiroh and H. Munawir, "Implementasi Algoritma Dijkstra Sebagai Solusi Efektif Pembuatan Sistem Bantuan," *J. Ilm. Tek. Ind.*, vol. 12, no. 29, pp. 223–234, 2019. Algoritma Dijkstra untuk Menentukan Rute Terpendek Pembacaan Water Meter Induk PDAM Tirta Kerta Raharja
- [9] U. M. Rifanti, "Pemilihan Rute Terbaik Menggunakan Algoritma Dijkstra Untuk Mengurangi Kemacetan Lalu Lintas di

- Purwokerto,” *JMPM J. Mat. dan Pendidik. Mat.*, vol. 2, no. 2, p. 90, 2017, doi: 10.26594/jmpm.v2i2.926.
- [10] Fitria and Apri Triansyah, “Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Antar Kota Disumatra Bagian Selatan,” *Kntia*, vol. 5, no. 2, pp. 1–7, 2011, [Online]. Available: <http://www.scribd.com/doc/16342804/C-orel-Draw-12>.
- [11] E. Kusuma and H. Agung, “Aplikasi Perhitungan Dan Visualisasi Jarak Terpendek Berdasarkan Data Coordinate Dengan Algoritma Dijkstra Dalam Kasus Pengantaran Barang Di Kawasan Jabodetabek,” *J. Sisfokom*, vol. 08, no. 1, pp. 14–23, 2019.
- [12] C. S. Rahayu, “PENERAPAN ALGORITMA DIKSTRA DALAM PENENTUAN LINTASAN TERPENDEK MENUJU UPT. PUSKESMAS CILODONG KOTA DEPOK,” 2020.