

Implementasi Algoritma Dijkstra untuk Menentukan Lokasi dan Jarak Tempuh Terpendek Bank Syariah di Samarinda

Yusuf Arif Setiawan^{1*}, Windu Gata², Sidik³, Fariszal Nova Arviantino⁴

^{1,2,3,4} STMIK Nusa Mandiri

*14002635@nusamandiri.ac.id

Abstract

Islamic banks are the main preference for some adherents of the Islamic religion when transacting or saving their money. Although the concept of Islamic banking has been around for a long time in Indonesia, until now the existence of Islamic banks has not been widely known to the public. This is understandable because the number of Islamic banks can still be counted on the fingers. Another thing, the location of Islamic banks also rarely occupies the protocol road, making it difficult to find. Therefore, information on the closest route that can be passed to the Sharia Bank office is needed. To determine the closest route, the algorithm commonly used for finding the shortest route is Dijkstra's Algorithm. With information on the closest route from the market to the Sharia Bank office, when in an urgent situation, customers no longer have trouble finding the closest route to the Sharia Bank office.

Keywords: Dijkstra's Algorithm, Islamic Bank, Shortest Route

Abstrak

Bank Syariah menjadi prefrensi utama bagi sebagian penganut agama Islam saat bertransaksi atau menyimpan uangnya. Walaupun sudah lama konsep Bank Syariah ada di Indonesia, namun hingga saat ini keberadaan Bank Syariah belum banyak diketahui masyarakat. Hal ini dapat dipahami karena jumlah Bank Syariah masih dapat dihitung dengan jari. Hal lainnya, lokasi Bank Syariah juga jarang menempati jalan protokol, sehingga sulit ditemukan. Oleh karena itu dibutuhkan informasi rute terdekat yang dapat dilalui menuju kantor Bank Syariah. Untuk penentuan rute terdekat digunakan Algoritma yang biasa dipakai untuk pencarian rute terpendek yaitu Algoritma Dijkstra. Dengan adanya informasi rute terdekat dari pasar menuju kantor Bank Syariah, maka ketika dalam keadaan mendesak nasabah tidak lagi kesulitan mencari jalur yang paling dekat menuju kantor Bank Syariah.

Kata kunci: Algoritma Dijkstra, Bank Syariah, Rute Terpendek

1. Pendahuluan

Bank syariah adalah suatu system perbankan yang dikembangkan berdasarkan prinsip syariah. Adanya perbankan syariah di Indonesia merupakan suatu wujud penerapan ekonomi Islam yang tentu dibutuhkan oleh masyarakat muslim dalam hal keperluan jasa perbankan. Namun demikian, masih terdapat kendala yang menghambat mereka untuk berpaling kepada bank syariah ini. Beberapa alasan yang mendasari adalah sulitnya mencari lokasi kantor layanan Bank Syariah yang ada. Bila diperhatikan lokasi Bank Syariah jarang yang terletak di jalan protokol, atau dekat pusat keramaian sehingga ketika nasabah atau calon nasabah akan mendatangi maka kesulitan untuk menjangkau.

Dengan meningkatnya minat masyarakat untuk menabung di Bank Syariah, maka informasi keberadaan Bank Syariah hendaknya mudah didapat. Secara karakteristik kantor Bank biasanya terletak didekat pusat bisnis seperti pasar. Lokasi yang berhubungan dengan pusat bisnis akan memudahkan mobilisasi masyarakat dalam melakukan usahanya sekaligus bertransaksi diperbankan. Sayangnya keberadaan Bank Syariah dipusat keramaian kadang tidak terdeteksi karena terlalu padatnya lalu lintas, atau tersembunyi oleh rimbunnya parkir kendaraan para nasabahnya. Kantor Bank Syariah sendiri rata rata terletak di sebuah ruko, sehingga sudah dikenali. Maka untuk membantu para nasabah dan walk in costumer

menemukan lokasi kantor Bank Syariah tertentu, dibuat informasi yang dapat dimanfaatkan untuk mencari rute terdekat menuju kantor Bank Syariah tersebut.

Salah satu metode yang sering digunakan untuk mencari rute terdekat adalah dengan menggunakan Algoritma Dijkstra. Dengan algoritma ini dapat dicari jalur dengan jumlah bobot yang paling minimum antara titik yang satu ke titik lainnya[1]. Jarak dan rute terdekat yang dihasilkan dalam Algoritma Dijkstra menggunakan parameter tempat asal dan tempat tujuan[2]. Pada penelitian ini Algoritma Dijkstra digunakan untuk menentukan rute terpendek dari *pasar* sebagai parameter tempat asal ke Bank Syariah sebagai parameter tempat tujuan. Untuk proses perhitungan jarak dan rute terdekat, dilakukan melalui perhitungan jarak tiap persimpangan jalan yang dilalui oleh *pasar* dan Bank Syariah di Samarinda sehingga didapatkan jalur terpendek yang dapat dilalui.

Graf merupakan representasi dari gambaran hubungan yang melibatkan antara satu objek dengan objek yang lain, dimana objek diwakili oleh simpul (*node*), yang saling terhubung melalui garis yang disebut juga dengan sisi (*edge*)[3]. Contoh penggunaan graf seperti penggambaran struktur organisasi, arus lalu lintas, peta, sistem penjualan dan sebagainya. Berdasarkan orientasi arah pada sisi maka graf dapat dibedakan mejadi dua jenis yaitu graf berarah dan graf tidak berarah. Simpul pada graf berarah dihubungkan oleh sisi yang memiliki orientasi arah. Dimana graf berarah (u, v) berbeda dengan graf berarah (v, u). Sedangkan pada graf tidak berarah, sisi yang menghubungkan antar simpul tidak memiliki orientasi arah, sehingga urutan pasangan simpul dapat diabaikan [4].

Lintang dan bujur yang ditampilkan pada *Google Maps* merupakan dasar dari sistem koordinat geografis yang digunakan untuk menunjukan suatu titik di bumi [5]. Jarak antara dua titik kordinat digambarkan sebagai bobot sisi di dalam graf pada penelitian ini.

Masalah lintasan terpendek merupakan salah satu masalah kombinatorial yang paling mendasar dan terkenal yang sering muncul di

berbagai bidang ilmu seperti aplikasi jaringan jalan, transportasi dan aplikasi lainnya[6]. Menentukan lintasan terpendek berarti menentukan lintasan dengan jarak paling minimum yang dibutuhkan dari tempat asal untuk sampai ke tempat tujuan. Untuk menyelesaikan masalah pencarian lintasan atau rute terpendek diperkenalkan algoritma untuk pencarian rute terpendek salah satunya yaitu Algoritma Dijkstra.

Algoritma merupakan kumpulan perintah yang logis dan sistematis yang diterjemahkan secara bertahap dari awal hingga akhir sehingga dapat menyelesaikan suatu masalah[7]. Terdapat tiga cara dalam penulisan Algoritma yaitu:

1. Kalimat deskriptif, dengan cara menuliskan instruksi-instruksi yang harus dilaksanakan dalam bentuk untaian kalimat deskriptif.
2. *Flowchart* (diagram alir), penulisan algoritma dengan menggunakan notasi grafis, yang memperlihatkan urutan atau langkah-langkah dari suatu program dan hubungan antar proses beserta pernyataannya.
3. *Pseudocode*, yang berisi deskripsi dari algoritma pemrograman komputer yang menggunakan struktur sederhana dari beberapa bahasa pemrograman tetapi bahasa tersebut hanya ditujukan agar dapat dibaca manusia [8]

Penulisan algoritma harus mengikuti langkah-langkah yang benar agar mudah dibaca dan dipahami. Walaupun setiap orang mempunyai aturan dan penulisan algoritma masing-masing yang berbeda, sebaiknya tulisan algoritma yang dibuat mempunyai korelasi dengan tulisan bahasa pemrograman yang sudah umum digunakan.

Algoritma Dijkstra populer digunakan untuk memecahkan masalah yang berhubungan dengan pencarian lintasan terpendek. Metode pencarian lintasan terpendek menggunakan Algoritma Dijkstra dapat digunakan apabila jarak antara sebuah simpul ke simpul lain dalam graf memiliki bobot positif, jika terdapat bobot negatif maka penyelesaian yang diberikan adalah infiniti (tak terhingga) [9]. *Node* atau simpul pada Algoritma Dijkstra digunakan karena dalam

penentuan rute lintasan terpendek menggunakan graf berarah.

Langkah-langkah dasar dari Algoritma Dijkstra adalah:

1. Dimulai dari *node* yang kita pilih sebagai *node* sumber (*node* awal), kemudian menganalisa graf untuk menentukan jalur terpendek antara *node* awal dengan semua *node* lain yang ada di dalam graf.
2. Algoritma kemudian menghitung jarak terpendek dari setiap *node* yang diketahui ke *node* awal. *Node* yang telah mempunyai jarak terpendek kemudian diberi label "*visited*" (telah dikunjungi) dan ditambahkan ke jalur

Proses dilanjutkan sampai semua *node* dalam graf sudah ditambahkan ke jalur. Sehingga kita mendapatkan jalur yang menghubungkan *node* awal ke semua *node* lain mengikuti jalur terpendek yang mungkin untuk mencapai semua *node*[6].

2. Metoda Penelitian

Metode yang digunakan untuk mengumpulkan data dalam penelitian ini adalah:

Melalui observasi dengan melakukan pengumpulan data Bank Syariah dan pusat bisnis / pasar di wilayah Samarinda.

Melalui studi pustaka dengan melihat hasil penelitian dan karya penulisan, baik itu dari buku-buku, jurnal-jurnal, studi literatur maupun dari internet, dengan tema yang sejenis, dengan tujuan untuk menghindari kesalahan yang pernah terjadi di penelitian sebelumnya.

3. Hasil Penelitian

Dalam penelitian ini, yang dianggap sebagai *node* (simpul) adalah persimpangan jalan, pusat bisnis / pasar, dan Bank Syariah, sedangkan jarak antar persimpangan dianggap sebagai sisi (*edge*) yang memiliki bobot. Data Bank Syariah, pusat bisnis / pasar dan persimpangan jalan di wilayah kota Samarinda ditunjukkan dalam tabel berikut:

Tabel 1. Daftar Nama Bank Syariah di Kota Samarinda

Simpul	Keterangan
A	Bank Negara Indonesia Syariah

Simpul	Keterangan
B	Bank Syariah Mandiri
C	Bank Rakyat Indonesia Syariah
D	Bank Muamalat Indonesia

Tabel 2. Daftar Pusat Bisnis di Kota Samarinda

Simpul	Keterangan
a	Lembuswana Mall
b	Pasar Pagi
c	Pasar Segiri Samarinda
d	Plasa Mulia
e	Mal Mesra

Tabel 3. Daftar Nama dan Koordinat Persimpangan di Kota Samarinda

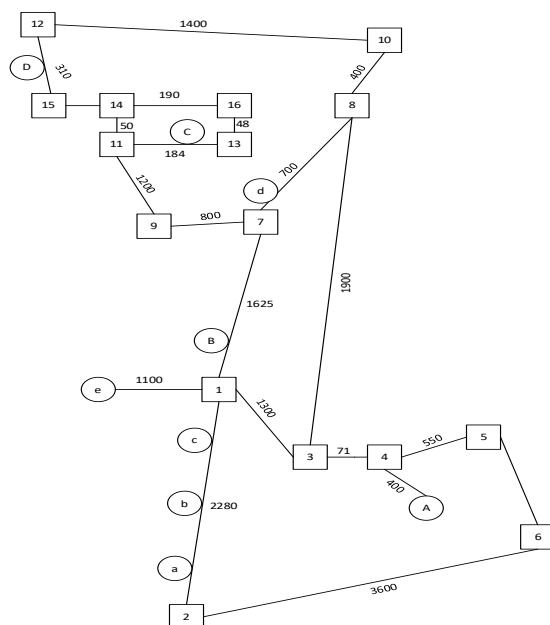
Simpul	Keterangan	Koordinat
1	Perempatan Lembusuana	-0.47549, 117.14629
2	Perempatan Juanda	-0.47255, 117.13854
3	Perempatan Suryanata	-0.48459, 117.12666
4	Pertigaan Cendana	-0.49814, 117.12677
5	Pertigaan RE Martadinata	-0.50021, 117.12735
6	Pertigaan BTN	-0.50084, 117.13517
7	Pertigaan Gunung Kinibalu	-0.50184, 117.14108
8	Perempatan Bank Kaltim	-0.50121, 117.14362
9	Pertigaan Mall Mesra	-0.50174, 117.14694
10	Pertigaan Jl. Abul Hasan	-0.49970, 117.14778
11	Perempatan Basuki Rahmat	-0.49577, 117.14970
12	Bundaran Taman Samarinda	-0.49678, 117.14395
13	Pertigaan Taman Makam Pahlawan	-0.48960, 117.14633
14	Pertigaan Agus Salim	-0.49307, 117.15145
15	Simpangan Jalan Gatot Subroto, Gang Mesjid	-0.48938, 117.15336
16	Perempatan Merak	-0.47925, 117.15475

Tabel 4. Data Jarak Antar Node

No.	Node	Jarak (m)	No.	Node	Jarak (m)
1	a-2	30	15	8-10	400
2	a-b	350	16	9-11	1200

No.	Node	Jarak (m)	No.	Node	Jarak (m)
3	b-c	1200	17	16-13	48
4	c-1	700	18	10-12	1400
5	e-1	1100	19	14-16	190
6	1-3	1300	20	14-15	140
7	2-6	3600	21	14-11	50
8	4-3	71	22	A-4	400
9	4-5	550	23	B-1	750
10	4-8	1900	24	B-d	850
11	5-6	1400	25	C-13	34
12	7-8	700	26	C-11	150
13	7-9	800	27	D-12	200
14	d-7	25	28	D-15	110

Berdasarkan tabel-tabel di atas dapat dibentuk graf seperti di bawah ini.



Gambar 1. Bank Syariah, Pusat Bisnis dan Persimpangan Jalan di Samarinda Kota

Sebagai *node* (simpul) awal adalah *node a* (Pusat Bisnis / Mall). Maka dengan menggunakan Algoritma Dijkstra dapat dihitung jarak terdekat *node a* ke masing-masing *node-node* lain di dalam graf. Simbol *infinity* dimaksudkan untuk representasi awal jarak *node a* ke *node* lain dalam graf[6].

Jarak yang terdekat dari *node a* ke *node* yang paling dekat yaitu *node 1* dan *2*. Diketahui *node 1,3,4,5, 6, dan 7* bersebelahan dengan *node* yang sudah ada di jalur karena

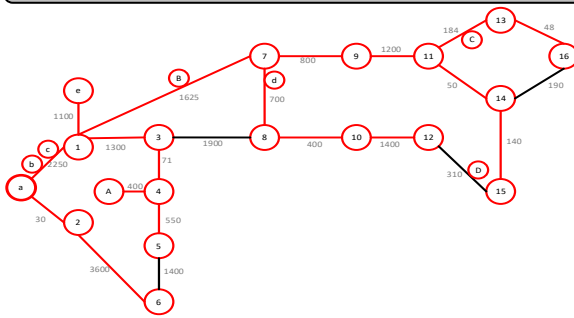
terhubung langsung dengan *node a* dan *2* secara berurutan.

Perhitungan jarak terdekat dari *node a* ke *node 6* melalui a-1-3-4-5-6 dengan total jarak 5566 m dan a-2-6 dengan total jarak 3630. Selanjutnya analisa *node* baru yang berdekatan dan bersebelahan dengan *node-node* yang sudah ada di jalur yaitu *node 8*. Diketahui jalur yang dapat dilalui dari *node a* ke *node 8* yaitu a-1-3-8 dengan total jarak 5450, dan a-1-7-8 dengan total jarak 4575. Analisa *node* baru yang berdekatan dan bersebelahan dengan *node-node* yang sudah ada di jalur yaitu *node 9* dan *10*. Jarak terdekat diperiksa dari *node a* ke *node-node* tersebut, ditambahkan pada jalur setelah dikunjungi. Selanjutnya dilakukan analisa *node* baru yang berdekatan dan bersebelahan dengan *node-node* yang sudah ada di jalur yaitu *node 11, 12 13,14,15, dan 16*. Jarak terdekat dari *node a* ke *node-node* yang sudah dikunjungi di tambahkan ke dalam jalur.

Tabel 5. Jarak Node Yang Sudah dikunjungi

No.	Jarak terdekat dari <i>node a</i>	Status	No.	Jarak terdekat dari <i>node a</i>	Status
a	0	visited	9	4675	visited
1	2250	visited	10	4975	visited
2	30	visited	11	5829	visited
3	3550	visited	12	6375	visited
4	3621	visited	12	6013	visited
5	4171	visited	13	5879	visited
6	3630	visited	14	6019	visited
7	3875	visited	15	6061	visited
8	4575	visited	16	6013	visited

Garis merah pada gambar menandai sisi yang termasuk dalam jalur terpendek. Garis merah sebagai jalur terpendek untuk mencapai *node* tertentu dalam grafik mulai dari *node a* seperti yang digambarkan di bawah:



Gambar 2. Rute Terpendek dari *Node a* ke *Node* Lain di Dalam Graf

Setelah dilakukan perhitungan maka diperoleh jalur terpendek dari pasar atau pusat bisnis ke lokasi bank syariah di kota Samarinda sebagai berikut:

Tabel 12. Rute dari Pasar/ Pusat Bisnis dengan Lokasi Bank Syariah di Kota Samarinda

Finish	A	B	C	D
a	a-1-3-4- A	a-1-B	a-1-7- 9-11-C	a-1-7-9-11- 14-15-D
b	b-1-3-4- A	b-1-B	b-1-7- 9-11-C	b-1-7-9-11- 14-15-D
c	c-1-3-4- A	c-1-B	c-1-7- 9-11-C	c-1-7-9-11- 14-15-D
d	d-8-3-4- A	d-7-B	d-7-9- 11-C	d-9-11-14- 15-D
e	e-1-3-4- A	e-1-B	e-1-7- 9-11-C	e-1-7-9-11- 14-15-D

4. Kesimpulan

Penelitian ini mengimplementasikan algoritma Djikstra untuk mendapatkan dan menentukan rute terpendek menuju Bank Syariah dari titik awal node pusat bisnis atau pasar di Samarinda. Pengguna dapat memilih Bank Syariah tujuan dari lokasi pusat bisnis, dengan rute yang terdekat. Hasil penghitungan jarak terpendek dalam pencarian lokasi Bank Syariah juga dipengaruhi oleh gadget yang digunakan user. Pada penelitian ini menggunakan google maps di computer.

5. Saran

Dalam penelitian ini merupakan langkah awal dalam penerapan algoritma Djikstra untuk menentukan lokasi dan jarak tempuh terpendek bank syariah di samarinda. Untuk penelitian selanjutnya akan ditambahkan

parameter lain yaitu jalan dua arah, agar hasil penelitian lebih dapat menggambarkan keadaan di lapangan yang sesungguhnya.

6. Daftar Pustaka

- [1] M. S. Yusuf, H. M. Az-zahra, and D. H. Apriyanti, "Implementasi Algoritma Dijkstra Dalam Menemukan Jarak Terdekat Dari Lokasi Pengguna Ke Tanaman Yang Di Tuju Berbasis Android (Studi Kasus di Kebun Raya Purwodadi)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 12, pp. 1779–1781, 2017.
- [2] S. Andayani and E. W. Perwitasari, "Penentuan Rute Terpendek Pengambilan Sampah di Kota Merauke Menggunakan Algoritma Dijkstra," *Aeminar Nas. Teknol. Inf. Komun. Terap.*, vol., no., pp. 164–170, 2014.
- [3] A. Dey, R. Pradhan, A. Pal, and T. Pal, "A genetic algorithm for solving fuzzy shortest path problems with interval type-2 fuzzy arc lengths," *Malaysian J. Comput. Sci.*, vol. 31, no. 4, pp. 255–270, 2018, doi: 10.22452/mjcs.vol31no4.2.
- [4] A. Noviriandini and M. Safitri, "Implementasi Algoritma Dijkstra Untuk Menentukan Jalur Terpendek Wilayah Pisangan Dan Kampus Nusa Mandiri Tangerang," *J. Pilar Nusa Mandiri*, vol. 13, no. 2, pp. 181–186, 2017.
- [5] Y. Yulianto, R. Ramadiani, and A. H. Kridalaksana, "Penerapan Formula Haversine Pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal," *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 13, no. 1, p. 14, 2018, doi: 10.30872/jim.v13i1.1027.
- [6] S. Broumi, A. Bakal, M. Talea, F. Smarandache, and L. Vladareanu, "Applying Dijkstra algorithm for solving neutrosophic shortest path problem," *Int. Conf. Adv. Mechatron. Syst. ICAMEchS*, vol. 0, pp. 412–416, 2016, doi: 10.1109/ICAMEchS.2016.7813483.
- [7] G. G. Maulana, "PEMBELAJARAN DASAR ALGORITMA DAN PEMROGRAMAN MENGGUNAKAN EL-GORITMA BERBASIS WEB," *J. Tek. Mesin*, vol. 06, no. 2, pp. 69–73, 2017.
- [8] A. M. Retta, A. Isroqmi, and T. D. Nopriyanti, "Pengaruh Penerapan

- Algoritma Terhadap Pembelajaran Pemrograman Komputer,” *Indiktika J. Inov. Pendidik. Mat.*, vol. 2, no. 2, p. 126, 2020, doi: 10.31851/indiktika.v2i2.4125.
- [9] M. Muharrom, “Algoritma, Implementasi Dalam, Dijkstra Jalur, Penentuan Kasus, Studi Tempat, Jarak Terdekat, Kuliah,” *Int. J. Bus. Intelegence*, vol. 3, no. 1, pp. 25–30, 2020.