



Penentuan rute terpendek antara dua titik di gudang menggunakan *Dijkstra's Algorithm* dan *Microsoft Excel*

Ekra Sanggala¹✉, Syafrial Fachri Pane², Roni Habibi²

Prodi D4 Logistik Bisnis, Sekolah Vokasi, Universitas Logistik Dan Bisnis Internasional, Bandung⁽¹⁾

Prodi D4 Teknik Informatika, Sekolah Vokasi, Universitas Logistik Dan Bisnis Internasional, Bandung⁽²⁾

DOI: [10.31004/jutin.v8i1.39060](https://doi.org/10.31004/jutin.v8i1.39060)

✉ Corresponding author:
[email: ekrasanggala@mail.ru]

Article Info	Abstrak
<p><i>Kata kunci:</i> <i>Shortest Path Problem;</i> <i>Dijkstra's Algorithm;</i> <i>Microsoft Excel;</i> <i>Gudang;</i> <i>Order Picking</i></p>	<p>Sebuah gudang merupakan suatu faktor penting dalam logistik dan mempunyai peran vital dalam mengontrol dan mengurangi biaya logistik. Secara umum operasional pada gudang terdiri dari lima fungsi dasar, yaitu: <i>receiving, sorting, storing, order picking</i> dan <i>delivering</i>. Kecepatan pada <i>order picking</i> merupakan faktor penting untuk kepuasan pelanggan. Maka mempersingkat waktu <i>order picking</i> merupakan hal yang penting. <i>Order Picking</i> yang paling sederhana adalah saat produk yang dibutuhkan pelanggan hanya terletak pada satu rak saja, sehingga <i>picker</i> hanya perlu bergera dari titik awal menuju ke titik rak dimana produk berada. Permasalahan penentuan rute terpendek antara dua titik dapat didefinisikan sebagai <i>Shortest Path Problem</i>. <i>Dijkstra's Algorithm</i> merupakan algoritma yang paling populer dalam menyelesaikan <i>Shortest Path Problem</i>. Untuk menyelesaikan <i>Shortest Path Problem</i> dengan <i>Dijkstra's Algorithm</i> diperlukan sebuah <i>tool</i> yang dapat membantu menyelesaikan perhitungannya. <i>Microsoft Excel</i> merupakan salah satu <i>tool</i> yang sangat populer dan mudah digunakan untuk menyelesaikan berbagai perhitungan. Dengan mengkombinasikan berbagai formula yang terdapat pada <i>Microsoft Excel</i> terbukti bahwa perhitungan <i>Dijkstra's Algorithm</i> untuk menyelesaikan <i>Shortest Path Problem</i> dapat dilakukan dengan baik.</p>
<p>Abstract</p> <p><i>A warehouse is an important factor in logistics and has a vital role in controlling and reducing logistics costs. In general warehouse operations consist of five basic functions, namely: receiving, sorting, storing, order picking and delivering. Speed in order picking is an important factor for customer satisfaction. So shortening order picking time is important. The simplest order picking is when the product needed by the customer is only located on one shelf, so the picker only needs to move from the starting point to the shelf point where the product is located. The</i></p>	

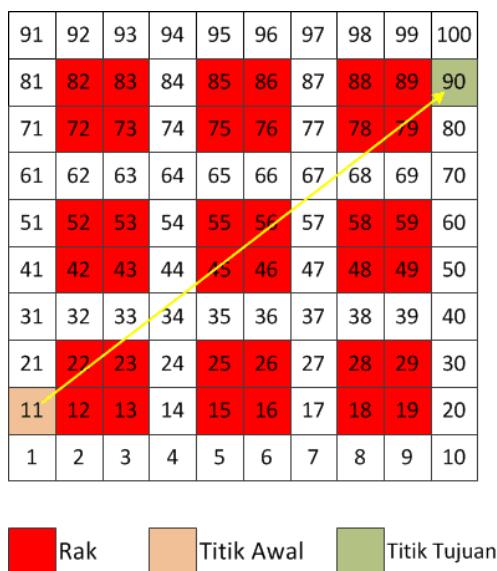
problem determining the shortest route between two points can be defined as the Shortest Path Problem. Dijkstra's Algorithm is the most popular algorithm in solving the Shortest Path Problem. To solve the Shortest Path Problem with Dijkstra's Algorithm, a tool is needed that can help complete the calculations. Microsoft Excel is one of the most popular and easy-to-use tools for completing various calculations. By combining various formulas in Microsoft Excel, it is proven that the Dijkstra's Algorithm calculation to solve the Shortest Path Problem can be done well.

1. INTRODUCTION

Sebuah gudang merupakan suatu faktor penting dalam logistik dan mempunyai peran vital dalam mengontrol dan mengurangi biaya logistik. Secara umum operasional pada gudang terdiri dari lima fungsi dasar, yaitu: *receiving, sorting, storing, order picking* dan *delivering*. Dari kelima fungsi dasar tersebut, *order picking* merupakan salah satu fungsi yang paling tinggi biayanya (Wisittipanich & Kasemset, 2015).

Kecepatan pada *order picking* merupakan faktor penting dalam *response time* terhadap *order* pelanggan dan sangat menentukan terhadap kepuasan pelanggan. Dengan demikian mempersingkat waktu *order picking* merupakan faktor yang penting dan bermanfaat dalam mengurangi biaya operasional gudang. Mempersingkat waktu *order picking* dapat dilakukan dengan mengoptimasikan rute *order picking* (Kordos et al., 2020).

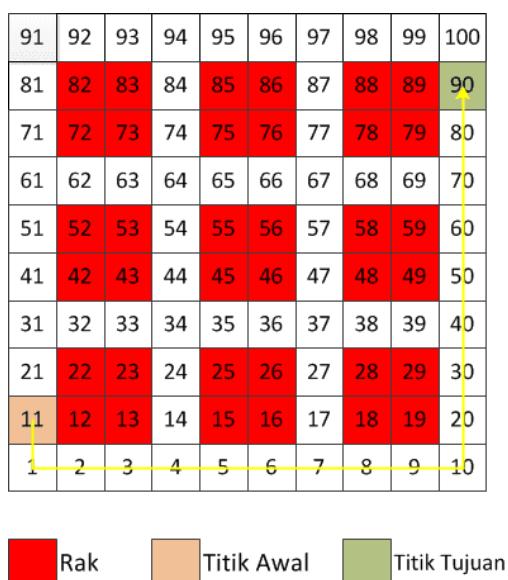
Order Picking yang paling sederhana adalah saat produk yang dibutuhkan pelanggan hanya terletak pada satu rak saja, dengan demikian *picker* hanya perlu bergerak dari titik awal *picker* menuju ke titik rak dimana produk berada. Walaupun *order picking* ini sederhana, bukan berarti penentuan rute pengambilan produknya menjadi sederhana juga. Adanya rak yang terdapat pada gudang mengakibatkan dari titik awal *picker* menuju ke titik rak tujuan tidak dapat langsung diambil garis lurus karena rak-rak yang ada menjadi penghalangnya. Ilustrasi mengenai hal ini dapat dilihat pada gambar 1.



Gambar 1: Ilustrasi Jarak Terdekat Antara Titik Awal Ke Lokasi Tujuan Dengan Garis Lurus

Pada gambar 1 dapat dilihat bahwa titik awal *picker* adalah titik 11 dengan titik tujuan adalah titik 90, jika diambil garis lurus langsung antara titik 11 ke titik 90 terlihat ada beberapa rak yang menghalangi garis tersebut, yaitu titik 22, titik 23, titik 45, titik 56, titik 78 dan titik 79.

Dikarenakan rute terpendek tidak dapat diperoleh langsung dengan cara mengambil garis lurus antara titik awal dengan titik tujuan tentunya perlu dicari rute yang mungkin untuk dilalui. Ilustrasi mengenai rute yang mungkin untuk dilalui dapat dilihat pada gambar 2.



Gambar 2: Ilustrasi Rute Yang Mungkin Untuk Dilalui

Pada gambar 2 dapat dilihat bahwa titik awal *picker* adalah titik 11 dengan titik tujuan adalah titik 90, dan dapat dilihat juga salah satu rute yang mungkin untuk dilalui dari titik 11 ke titik 90, yaitu titik 11 – titik 1 – titik 2 – titik 3 – titik 4 – titik 5 – titik 6 – titik 7 – titik 8 – titik 9 – titik 10 – titik 20 – titik 30 – titik 40 – titik 50 – titik 60 – titik 70 – titik 80 – titik 90.

Rute yang mungkin dilalui antara dua titik belum tentu merupakan rute terpendek, oleh karena itu perlu dicari rute terpendeknya agar dapat menghemat waktu dan menghemat biaya (Mombelli et al., 2022). Permasalahan penentuan rute terpendek antara dua titik dapat didefinisikan sebagai *Shortest Path Problem* (Dosantos et al., 2024). Terdapat beberapa algoritma yang dapat menyelesaikan *Shortest Path Problem*, yaitu (Ortega-Arranz et al., 2022):

1. *Dijkstra's Algorithm*
2. *Bellman-Ford Algorithm*
3. *A* Search Algorithm*
4. *Floyd-Warshall Algorithm*
5. *Johnson's Algorithm*
6. *Viterbi Algorithm*

Dijkstra's Algorithm merupakan algoritma yang paling populer dalam menyelesaikan *Shortest Path Problem* karena mudah untuk dipahami (Alam & Faruq, 2019). Untuk menyelesaikan *Shortest Path Problem* dengan *Dijkstra's Algorithm* diperlukan sebuah *tool* yang dapat membantu menyelesaikan perhitungannya (Jabbar et al., 2024). *Microsoft Excel* merupakan salah satu *tool* yang sangat populer dan mudah digunakan untuk menyelesaikan berbagai perhitungan (Sanggala, 2023).

Dengan adanya *Shortest Path Problem* pada gudang dan adanya *Dijkstra's Algorithm* untuk menyelesaikannya, serta adanya *Microsoft Excel* untuk menyelesaikan perhitungannya, maka pada *paper* ini akan dibahas mengenai penentuan rute terpendek antara dua titik di gudang menggunakan *Dijkstra's Algorithm* dan *Microsoft Excel*.

2. METHODS

Metode yang akan digunakan pada penelitian ini adalah *Dijkstra's Algorithm*. Langkah-langkah dari *Dijkstra's Algorithm* dalam menyelesaikan *Shortest Path Problem* adalah sebagai berikut ini (Cevizci, 2018):

1. Inisialisasi
 - Jadikan titik awal sebagai *current node*.
 - Hitung jarak *current node* ke titik-titik tetangganya.
 - Titik-titik yang tidak bertetangga dengan *current node*, untuk jaraknya berupa nilai yang sangat besar.

- Beri status seluruh titik sebagai "Non Permanen".
 - Buat sebuah himpunan dari seluruh titik "Non Permanen". Beri nama himpunan tersebut dengan "Himpunan Titik Non Permanen".
2. Kunjungi Titik-titik tetangga
- Untuk *current node*, pertimbangkan seluruh titik "Non Permanen" yang bertetangga dengannya dan hitung jarak tentatif antara *current node* dengan titik-titik tersebut.
 - Bandingkan jarak-jarak tentatif yang baru dihitung dengan jarak sekarang dan ganti jarak sekarang dengan jarak terkecil. Sebagai contoh, jika A berupa *current node* dan memiliki jarak 5, sedangkan jarak antara A dengan titik tetangganya B adalah 1, maka jarak menuju titik B melalui A adalah $5 + 1 = 6$. Jika titik B sebelumnya mempunyai jarak yang lebih besar dari 6 maka ganti jarak titik B dengan 6. Jika titik tetangga sudah berstatus "Permanen", abaikan titik tersebut..
3. Beri Status *current node* dengan "Permanen"
- Ketika seluruh titik tetangga dari *current node* sudah berstatus "Permanen", berilah status *current node* dengan "Permanen". Titik yang sudah berstatus "Permanen" tidak akan dicek lagi.
4. Pilih *current node* berikutnya
- Jadikan titik "Non Permanen" dengan jarak tentatif terkecil sebagai *current node*, lalu kembali ke langkah ke-2.
5. *Repeat*
- Ulangi langkah ke-2 sampai dengan langkah ke-4 hingga seluruh titik mendapatkan status "permanen". Maka rute terpendek (*Shortest Path*) telah diperoleh.

Pada penelitian ini, untuk menghitung jarak antara dua titik digunakan persamaan *Euclidean Distance*. Bentuk persamaan *Euclidean Distance* adalah sebagai berikut ini (Liberti & Lavor, 2017):

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

$d_{i,j}$: Jarak dari titik i ke titik j

x_i : absis titik i

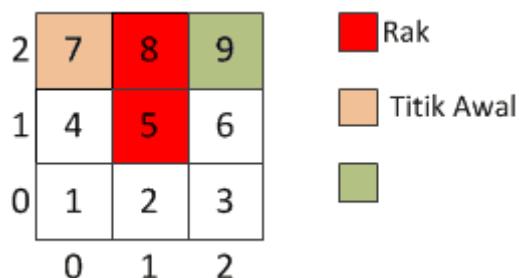
x_j : absis titik j

y_i : ordinat titik i

y_j : ordinat titik j

3. RESULT AND DISCUSSION

Untuk menjelaskan bagaimana *Shortest Path Problem* diselesaikan oleh *Dijkstra's Algorithm* dan *Microsoft Excel*, diperlukan sebuah contoh kasus. Dikarenakan keterbatasan tempat pada *paper* ini maka contoh kasus yang akan digunakan merupakan kasus yang sangat sederhana, sehingga langkah-langkah pada *Microsoft Excel* dapat termuat pada *paper* ini. Pada gambar 3 dapat dilihat contoh kasus yang sangat sederhana.



Gambar 3: Contoh Kasus Yang Sangat Sederhana

Pada gambar 3 terlihat bahwa gudang terdiri dari 9 titik dengan raknya terletak pada 2 titik saja, yaitu titik 5 dan titik 8. Untuk titik awal berada pada titik 7 dan untuk titik tujuan berada pada titik 9.

Berdasarkan gambar 3, maka absis dan ordinat setiap titik dapat diketahui seperti yang terlihat pada tabel 1.

Tabel 1: Absis Dan Ordinat Setiap Titik

Titik	X	Y
1	0	0
2	1	0
3	2	0
4	0	1
5	1	1
6	2	1
7	0	2
8	1	2
9	2	2

Untuk perhitungan menggunakan *Microsoft Excel*, digunakan berbagai formula antara lain *if*, *and*, *index*, *match*, *abs*, *sqrt*, *min*, *countif*, *iferror*, *offset*, *sum*, *small* dan *array formula*. Kombinasi dari formula-formula tersebut cukup rumit, sehingga sangat sulit untuk dijelaskan pada *paper* ini, oleh karena itu para pembaca yang membutuhkan *file excel*-nya dapat menghubungi penulis melalui surat elektronik.

Hasil perhitungan dengan menggunakan *Microsoft Excel* dapat dilihat pada tabel 2.

Tabel 2: Hasil Perhitungan Dengan Microsoft Excel

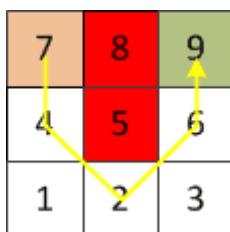
Iterasi	Current Node	Keterangan	1	2	3	4	5	6	8	9
1	7	Jarak	9999999	9999999	9999999	1	9999999	9999999	9999999	9999999
		Titik Sebelumnya	7	7	7	7	7	7	7	7
		Status	Non Permanen							
2	4	Jarak	2	2,414213562	9999999	1	9999999	9999999	9999999	9999999
		Titik Sebelumnya	4	4	7	7	7	7	7	7
		Status	Non Permanen	Non Permanen	Non Permanen	Permanen	Non Permanen	Non Permanen	Non Permanen	Non Permanen
3	1	Jarak	2	2,414213562	9999999	-	9999999	9999999	9999999	9999999
		Titik Sebelumnya	4	4	7	-	7	7	7	7
		Status	Permanen	Non Permanen	Non Permanen	-	Non Permanen	Non Permanen	Non Permanen	Non Permanen
4	2	Jarak	-	2,414213562	3,414213562	-	9999999	3,828427125	9999999	9999999
		Titik Sebelumnya	-	4	2	-	7	2	7	7
		Status	-	Permanen	Non Permanen	-	Non Permanen	Non Permanen	Non Permanen	Non Permanen
5	3	Jarak	-	-	3,414213562	-	9999999	3,828427125	9999999	9999999
		Titik Sebelumnya	-	-	2	-	7	2	7	7
		Status	-	-	Permanen	-	Non Permanen	Non Permanen	Non Permanen	Non Permanen
6	6	Jarak	-	-	-	-	9999999	3,828427125	9999999	4,828427125
		Titik Sebelumnya	-	-	-	-	7	2	7	6
		Status	-	-	-	-	Non Permanen	Permanen	Non Permanen	Non Permanen
7	9	Jarak	-	-	-	-	9999999	-	9999999	4,828427125
		Titik Sebelumnya	-	-	-	-	7	-	7	6
		Status	-	-	-	-	Non Permanen	-	Non Permanen	Permanen
8	5	Jarak	-	-	-	-	9999999	-	9999999	-
		Titik Sebelumnya	-	-	-	-	7	-	7	-
		Status	-	-	-	-	Permanen	-	Non Permanen	-
9	8	Jarak	-	-	-	-	-	-	9999999	-
		Titik Sebelumnya	-	-	-	-	-	-	7	-
		Status	-	-	-	-	-	-	Permanen	-

Dari hasil perhitungan dengan *Microsoft Excel* dapat diperoleh rute terpendeknya. Pada tabel 3 dapat dilihat rute terpendeknya.

Tabel 3: Rute Terpendek Yang Diperoleh

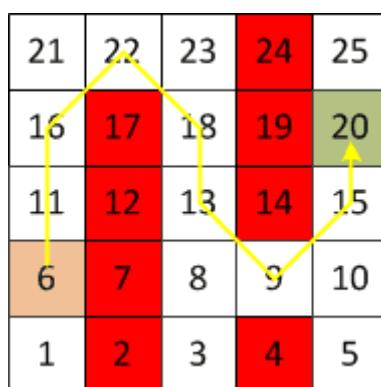
Rute	Jarak
7	
4	1
2	1,414214
6	1,414214
9	1
-	-
-	-
-	-
-	-
Total Jarak	4,828427

Dengan demikian rute terpendeknya adalah Titik 7 – titik 4 – titik 2 – titik 6 – titik 9 dengan jarak tempuh sejauh 4,828427. Untuk gambar dari rute terpendeknya dapat dilihat pada gambar 4.

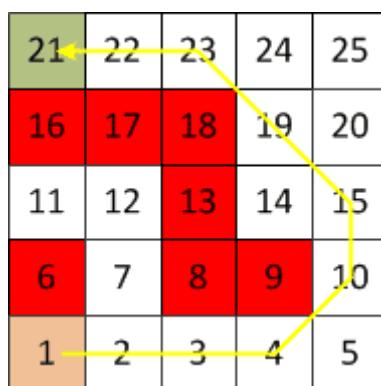


Gambar 4: Rute Terpendek Yang Diperoleh

Kasus yang lebih rumit pun dapat diselesaikan dengan baik. Pada gambar 5, gambar 6 dan gambar 7 dapat dilihat contoh kasus yang lebih rumit yang telah diselesaikan.



Gambar 5: Contoh Kasus Yang Lebih Rumit Yang Telah Diselesaikan (1)



Gambar 6: Contoh Kasus Yang Lebih Rumit Yang Telah Diselesaikan (2)

21	22	23	24	25
16	17	18	19	20
11	12	13	14	15
6	7	8	9	10
1	2	3	4	5

Gambar 7: Contoh Kasus Yang Lebih Rumit Yang Telah Diselesaikan (3)

4. CONCLUSION

Shortest Path Problem merupakan permasalahan yang sangat umum, sehingga *Shortest Path Problem* tidak hanya terjadi di gudang saja, tetapi juga bisa terdapat pada berbagai hal, antara lain sebagai berikut ini:

- *Transportation and Logistics* (Optimasi Rute dan Transportasi Publik).
- *Telecommunications (Network Routing)*.
- Sistem Navigasi (GPS dan Aplikasi Mapping).
- Pengembangan Game (AI Pathfinding).
- Robotik (Autonomous Vehicles).
- Social Networks (Friend Recommendations).
- Supply Chain Management (Distribusi).
- Urban Planning (Pengembangan Infrastruktur).
- Healthcare (Emergency Response).
- Keuangan (Analisis Jaringan).

Dalam gudang, *Shortest Path Problem* merupakan hal yang paling dasar untuk optimasi rute *order picking*. Barang-barang dari *order* pelanggan sebagian besar terletak pada berbagai rak, sehingga permasalahannya berkembang menjadi *Travelling Salesman Problem (TSP)* atau *Vehicle Routing Problem (VRP)*. Untuk menyelesaikan permasalahan *TSP* dan *VRP* ini diperlukan matrik jarak antar titik di gudang. Untuk hal ini *Shortest Path Problem* dapat berperan untuk menghitung matrik jarak antar titik di gudang.

Walaupun *Microsoft Excel* merupakan *tool* yang mudah digunakan, tetapi *Microsoft Excel* bukan merupakan *tool* yang *powerful* untuk melakukan perhitungan yang kompleks. Dengan demikian untuk perhitungan yang lebih kompleks dapat digunakan *Matlab*, *Octave* atau *Python*.

5. ACKNOWLEDGMENTS

Paper ini didukung oleh Direktorat Riset, PPM, Inovasi, Kewirausahaan & Kemitraan (RPIKK) Universitas Logistik Dan Bisnis Internasional. Melalui programnya yang bernama "Hibah Penelitian Internal" telah mendanai *paper* ini. Ucapan terima kasih diberikan kepada Direktorat RPIKK ULBI atas segala dukungannya.

6. REFERENCES

- Alam, M. A., & Faruq, M. O. (2019). Finding shortest path for road network using Dijkstra's algorithm. *Bangladesh Journal of Multidisciplinary Scientific Research*, 1(2), 41–45.
- Cevizci, B. (2018). Calculating the Shortest Path Using Dijkstra's Algorithm. *Journal of Inquiry Based Activities*, 8(2), 70–85.
- Dosantos, P. S., Bouchet, A., Mariñas-Collado, I., & Montes, S. (2024). Solving the Shortest Path Problem with Intervals as Costs Through Aggregation Functions. *Intelligent Management of Data and Information in Decision Making: Proceedings of the 16th FLINS Conference on Computational Intelligence in Decision and Control & the 19th ISKE Conference on Intelligence Systems and Knowledge Engineering (FLINS-ISKE 2024)*, 359–366.
- Jabbar, L. S., Abbas, E. I., & Hasan, S. D. (2024). Optimal dynamic path determination for a mobile robot using the

- Dijkstra algorithm based on updating weights in MATLAB. *AIP Conference Proceedings*, 3091(1).
- Kordos, M., Boryczko, J., Blachnik, M., & Golak, S. (2020). Optimization of warehouse operations with genetic algorithms. *Applied Sciences (Switzerland)*, 10(14). <https://doi.org/10.3390/app10144817>
- Liberti, L., & Lavor, C. (2017). *Euclidean distance geometry* (Vol. 3). Springer.
- Mombelli, A., Quilliot, A., & Baiou, M. (2022). Searching for a Safe Shortest Path in a Warehouse. *11th International Conference on Operations Research and Enterprise Systems*, 115–122.
- Ortega-Arranz, H., Gonzalez-Escribano, A., & Llanos, D. R. (2022). *The shortest-path problem: Analysis and comparison of methods*. Springer Nature.
- Sanggala, E. (2023). PENYELESAIAN CAPACITATED VEHICLE ROUTING PROBLEM (CVRP) DENGAN EVOLUTIONARY ALGORITHM & EXCEL SOLVER (STUDI KASUS: RUSSIA-20-NODES-CVRP INSTANCE). *PROFISIENSI: Jurnal Program Studi Teknik Industri*, 11(2), 144–151.
- Wisittipanich, W., & Kasemset, C. (2015). Metaheuristics for warehouse storage location assignment problems. *Journal of CMUJ NS Special Issue on Logistics and Supply Chain Systems*, 14(4).