# PENERAPAN ALGORITMA DIJKSTRA DALAM PENCARIAN RUTE TERPENDEK WISATA TAMAN AIR DI KOTA SURABAYA

# 

**Oleh :**

MUHAMMAD DZAKY ARIFIN (23051204120)

MUHAMMAD GUS SAIRI (23051204126)

ALIF RASYID FEBRIANSYAH (23051204131)

MU’IZUDDIN AHMAD GANI (23051204133)

**Dosen :** Dr. Yuni Yamasari, S.Kom., M.Kom.

**UNIVERSITAS NEGERI SURABAYA**

**FAKULTAS TEKNIK**

**JURUSAN TEKNIK INFORMATIKA**

**2023**

**KATA PENGANTAR**

Kami bersyukur kepada Allah SWT atas rahmat dan petunjuk-Nya yang memungkinkan kami menyelesaikan tugas berjudul "Penerapan Algoritma Dijkstra dalam Pencarian Rute Terpendek Wisata Taman Air di Kota Surabaya" ini tepat waktu.

Laporan ini ditulis dengan tujuan memenuhi persyaratan mata kuliah matematika diskrit, sekaligus untuk memberikan pemahaman tambahan tentang penerapan algoritma Dijkstra dalam kehidupan sehari-hari kepada pembaca dan penulis.

Kami ingin mengungkapkan rasa terima kasih kepada Ibu Dr. Yuni Yamasari, S.Kom., M.Kom., sebagai dosen matematika diskrit yang memberikan tugas ini, yang telah memberikan kesempatan bagi kami untuk memperluas pengetahuan dan wawasan dalam bidang studi yang kami geluti.

Tidak lupa, kami juga berterima kasih kepada semua pihak yang telah memberikan dukungan, meskipun tidak semua dapat disebutkan di sini, atas bantuan yang diberikan sehingga kami dapat menyelesaikan tugas ini.

Kami menyadari bahwa tulisan ini masih jauh dari sempurna. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun guna meningkatkan kesempurnaan laporan ini.

Surabaya, 5 Mei 2024

**Abstrak**

Wisata taman air atau *waterpark* merupakan salah satu bentuk wisata utama yang terdapat di Surabaya. Secara umum, wisata taman air di Surabaya adalah destinasi yang sering dikunjungi untuk kegiatan pemandian umum. Untuk mempermudah para pengunjung, menemukan rute terpendek ke destinasi wisata taman air tertentu menjadi hal yang sangat penting. Dalam proyek ini, kami akan membahas implementasi algoritma Dijkstra untuk mencari jalur terpendek menuju destinasi wisata taman air di Kota Surabaya menggunakan program *software* Tora. Hasil penelitian menunjukkan bahwa terdapat sembilan rute terpendek yang dimulai dari Terminal Bungurasih.

*Abstract*

*Waterpark tourism is one of the main forms of tourism in Surabaya. Generally, waterpark tourism in Surabaya is a destination often visited for public bathing activities. In order to facilitate tourists, it is important to find the shortest route to specific waterpark destinations. In this project, we will discuss the implementation of the Dijkstra algorithm to find the shortest path to waterpark destinations in the city of Surabaya using the Tora software program. The research results show that there are nine shortest routes starting from Terminal Bungurasih.*

**BAB I**

**PENDAHULUAN**

* 1. **LATAR BELAKANG**

Surabaya, sebagai salah satu kota terbesar di Indonesia, memiliki beragam objek wisata yang menarik perhatian baik dari wisatawan lokal maupun mancanegara. Salah satu jenis wisata yang cukup populer adalah wisata taman air atau *waterpark*. Objek wisata ini tidak hanya menjadi tempat rekreasi bagi warga setempat, tetapi juga menarik minat wisatawan yang berkunjung ke Surabaya untuk menikmati suasana menyegarkan dan keseruan di taman air.

Namun, meskipun wisata taman air memiliki daya tarik yang tinggi, seringkali wisatawan mengalami kesulitan dalam menemukan jalur tercepat menuju destinasi wisata tersebut. Hal ini dapat menjadi kendala yang mengurangi pengalaman wisata yang optimal, terutama bagi wisatawan yang tidak familiar dengan wilayah Surabaya.

Dalam upaya untuk meningkatkan pengalaman wisata para pengunjung, serta memudahkan aksesibilitas menuju objek wisata taman air di Surabaya, penerapan teknologi dan metode pencarian rute terpendek menjadi suatu kebutuhan. Salah satu metode yang efektif dan efisien adalah dengan menggunakan algoritma Dijkstra. Algoritma ini telah terbukti efektif dalam menemukan jalur terpendek pada graf berbobot, seperti jaringan jalan di kota.

Dengan menerapkan algoritma Dijkstra, dapat dihasilkan solusi yang memberikan informasi mengenai jalur-jalur terpendek dengan titik awal di Terminal Bungurasih menuju objek wisata taman air tertentu. Hal ini akan sangat bermanfaat bagi wisatawan, baik lokal maupun mancanegara, dalam merencanakan perjalanan mereka serta memaksimalkan waktu yang dimiliki untuk menikmati keindahan taman air di kota Surabaya.

Oleh karena itu, penelitian ini bertujuan untuk menjelajahi penerapan algoritma Dijkstra dalam pencarian rute terpendek menuju destinasi wisata taman air di kota Surabaya. Dengan demikian, diharapkan dapat memberikan kontribusi positif dalam meningkatkan kualitas layanan wisata dan memperkuat daya tarik pariwisata Surabaya sebagai destinasi unggulan di Indonesia.

* 1. **RUMUSAN MASALAH**

Rumusan masalah bedasarkan latar belakang dapat diurai sebagai berikut:

1. Bagaimana cara algoritma Dijkstra diterapkan dalam mencari jalur terpendek ke destinasi wisata taman air di kota Surabaya menggunakan program Tora?
2. Bagaimana identifikasi rute terpendek dari titik awal Terminal Bungurasih menuju destinasi wisata taman air tertentu menggunakan algoritma Dijkstra?
3. Bagaimana efektivitas dan efisiensi penerapan algoritma Dijkstra dalam meningkatkan aksesibilitas dan mengoptimalkan pengalaman wisatawan dalam mengunjungi objek wisata taman air di Surabaya?
   1. **TUJUAN DAN MANFAAT**

Ada juga tujuan dan manfaat dari laporan ini sebagai berikut :

1. Mencari rute terpendek menggunakan algoritma Dijkstra menuju objek wisata taman air di kota Surabaya.
2. Memahami cara yang dilakukan untuk mengidentifikasi rute terpendek dari titik awal Terminal Bungurasih menuju destinasi wisata taman air tertentu.
3. Memahami efektivitas dan efisiensi penerapan algoritma Dijkstra dalam meningkatkan aksesibilitas dan mengoptimalkan pengalaman wisatawan dalam mengunjungi objek wisata taman air di Surabaya

**BAB II**

**TINJAUAN PUSTAKA**

**2.1 TINJAUAN PUSTAKA**

Yusuf, Az-zahra, & Apriyanti, (2017), Edsger Dijkstra menciptakan sebuah algoritma untuk menentukan jalur terpendek antara dua titik dalam sebuah grafik berbobot dengan bobot positif.. Fitria & Triansyah, (2013), Secara konseptual, algoritma ini mirip dengan algoritma Prim karena keduanya membangun keluaran pohon *T* dari titik awal dengan ditambahkan simpul dan tepi. Namun, perbedaannya terletak pada metode untuk pemilihan simpul selanjutnya yang akan ditambahkan. Setiap simpul dapat dipastikan oleh Dijkstra untuk ditambahkan ke pohon T memiliki jalur terpendek yang diidentifikasi dari titik awal ke simpul tersebut.

Wang, (2012), menyatakan bahwa pada tahap awal penerapan algoritma, setiap simpul *u* dalam graf *G* diberi label *L(u),* yang mencerminkan perkiraan terbaik saat itu dari panjang jalur terpendek dari titik awal *a* ke simpul *u*. Ardyan, Suyitno, & Mulyono, (2017), Label *L(a)* pada awalnya disetel ke 0 karena jarak dari *a* ke *a* merupakan jalur terpendek dan memiliki panjang nol. Namun, karena tidak ada informasi sebelumnya tentang panjang jalur terpendek dari a ke simpul lain di dalam graf *G*, label *L(u)* untuk setiap simpul *u* selain *a* awalnya diatur sebagai angka tak terhingga, ditandai dengan *∞,* yang melebihi jumlah bobot semua tepi di dalam *G*. Ketika algoritma dieksekusi, nilai *L(u)* diperbarui secara bertahap hingga akhirnya merefleksikan panjang jalur terpendek yang sebenarnya dari *a* ke *u* di dalam graf *G*.

Retnati, Istiadi, & Roqib, (2015); Sholihah, (2015); Harapah & Khairina, (2017), Mirip dengan algoritma Kruskal dan Prim yang digunakan untuk mencari pohon rentang minimum, terdapat suatu pendekatan yang sederhana tetapi kurang efisien dalam menemukan jalur terpendek dari titik awal *a* ke titik tujuan *z*: yaitu dengan menghitung panjang semua jalur yang mungkin dan memilih yang paling pendek. Namun, pendekatan ini memiliki kelemahan yang signifikan karena bahkan untuk grafik yang relatif kecil, proses pencarian jalur terpendek dapat memakan waktu berabad-abad. Berbeda dengan hal itu, algoritma Dijkstra mampu menyelesaikan tugas ini dalam hitungan detik.

**BAB III**

**METODE PENELITIAN**

**3.1 TEKNIK PENGUMPULAN DATA**

Dalam penelitian ini, data yang digunakan adalah data sekunder yang diperoleh dari *Google Maps*, yang mencakup jarak antara berbagai objek wisata di Kota Surabaya, termasuk Kolam Renang Primavera, Bukit Mas Kolam Renang, Kolam Renang UNESA, Swimming Pool Water Palace,Kolam Renang G-Walk, Kolam Renang Tanjung Sari, Kolam Renang Monkasel, Kolam Renang Koni, Water Fun Plaza Marina*.*

**3.2 TEKNIK ANALISIS DATA**

Langkah-langkah berikut akan diterapkan untuk mengatasi permasalahan yang dihadapi.

1. Mengidentifikasi permasalahan yang ada di Kota Surabaya melibatkan pertanyaan tentang bagaimana menemukan rute terpendek yang bermula dari Terminal Bungurasih. Ini dilakukan dengan menganggap setiap objek wisata sebagai titik dan jalur yang menghubungkannya sebagai lintasan yang akan ditempuh.
2. Membangun model matematis untuk permasalahan jarak terpendek yang harus ditempuh melibatkan penggunaan matematika diskrit serta penerapan Algoritma Dijkstra.
3. Proses pembuatan rute terpendek antara objek wisata menggunakan Algoritma Dijkstra dieksekusi baik secara manual maupun dengan menggunakan *sofrware* Tora, dengan titik awal yang ditetapkan di Terminal Bungurasih.
4. Hasil penelitian diimplementasikan dengan menetapkan rute terpendek dari Terminal Bungurasih ke sembilan objek wisata di Kota Surabaya.

**3.3 ALGORITMA DIJKSTRA**

Notasi dan terminologi dalam konteks ini mengikuti pedoman dari penelitian sebelumnya (EPP, 2011; Prajapati, Singhal, Ranjan, & Chourasia, 2017).

Data yang digunakan meliputi *G*, sebuah graf sederhana dan memiliki keterhubungan denhan bobot yang positif pada setiap sisi; *∞,* angka yang lebih besar dari total bobot semua sisi dalam graf; *w(u,v),* yang menunjukkan bobot sisi antara simpul *u* dan *v; a*, sebagai titik awal; dan *z*, sebagai titik akhir.

Algoritma:

1. Tahap awal, *T* diinisialisasi sebagai sebuah graf yang hanya memiliki simpul *a* tanpa adanya sisi. Didefinisikan bahwa *V(T)* adalah himpunan dari semua titik dalam *T*, sementara *E(T)* adalah himpunan dari semua sisi dalam *T*.
2. Langkah selanjutnya adalah menetapkan nilai awal *L(a)* sebagai 0, sedangkan untuk setiap titik *u* dalam graf *G* kecuali *a*, nilai *L(u)* diatur sebagai ∞.
3. Proses selanjutnya adalah menginisialisasi simpul *v* dengan nilai *a*, dan himpunan *F* diset menjadi *{a}.*
4. Selama masih terdapat titik yang belum diproses dalam *T:*
5. Himpunan *F* diperbarui dengan menghapus simpul *v* dari *F* dan menambahkan semua simpul yang berdekatan dengan *v* dan tidak termasuk dalam himpunan *V(T).*
6. Untuk tiap simpul *u* yang bertetangga dengan *v* dan tidak termasuk dalam himpunan *V(T),* jika nilai *L(v)* ditambah dengan bobot *w(v, u)* lebih kecil dari nilai *L(u),* maka nilai *L(u)* diupdate menjadi *L(v)* ditambah *w(v, u).* Selain itu, nilai *D(u)* diatur sebagai *v*.
7. Cari simpul *a* dalam himpunan *F* yang memiliki label terendah. Kemudian, tambahkan simpul *x* ke dalam himpunan *V(T),* dan sisi *{D(x), x}* ditambahkan ke dalam himpunan *E(T)*. Selanjutnya, simpul *v* diatur menjadi *x*.

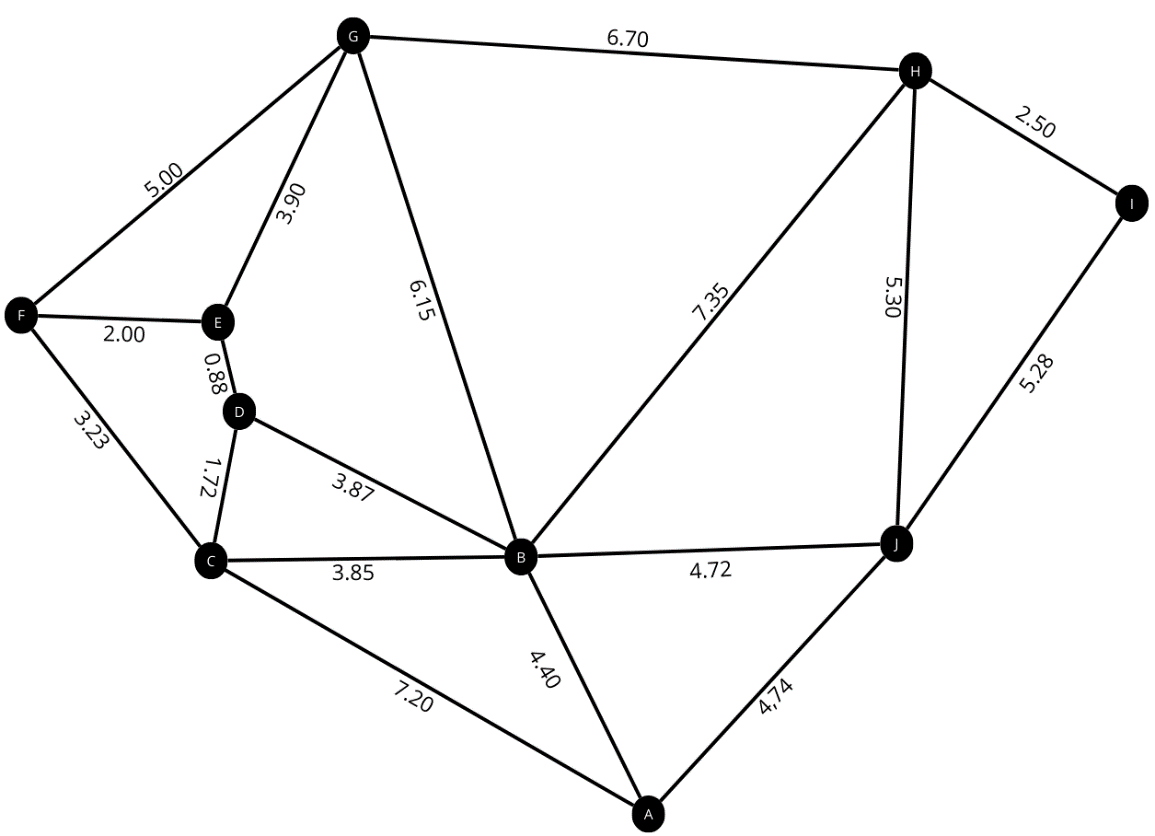
Berakhiran sementara

*keluaran: L(z)*

**BAB IV**

**PEMBAHASAN**

**4.1 HASIL DAN DISKUSI**

****Dalam penelitian ini, lokasi wisata dianggap sebagai simpul, sedangkan jarak antara lokasi wisata dianggap sebagai sisi dengan bobot yang sesuai. Informasi yang ditemukan dari penelitian ini, seperti nama lokasi wisata, disajikan dalam Gambar 1 dan Tabel 1. Data yang digunakan dalam penelitian ini berasal dari *Google Maps*.

Gambar 1. Wisata taman air di Kota Surabaya

Tabel 1. Kode huruf yang menandai lokasi wisata taman air di Kota Surabaya.

|  |  |
| --- | --- |
| Huruf | Keterangan |
|  |  |
| **A** | Terminal Bungurasih |
| **B** | Kolam Renang Primavera |
| **C** | Wisata Bukit Mas Kolam Renang |
| **D** | Kolam Renang UNESA |
| **E** | Swimming Pool Water Palace |
| **F** | Kolam Renang G-Walk |
| **G** | Kolam Renang Tanjung Sari |
| **H** | Kolam Renang Monkasel |
| **I** | Kolam Renang Koni |
| **J** | Water Fun Plaza Marina |
| **A** | Titik Awal |

Data hasil penelitian mengenai jarak antara lokasi wisata di Kota Surabaya dapat ditemukan dalam Tabel 2.

Tabel 2. Rute terpendek untuk tempat-tempat wisata taman air di Surabaya direpresentasikan dengan menggunakan titik dan garis..

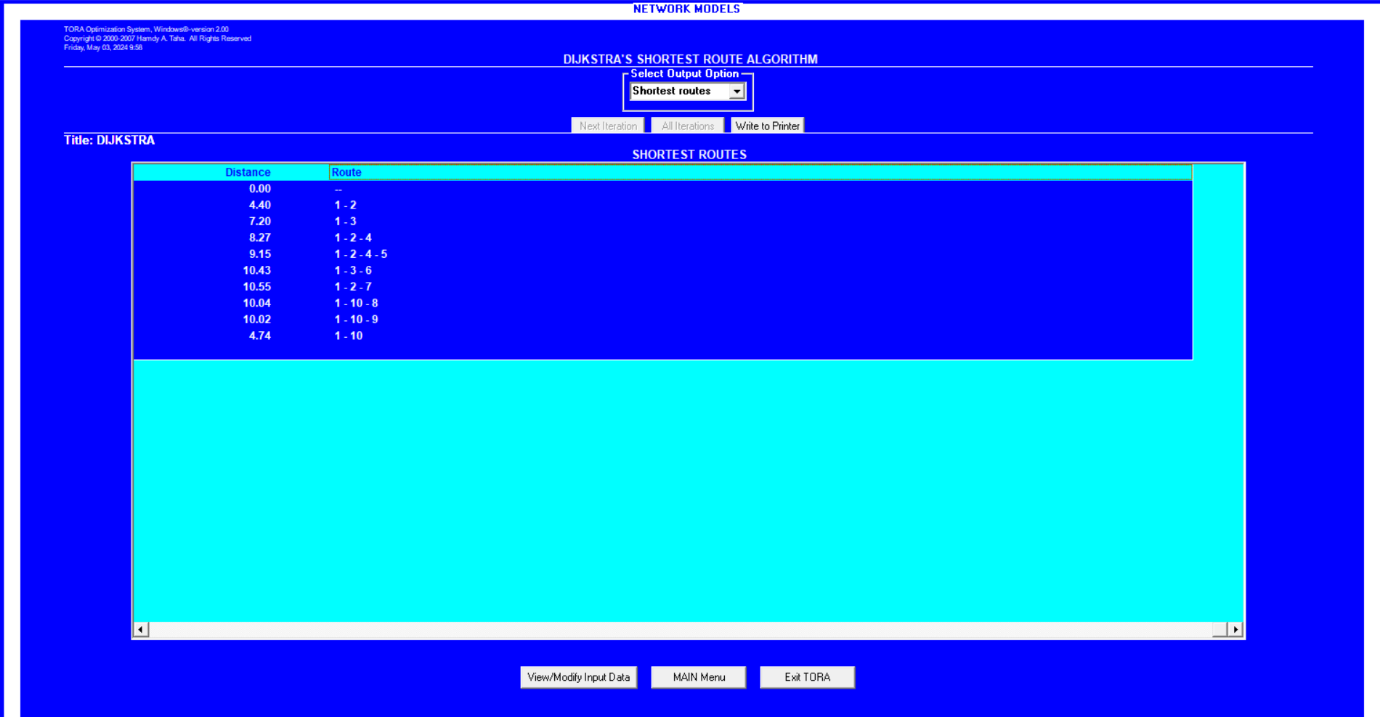
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Gambar | Tujuan | Rute | Jarak |
|  |  |  |  |  |
| **1.** | root | **A** | **\_ \_** | **0,00 KM** |
| **2.** |  | **B** | **A – B** | **4,40 KM** |
| **3.** |  | **C** | **A – C** | **7.20 KM** |
| **4.** |  | **D** | **A – B – D** | **8.27 KM** |
| **5.** |  | **E** | **A – B – D – E** | **9.15 KM** |
|  |  |  |  |  |
| **6.** |  | **F** | **A – C – F** | **10.43 KM** |
| **7.** |  | **G** | **A – B – G** | **10.55 KM** |
| **8.** |  | **H** | **A – J – H** | **10.04 KM** |
| **9.** |  | **I** | **A – J – I** | **10.02 KM** |
| **10.** |  | **J** | **A – J** | **4.74 KM** |
|  |  |  |  |  |

Data dalam Tabel 2 menunjukkan rute terpendek dari titik awal ke semua kombinasi titik. Di sisi lain, Tabel 3 menampilkan hasil perhitungan manual rute terpendek menggunakan Algoritma Dijkstra dengan Terminal Bungurasih sebagai titik awal.

Tabel 3. Rute terpendek tempat wisata taman air di Surabaya dengan titik awal Terminal Bungurasih.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Titik Awal | Tujuan | Rute | Jarak |
|  |  |  |  |  |
| 1. | Terminal Bungurasih | Kolam Renang Primavera | Terminal Bungurasih – Kolam Renang Primavera | 4.40 KM |
| 2. | Terminal Bungurasih | Wisata Bukit Mas Kolam Renang | Terminal Bungurasih – Wisata Bukit Mas Kolam Renang | 7.20 KM |
| 3. | Terminal Bungurasih | Kolam Renang UNESA | Terminal Bungurasih – Kolam Renang Primavera – Kolam Renang UNESA | 8.27 KM |
| 4. | Terminal Bungurasih | Swimming Pool Water Palace | Terminal Bungurasih – Kolam Renang Primavera – Kolam Renang UNESA – Swimming Pool Water Palace | 9.15 KM |
| 5. | Terminal Bungurasih | Kolam Renang G-Walk | Terminal Bungurasih – Wisata Bukit Mas Kolam Renang – Kolam Renang G-Walk | 10.43 KM |
| 6. | Terminal Bungurasih | Kolam Renang Tanjung Sari | Terminal Bungurasih – Kolam Renang Primavera – Kolam Renang Tanjung Sari | 10.55 KM |
| 7. | Terminal Bungurasih | Kolam Renang Monkasel | Terminal Bungurasih – Water Fun Plaza Marina – Kolam Renang Monsakel | 10.04 KM |
| 8. | Terminal Bungurasih | Kolam Renang Koni | Terminal Bungurasih – Water Fun Plaza Marina – Kolam Renang Koni | 10.02 KM |
| 9. | Terminal Bungurasih | Water Fun Plaza Marina | Terminal Bungurasih – Water Fun Plaza Marina | 4.74 KM |
|  |  |  |  |  |

Untuk memvalidasi hasil perhitungan secara manual, digunakan *software* Tora yang telah dilengkapi dengan paket Algoritma Dijkstra. *Output* dari *software* Tora yang menampilkan rute terpendek antara tempat-tempat wisata taman air di Kota Surabaya dengan Terminal Bungurasih sebagai titik awal dapat dilihat pada Gambar 2.



Gambar 2. *Output* *software* Tora rute terpendek tempat wisata taman air yang ada di Surabaya dengan titik awal Terminal Bungurasih

Hasil akhir yang diperoleh dari *software* Tora seperti yang terlihat pada Gambar 2 mencerminkan hasil yang sama dengan perhitungan manual menggunakan algoritma Dijkstra, sebagaimana ditunjukkan dalam Tabel 3. Hal ini menunjukkan konsistensi hasil antara perhitungan manual dan menggunakan *software* Tora. Oleh karena itu, algoritma Dijkstra terbukti efektif dalam menemukan rute terpendek ke tempat-tempat wisata taman air di Kota Surabaya.

Hasil penelitian menemukan bahwa terdapat sembilan rute terpendek dari Terminal Bungurasih ke setiap lokasi objek wisata taman air. Dalam rute-rute tersebut, terlihat bahwa beberapa rute terpendek tidak berbeda, memungkinkan wisatawan untuk mencapai satu tujuan melalui beberapa objek wisata lainnya. Hal ini menunjukkan potensi untuk mengembangkan paket wisata yang mencakup beberapa objek wisata dalam satu rute. Sebagai contoh, paket wisata yang mencakup kunjungan ke Wisata Kolam Renang Primavera, Kolam Renang UNESA, dan Swimming Pool Water Palace dapat disusun dari Terminal Bungurasih dalam satu hari.

**BAB V**

**PENUTUP**

**5.1 SIMPULAN**

Penelitian ini menghasilkan temuan yang signifikan terkait penggunaan Algoritma Dijkstra dalam menemukan rute terpendek menuju tempat-tempat wisata taman air di Kota Surabaya. Melalui analisis manual dan penerapan *software* Tora, ditemukan bahwa algoritma ini memberikan hasil yang konsisten dan dapat diandalkan. Oleh karena itu, algoritma Dijkstra terbukti efektif dan efisien dalam menemukan rute terpendek ke tempat-tempat wisata taman air di Kota Surabaya. Kesesuaian hasil antara perhitungan manual dan penggunaan *software* menunjukkan keandalan Algoritma Dijkstra dalam situasi praktis. Temuan ini memberikan kontribusi penting dalam pengembangan rute wisata yang efisien dan terencana, serta memberikan dasar yang kuat untuk penggunaan teknologi dalam industri pariwisata. Oleh karena itu, implementasi Algoritma Dijkstra dalam pencarian rute wisata dapat menjadi landasan yang berharga bagi pemangku kepentingan dalam industri pariwisata untuk meningkatkan pengalaman wisatawan dan efisiensi operasional.

**DAFTAR PUSTAKA**

N. A., Sudibyo, P. E., Setyawan, Y. P. S. R., Hidayat. (2020). Implementasi Algoritma Dijkstra Dalam Pencarian Rute Terpendek Tempat Wisata Di Kabupaten Klaten.

M., Masri, A. P., Kiswanto, B. S., Kusuma. (2019). Implementasi Algoritma Dijkstra Dalam Perancangan Aplikasi Penentuan Rute Terpendek Pada Objek Pariwisata Danau Toba dan Sekitarnya.

Yusuf, M. S., Az-zahra, H. M., & Apriyanti, D. H. (2017). Implementasi Algoritma Dijkstra. Dalam Menemukan Jarak Terdekat Dari Implementasi Algoritma Dijkstra Dalam Menemukan Jarak Terdekat Dari Lokasi Pengguna Ke Tanaman Yang Di Tuju Berbasis Android (Studi Kasus di Kebun Raya Purwodadi).

Fitria, Triansyah, A. (2013). Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan.

Retnani, W. E. Y., Istiadi, D., Roqib, A. (2015). Pencarian SPBU Terdekat dan Penentuan Jarak Terpendek Menggunakan Algoritma DIJKSTRA (Studi Kasus di Kabupaten Jember).

Ardyan, S., Suyitno, A., & Mulyono. (2017). Implementasi Algoritma Dijkstra Dalam Pencarian Rute Terpendek Tempat Wisata Di Kabupaten.

K. A., Santoso, Z., Afifah, I., Halikin (2023). Penerapan Algoritma Dijkstra dalam Pencarian Rute Terpendek Wisata Religi.

Wang, S. X. (2012). The improved Dijkstra’s shortest path algorithm and its application.

EPP, S. S. (2011). *Discrete Mathematics With Applications* (*fourth*). Canada: Richard Stratton.