Optimasi Algoritma Turun Penggunungan Swiss dengan BFS dan BaB

http://dx.doi.org/10.28932/jutisi.vXiX.X

Riwayat Artikel

*Received*: 31 Mei 2024 | *Final Revision*: 31 Mei 2024 | *Accepted*: 31 Mei 2024

Creative Commons License 4.0 (CC BY – NC) 

Robert Antonius🖂 \*1, Yohannes\*2

\* Informatika, Universitas Multi Data Palembang  
Jl. Rajawali No.14, Palembang, 30113, Indonesia

1robert.antonius@mhs.mdp.ac.id

3yohannesmasterous@mdp.ac.id

🖂Corresponding author: robert.antonius23@gmail.com

Abstrak — Swiss adalah sebuah negara dengan sektor turisme pegunungan yang kuat. Dengan banyaknya turis yang datang ke Swiss untuk mendaki, banyak pula keperluan untuk menjaga keamanan pendaki turis Swiss, seperti dengan peta yang baik. Peta-peta umumnya telah memiliki rute yang telah diinspeksi dan telah terjamin aman, namun rute ini tidak selalu tersedia. Untuk itu diperlukan suatu cara untuk memberi pendaki rute yang baik, terutama saat pendaki mencoba turun dari gunung. Algoritma BFS dapat terlalu lama dalam mencari rute yang optimal, sedangkan BaB belum tentu menemukan rute yang optimal. Pergabungan kedua algoritma ini membuat algoritma *informed search* yang mengambil bobot dari konsep BFS dan BaB, sehingga kedua algoritma dapat saling menutupi kelemahan satu sama lain. Penelitian ini menemukan bahwa dengan rasio bobot BFS dan bobot BaB 1:1, didapat algoritma yang dapat menemukan rute optimal dalam lebih sedikit langkah dari BFS dan dengan konsistensi lebih baik dari BaB.

Kata kunci— BFS, Branch and Bound, rute turun gunung Swiss.

Swiss Mountain Descent Algorithm Optimization with BFS and BaB

Abstract — Swiss is a nation with a strong mountain tourism industry. With how many tourists came to Swiss for hiking, there’s also a lot of emphasis put on ensuring the safety of those tourists, like by providing them with a good map. Maps usually have routes that have been charted and inspected to ensure their safety, but these routes might not always be available. For that, there’s a need for a way to provide good routes for the hikers on the fly, especially when they try to hike down the mountain. The BFS algorithm could take too long to search for an optimal route, while BaB might give suboptimal routes. The combination of these algorithm makes an informed search algorithm that takes weights from both BFS’s and BaB’s concept, so that the two algorithm could cover each other’s weaknesses. This research found that with a ratio of BFS weight and BaB weight of 1:1, an algorithm that could find the optimal route in less steps than BFS and with more consistency than BaB is exists.

Keywords— BFS, Branch and Bound, Swiss mountain descent route.

1. Latar Belakang

Swiss, atau formalnya Konfederasi Swiss, adalah salah satu negara dengan jumlah area pegunungan yang paling banyak di dunia [1]. Swiss juga adalah sebuah destinasi turisme paling kuat terkait liburan di pegunungan [2]. Swiss juga beriklim sedang dan bahkan dingin di beberapa area pegunungannya, sehingga menjadi tempat wisata ski yang sangat populer [3].

Berkat banyaknya area pegunungan terbuka di Swiss, ada pula keperluan tinggi untuk menjamin pengunjung mendaki tersebut supaya dapat turun dari gunung dengan selamat [4]. Swiss adalah salah satu negara dengan infrastruktur penyelematan gunung terbaik, dan juga salah satu negara yang memprivatisasi sebagaian bisnis penyelamat gunungnya untuk mendorong perusahaan dalam membantu menjaga keselamatan pendaki di gunung-gunung Swiss [4].

Salah satu masalah yang paling sering dihadapi oleh pendaki gunung adalah saat cuaca dipegunungan tiba-tiba berubah ekstrim dan menyebabkan rute mendaki mereka tertutup dan mereka harus mencari rute lain [4]. Kebanyakan turis pendaki gunung tidak terbiasa dengan keterampilan bertahan hidup yang dimiliki pendaki senior seperti membaca peta, sehingga untuk dibuatlah aplikasi peta yang dapat digunakan untuk membuat rute secara tepat [4].

Aplikasi peta umumnya menggunakan salah satu dari 2 metode untuk memberi rute bagi penggunanya, antara dari memberi peta yang telah diinspeksi dan dinilai oleh seorang petugas [5], atau dengan membuat peta secara otomatis dengan algoritma pencari rute [6]. Rute yang telah diinspeksi akan jauh lebih terjamin dan aman, namun karena luasnya pegunungan Swiss akan selalu ada rute yang tidak pernah dijelahi oleh manusia [7]. Untuk itu ada aplikasi peta dengan fungsionalitas mencari rute untuk memenuhi kebutuhan tersebut [8].

1. Tujuan

Tujuan dari tugas akhir ini adalah untuk menerapkan algoritma BFS dengan optimasi BaB untuk kasus mencari rute turun dari pegunungan di Konfederasi Swiss dengan bobot yang tepat dan perbedaan kompleksitas waktu yang berdampak. Tujuan ini dapat dibagi menjadi 3 bagian, antara lain:

1. Mengabungkan algoritma BFS dan BaB
2. Menerapkan algoritma baru untuk pencarian rute turun pegunungan Swiss
3. Menganalisa bobot-bobot yang dapat digunakan untuk algoritma tersebut
4. Tinjauan Pustaka
5. *BFS*

Salah satu metode untuk mencari rute terpendek antara dua titik adalah *Breadth First Search* (BFS) [9]. BFS akan selalu dapat mencari rute yang optimal karena mencari seluruh jalur rute secara merata [9]. BFS memiliki kelemahan berupa kompleksitas waktu yang sangat tinggi, sampai rute nonoptimal dapat mencapai titik akhir rute sebelum BFS berhasil menemukan rute optimalnya. BFS umum digunakan untuk kasus dengan jumlah titik pencarian sedikit, namun terkalahkan oleh algoritma lain seperti Djikstra dan A\* untuk kasus masukan besar [10].

1. Branch and Bound

Salah satu cara untuk melakukan optimasi pada BFS adalah dengan menerapkan konsep *Branch and Bound* (BaB) [11]. BaB adalah metode pencarian berbobot dimana diutamakan terlebih dahulu solusi atau *node* dengan bobot yang lebih rendah atau lebih tinggi [11]. Dengan memberi bobot pada solusi yang lebih awal dan bobot pada solusi yang lebih mendekati tujuan, BaB BFS, atau juga disebut FIFO Search, berusaha menyeimbangkan BFS dan BaB untuk mendapat solusi optimal dengan kompleksitas waktu yang lebih rendah [11].

1. Bobot

Kesultian terbesar dalam menggunakan *informed search* seperti FIFO Search adalah bagaimana mendapat bobot yang tepat untuk menyelesaikan masalah pencarian [9]. Jika bobot terlalu lemah, maka dampak penerapan bobotnya akan berkurang dan FIFO Search akan selambat BFS biasa, sedangkan jika bobot terlalu kuat, maka FIFO Search akan mulai mengikuti bobot tersebut daripada dengan bijak mempertimbangkan informasi lain yang ada pada keadaan [12]. Kelemahan ini ada di setiap algoritma pencarian berbobot termasuk A\* [9].

1. Metodologi

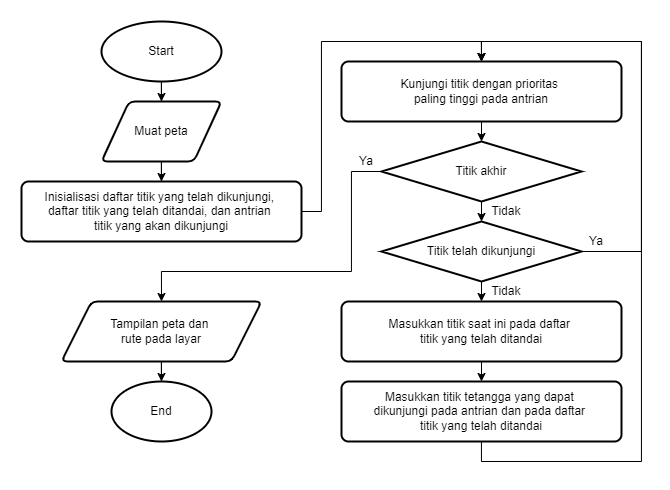
Metodologi penerapan algoritma yang dirancang adalah antara lain melalui:

1. Data

Data yang digunakan diambil dari model swissALTI3D yang disediakan oleh Kantor Serikat Topografi Swiss (Swisstopo). Data yang digunakan berupa *scan* elevasi area 1x1 km dengan ketepatan 2 meter dalam format koordinat XYZ.

1. *Algoritma*

Algoritma yang digunakan adalah gabungan antara BFS dan *Branch and Bound*. Algoritma mengikuti struktur BFS dengan sebuah antrian, namun untuk mengakomodasi bobot-bobot *Branch and Bound* digunakan sebuah priority queue dibanding queue biasa. Sesuai konsepnya, algoritma memberi bobot ke masing-masing langkah yang dapat diambil diurutkan berdasarkan langkah yang telah lama masuk dan langkah yang memiliki elevasi lebih rendah. Algoritma kemudian melihat titik dengan bobot terendah, menjelajahinya jika belum dan melewatinya jika sudah. Setelah menjelajahi sebuah titik, algoritma memasukkan titik-titik yang belum ada di *priority queue* untuk dijelajahi.



Gambar 1. Flowchart program

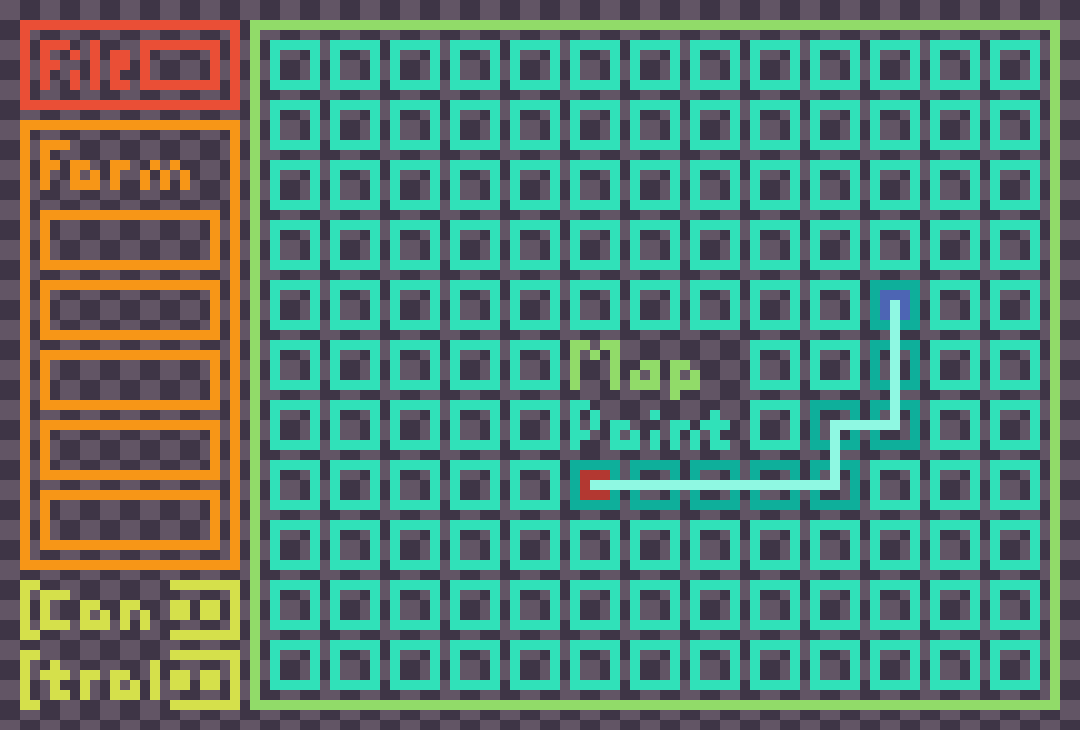
Untuk penentuan prioritas langkah yang dikunjungi digunakan rumus seperti pada Persamaan 1. Prioritas tertinggi adalah langkah dengan nilai bobot terendah.

*Recency* adalah urutan langkah masuk ke dalam antrian, dimulai dari 0 dan bertambah setiap 1 untuk setiap langkah. *Elevation* adalah tingkat ketinggian pada langkah tersebut, dalam skala 0 sampai 25. *Recency weight* dan *elevation weight* adalah bobot atau koefisien yang diberikan pada nilai *recency* dan *elevation* dalam menentukan prioritas langkah.

1. *Rancangan Program*

Program akan dibuat menggunakan JavaScript dengan *framework* React, serta *framework* CSS Tailwind, alat pembantu menjalankan program Vite, dan di-*hosting* pada Github Pages. Aset program dibuat dengan Aseprite, dan digunakan Python dan Google Colab sebagai alat menganalisa pengujian. Program berisi beberapa komponen untuk menampilkan kerja algoritma antara lain:

1. Tampilan grid untuk gambar peta
2. Kolom input file untuk memasukkan data peta
3. Form berisi kolom-kolom terkait jalan kerja algoritma seperti bobot dan jarak antar langkah simulasi.
4. Tombol-tombol kendali untuk mengatur tampilan jalan simulasi
5. Tombol untuk mengubah titik awal dan titik akhir rute



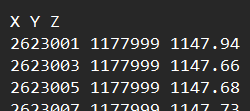
Gambar 2. Rancangan program

1. Implementasi

Implementasi penerapan algoritma yang telah dirancang adalah antara lain melalui:

1. Input dan Output Program

Input program berupa data ketinggian dalam format XYZ. Untuk penerapan ini, data dari swissALTI3D mengenai ketinggian wilayah Swiss digunakan. Pengukuran dilakukan dalam skala meter. Satu titik pada peta data XYZ merepresentasikan ketinggian pada sebuah area 2x2 meter persegi. Satu *file* data XYZ dari swissALTI3D berisi 500x500 titik atau area sekitar 1x1 kilometer persegi. X dan Y adalah posisi relatif titik pada wilayah Swiss, dan Z adalah ketinggian pada titik tersebut.



Gambar 3. Bentuk input

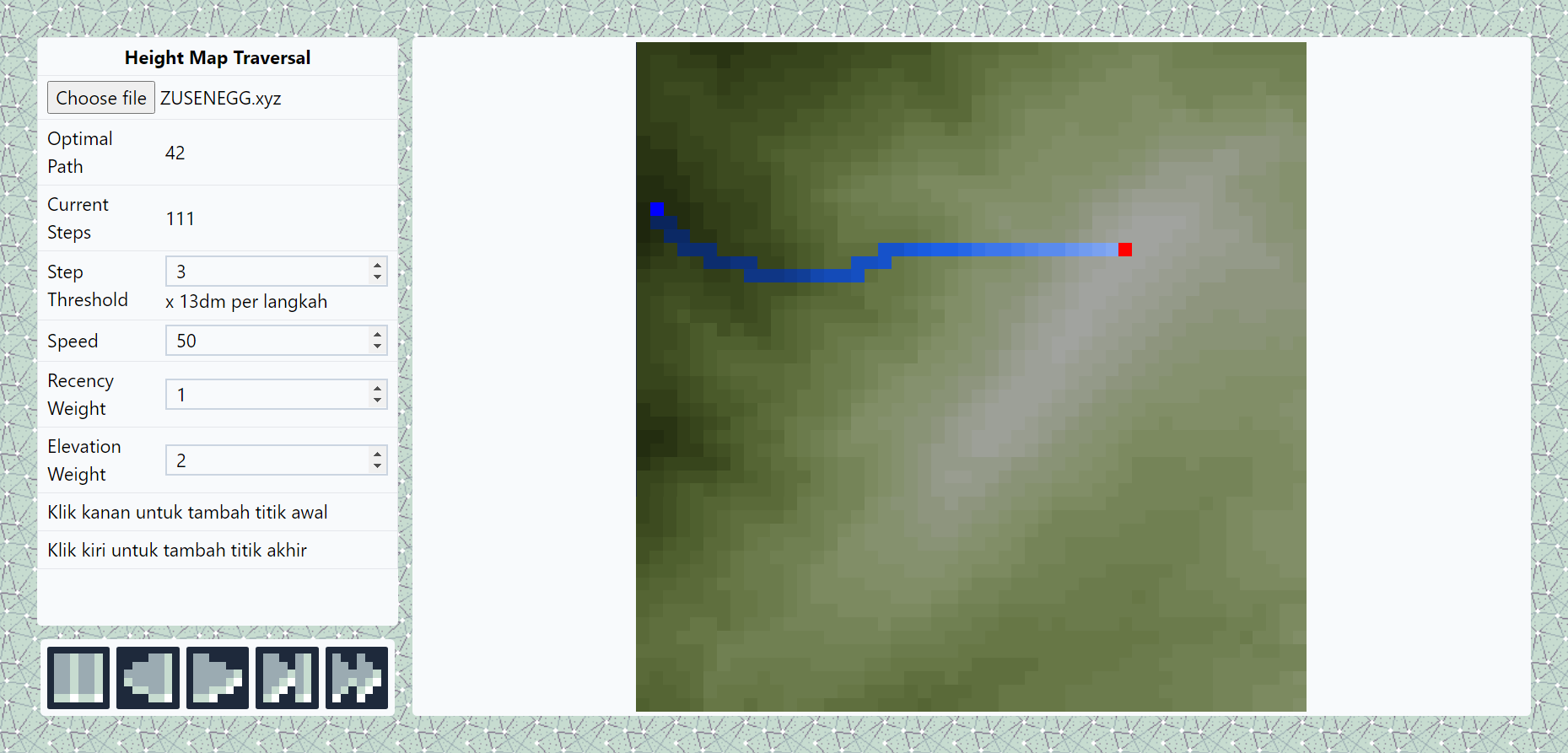
Output program berupa rute terpendek antara titik awal dan titik akhir yang ditentukan. Satu titik pada peta program merepresentasikan 100 titik pada peta data XYZ atau 20x20 meter persegi.



Gambar 4. Bentuk output

1. *Tampilan Program (GUI)*

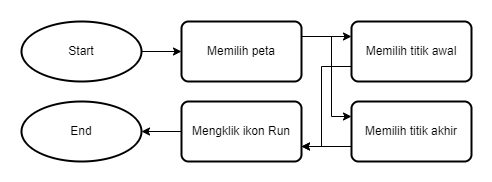
Tampilan GUI program dirancang dengan ReactJS dan Tailwind, dengan ikon dan latar belakang digambar menggunakan Aseprite.



Gambar 5. Tampilan program

1. *Penjelasan Cara Penggunaan*

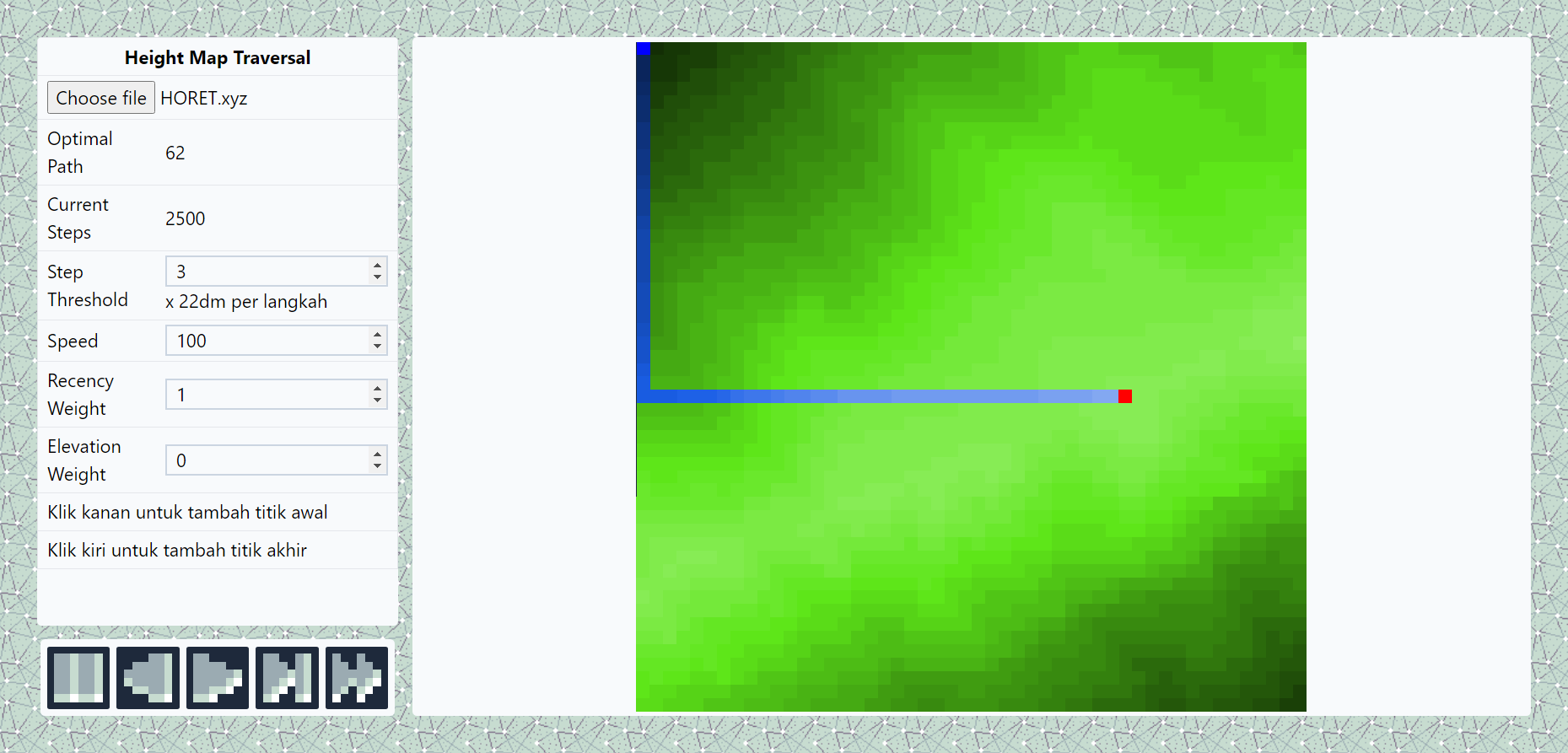
Pengguna pertama memilih data ketinggian yang ingin digunakan sebagai basis pencarian rute. Pengguna kemudian memiliki titik awal dan titik akhir rute. Pengguna lalu mengklik tombol Run untuk memulai pencarian rute.



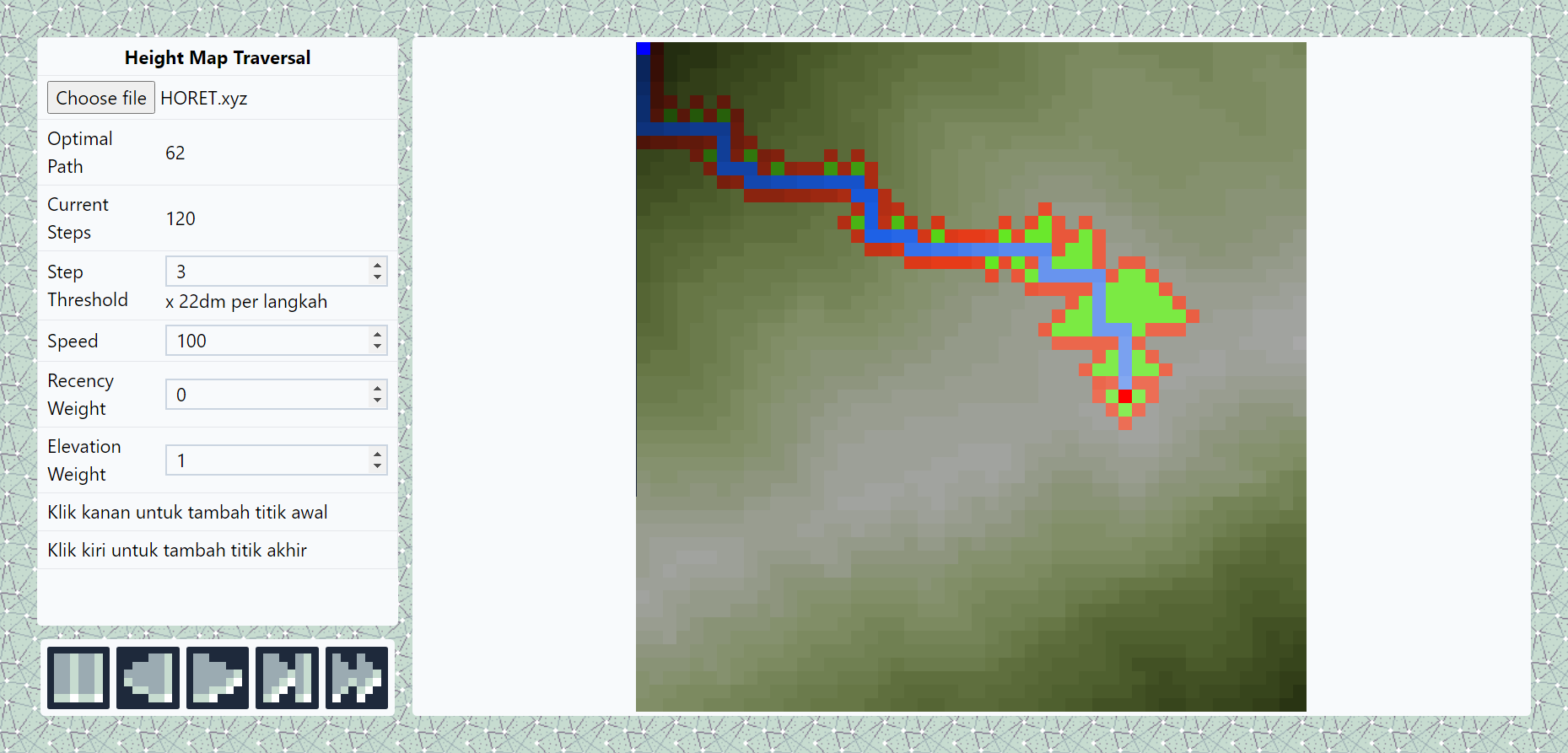
Gambar 6. Flowchart penggunaan program

Untuk mengubah kecepatan jalannya simulasi, pengguna dapat mengubah angka pada kolom *Speed*. Kolom *Step Threshold* adalah mengatur batasan kenaikan dan penurunan ketinggian yang dapat dilakukan per langkah. Untuk pengujian ini digunakan nilai 3 sebagai aproksimasi kemampuan mendaki rata-rata pendaki [8].

Untuk mengubah bobot yang digunakan oleh algoritma, pengguna dapat mengubah angka pada kolom *Recency* *Weight* untuk bobot yang diberi pada node yang baru masuk dan angka pada kolom *Elevation Weight* untuk bobot yang diberi pada *node* yang lebih rendah.



Gambar 7. Bobot untuk BFS



Gambar 8. Bobot untuk Greedy Best First Search

1. Pengujian

Pengujian dilakukan terhadap bentuk-bentuk peta berbeda, titik awal dan titik akhir berbeda, dan dengan rasio bobot-bobot berbeda.

Tabel 1. Area peta yang diuji

|  |
| --- |
| **Area Peta** |
| Zusenegg |
| Horet |
| Oberwald |

Tabel 2. Posisi titik awal dan titik akhir yang diuji

|  |  |
| --- | --- |
| **Titik Awal** | **Titik Akhir** |
| Titik tertinggi | Titik terendah |
| Titik terendah | Titik tertinggi |
| Titik setara | Titik setara |

Tabel 3. Rasio bobot

|  |  |
| --- | --- |
| **Recency Weight** | **Elevation Weight** |
| 1 | 0 |
| 1 | 1 |
| 1 | 2 |
| 1 | 3 |
| 1 | 6 |

Tabel 4. Pengujian terhadap peta Zusenegg

|  |  |  |  |
| --- | --- | --- | --- |
| **Zusenegg** | **Tinggi – rendah** | **Rendah – tinggi** | **Setara – setara** |
| **1 : 0** |  |  |  |
| **1 : 1** |  |  |  |
| **1 : 2** |  |  |  |
| **1 : 3** |  |  |  |
| **1 : 6** |  |  |  |

Tabel 5. Pengujian terhadap peta Horet

|  |  |  |  |
| --- | --- | --- | --- |
| **Horet** | **Tinggi – rendah** | **Rendah – tinggi** | **Setara – setara** |
| **1 : 0** |  |  |  |
| **1 : 1** |  |  |  |
| **1 : 2** |  |  |  |
| **1 : 3** |  |  |  |
| **1 : 6** |  |  |  |

Tabel 6. Pengujian terhadap peta Oberwald

|  |  |  |  |
| --- | --- | --- | --- |
| **Oberwald** | **Tinggi – rendah** | **Rendah – tinggi** | **Setara – setara** |
| **1 : 0** |  |  |  |
| **1 : 1** |  |  |  |
| **1 : 2** |  |  |  |
| **1 : 3** |  |  |  |
| **1 : 6** |  |  |  |

Didapat hasil dari pengujian yang telah dilakukan, dimana terdapat *Optimal Path* atau jumlah langkah jalur terpendek yang ditemukan dan *Current Step* atau pada langkah keberapa solusi rute ditemukan.

Tabel 7. Hasil pengujian terhadap peta Zusenegg

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Zusenegg** | **Tinggi – rendah** | | **Rendah – tinggi** | | **Setara – setara** | |
| Path | Steps | Path | Steps | Path | Steps |
| **1 : 0** | 39 | 1878 | 39 | 1175 | 58 | 2316 |
| **1 : 1** | 39 | 1133 | 39 | 1443 | 58 | 2125 |
| **1 : 2** | 43 | 134 | 39 | 1849 | 58 | 1636 |
| **1 : 3** | 57 | 184 | 39 | 2266 | 62 | 1105 |
| **1 : 6** | 63 | 120 | 41 | 2490 | 70 | 834 |

Tabel 8. Hasil pengujian terhadap peta Horet

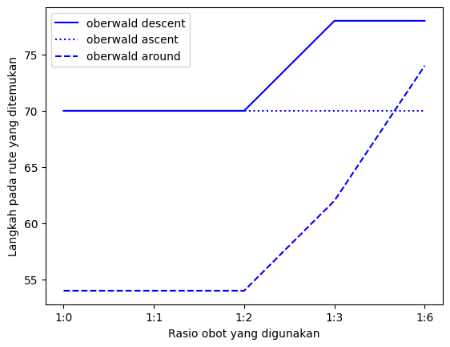
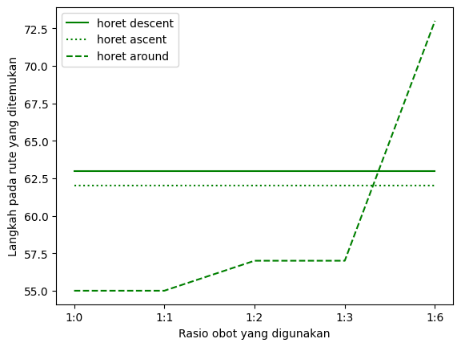
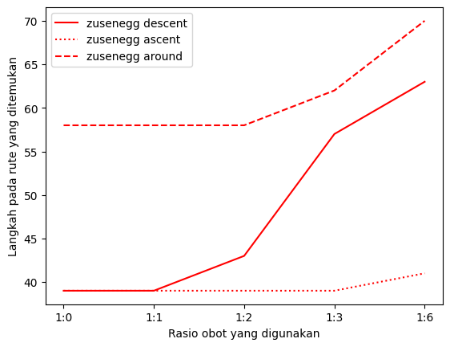
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Horet** | **Tinggi – rendah** | | **Rendah – tinggi** | | **Setara – setara** | |
| Path | Steps | Path | Steps | Path | Stepss |
| **1 : 0** | 63 | 2500 | 62 | 1812 | 55 | 2228 |
| **1 : 1** | 63 | 2389 | 62 | 2011 | 55 | 2205 |
| **1 : 2** | 63 | 1610 | 62 | 2505 | 57 | 2173 |
| **1 : 3** | 63 | 1047 | 62 | 2331 | 57 | 2069 |
| **1 : 6** | 63 | 706 | 62 | 2401 | 73 | 2052 |

Tabel 9. Hasil pengujian terhadap peta Oberwald

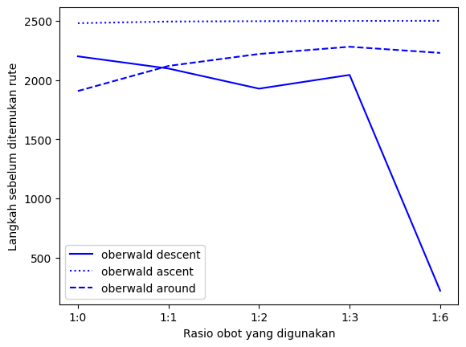
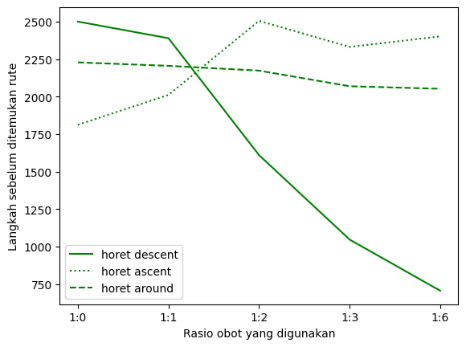
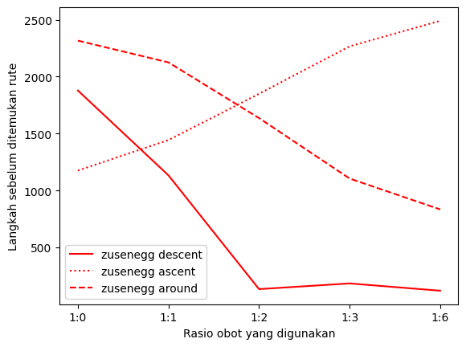
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Oberwald** | **Tinggi – rendah** | | **Rendah – tinggi** | | **Setara – setara** | |
| Path | Steps | Path | Steps | Path | Steps |
| **1 : 0** | 70 | 2200 | 70 | 2480 | 54 | 1907 |
| **1 : 1** | 70 | 2097 | 70 | 2494 | 54 | 2119 |
| **1 : 2** | 70 | 1927 | 70 | 2497 | 54 | 2220 |
| **1 : 3** | 78 | 2043 | 70 | 2499 | 62 | 2281 |
| **1 : 6** | 78 | 220 | 70 | 2500 | 74 | 2229 |

1. Analisis

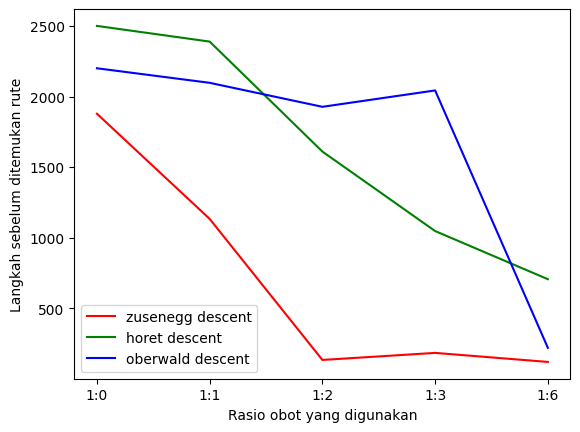
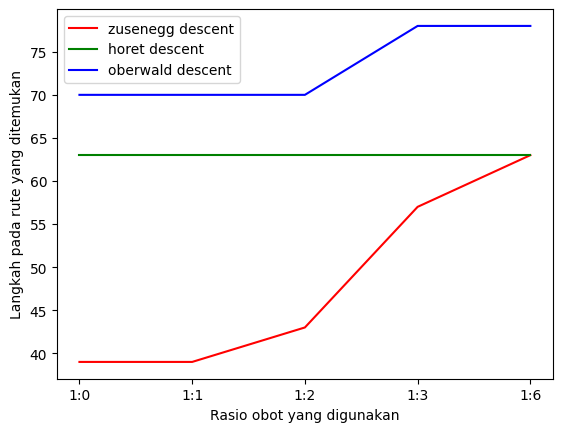
Dapat dilihat bahwa dengan menggunakan algoritma BFS BaB untuk mencari dari titik yang tinggi ke titik yang rendah, terdapat penurunan jumlah langkah yang diperlukan sebelum ditemukannya sebuah solusi, namun juga terdapat kenaikan jumlah langkah yang diperlukan pada rute untuk mencapai tujuan. Ini berarti ada *trade off* antara kecepatan pencarian rute dengan kualitas rute yang ditemukan, yang dipengaruhi oleh bobot-bobot yang digunakan.



Gambar 9. Jumlah langkah pada rute yang ditemukan



Gambar 10. Langkah yang diperlukan sebelum ditemukan rute



Gambar 12. Graf jumlah langkah pada rute dan jumlah langkah pada pencarian untuk kasus penurunan gunung

Dapat dilihat bahwa pada kasus penurunan gunung pada Gambar 12, algoritma berhasil merendahkan jumlah langkah yang diperlukan untuk mencari rute. Algoritma juga berhasil mencapai rute optimal sampai pada bobot 1:1, dan semakin berkurangnya rasio bobot, semakin jauh algoritma dari rute optimal.

Dapat dilihat pula bahwa untuk setiap kasus peletakan titik awal dan titik akhir, rute yang ditemukan selalu akan optimal pada rasio bobot 1:0 dan 1:1, dan walau ada kenaikan pada jumlah langkah yang diperlukan untuk mencari rute, kenaikan tersebut tidak sepadan dengan penurunan jumlah langkah yang diperoleh pada kasus lain.

Dari ini dapat dihipotesiskan bahwa semakin besar rasio bobot, semakin besar jumlah langkah yang diperlukan untuk menemukan rute dan semakin besar kemungkinan rute yang ditemukan adalah rute optimal. Dapat dihipotesiskan pula bahwa berdasarkan hasil pengujian yang telah dilakukan, rute optimal selalu akan ditemukan jika rasio bobot antara 1:0 sampai 1:1. Kedua hipotesis ini perlu diuji lagi pada penelitian selanjutnya.

1. Kesimpulan dan Saran

Didapat bahwa algoritma telah diimplementasi dengan baik dan berhasil melakukan penurunan jumlah langkah yang dibutuhkan untuk mendapatkan rute optimal penurunan area Swiss. Penurunan langkah yang diperlukan dapat berkisar antara 10% sampai 30% untuk pencarian rute turun optimal.

Untuk penelitian selanjutnya ada beberapa hal yang dapat diselidiki, seperti apakah algoritma ini dapat diterapkan pada area peta lain, apakah algoritma ini dapat diadaptasikan untuk mendeteksi apakah rute turun atau naik, apakah algoritma ini selalu optimal untuk seluruh kasus, dan seterusnya.

Ada pula investigasi lebih seksama yang dapat dilakukan pada bobot yang digunakan, seperti apakah bobot 1:1 batasnya atau justru kurang atau lebih dari itu batas sebenarnya, dan mengapa ada batas bobot yang dapat digunakan.

Terakhir pula dapat diselidiki bagaimana perbandingan algoritma yang telah dibuat dengan algoritma bersifat informed, berhubung algoritma ini sendiri juga menggunakan konsep *heuristic* seperti algoritma informed lain. Apakah dengan menggunakan algoritma ini dapat dibuat sebuah algoritma informed search yang optimal seperti A\* dan Djikstra, atau apakah algoritma ini tergolong algoritma greedy dan tidak selalu dapat mendapatkan nilai optimal.

Ucapan Terima Kasih

Terima kasih kepada Universitas Multi Data Palembang yang telah mengajarkan saya konsep-konsep terkait pemrograman yang telah membawa saya selama ini, terutama Unit Kegiatan Mahasiswa Programming yang telah sangat mendukung terhadap pembelajaran pemrograman saya dan memperkenalkan saya pada React. Terima kasih pula kepada beribu-ribu programmer terdahulu yang telah berbagi ilmu dan cerita mereka, baik pembuat artikel atau video pemrograman, dosen-dosen, dan teman sekalian yang telah membantu memberi wawasan mereka.

Daftar Pustaka

1. Elizabeth Kay Berner and R. A. Berner, Global Environment. Princeton University Press, 2012.
2. OECD, OECD Economic Surveys: Switzerland 2024. OECD Publishing, 2024.
3. Wipo, Global Innovation Index 2023. 2023.
4. “Handbook for Investors,” S-GE. https://www.s-ge.com/en/publication/handbook-investors/handbook-investors
5. “swisstopo - Apps on Google Play,” play.google.com. https://play.google.com/store/apps/details?id=ch.admin.swisstopo&hl=en&gl=US (accessed May 28, 2024).
6. “SwitzerlandMobility - Apps on Google Play,” play.google.com. https://play.google.com/store/apps/details?id=ch.schweizmobil&hl=en (accessed May 28, 2024).
7. L. Watts, “Hyke: A Mobile App to Assist in Hiking,” 22ND ANNUAL GEORGIA COLLEGE STUDENT RESEARCH CONFERENCE, 2019, Available: https://kb.gcsu.edu/src/2019/friday/10/
8. “Lecture 3” inst.eecs.berkeley.edu. https://inst.eecs.berkeley.edu/~cs188/sp22/assets/slides/ (accessed May 28, 2024).
9. I.-C. Chang, H.-T. Tai, F.-H. Yeh, D.-L. Hsieh, and S.-H. Chang, “A VANET-Based A\* Route Planning Algorithm for Travelling Time- and Energy-Efficient GPS Navigation App,” International Journal of Distributed Sensor Networks, vol. 9, no. 7, p. 794521, Jul. 2013, doi: https://doi.org/10.1155/2013/794521.
10. K. Deilami et al., “Allowing Users to Benefit from Tree Shading: Using a Smartphone App to Allow Adaptive Route Planning during Extreme Heat,” Forests, vol. 11, no. 9, p. 998, Sep. 2020, doi: https://doi.org/10.3390/f11090998.
11. M. Setak, M. Habibi, H. Karimi, and M. Abedzadeh, “A time-dependent vehicle routing problem in multigraph with FIFO property,” Journal of Manufacturing Systems, vol. 35, pp. 37–45, Apr. 2015, doi: https://doi.org/10.1016/j.jmsy.2014.11.016.
12. A. Dymkov, “TRAVEL ROUTE PLANNING AND TRACKING APPS,” SYNCHROINFO JOURNAL, vol. 7, no. 2, pp. 22–31, 2021, doi: https://doi.org/10.36724/2664-066x-2021-7-2-22-31.