

ALGORITMA GREEDY COLORING PADA WEBSITE ENSIKLOPEDIA PETA

¹Ibnu Aldi Nugroho, ²Fauziah, ³Nur Hayati

^{1,2,3}Fakultas Teknologi Komunikasi dan Informatika Universitas Nasional
Jl. Sawo Manila No.61, RT.14/RW.7, Pejaten Bar., Kec. Ps. Minggu,
Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta 12520

¹nibnualdi@gmail.com, ²fauziah@civitas.unas.ac.id, ³nurhayati@civitas.unas.ac.id

Abstrak

Pewarnaan graf banyak dimanfaatkan karena hasil dari pewarnaan graf dapat dimanfaatkan pada berbagai macam sistem, seperti proses penjadwalan dan pewarnaan wilayah peta pada penelitian ini. Tujuan dalam implementasi pewarnaan graf di sini yaitu untuk mendapatkan jumlah warna seminimal mungkin dalam pewarnaan 44 batas wilayah desa Jakarta Pusat. Masalah pewarnaan graf pada peta desa pada Jakarta Pusat menggunakan algoritma greedy yang ada pada peta berbasis web dan pembuatan ensiklopedia sebagai pendukung deskripsi profil atau sejarah singkat pada wilayah yang telah diberi warna. Greedy algorithm adalah salah satu bentuk algoritma yang berkembang pada saat ini dalam perkara optimasi dengan membentuk solusi langkah demi langkah. Pemberian warna pada graf bisa diselesaikan dengan menggunakan algoritma greedy. Penyelesaian terbaik dalam pewarnaan graf yaitu dengan menentukan bilangan kromatik $\chi(G)$ menggunakan algoritma greedy. Pewarnaan graf pada peta dilakukan dengan menggunakan algoritma greedy sehingga diperoleh jenis warna seminimal mungkin tanpa adanya wilayah yang saling berbatasan menggunakan warna yang sama. Berdasarkan hasil penelitian didapatkan total warna paling sedikit pada pemberian warna pada peta desa pada Jakarta Pusat sebanyak 44 wilayah adalah 6 jenis warna.

Kata Kunci: Algoritma Greedy, Algoritma Greedy Coloring, Mapbox, Sistem Informasi Geografis

Abstract

Graph coloring is used, because the results of graph coloring can be used in several systems, such as the scheduling and the coloring of map areas in this study. The goal in implementing graph coloring here is to get as few colors as possible in the 44 borders of villages in Jakarta Pusat. The problem of coloring graphs on the Jakarta Pusat's villages map uses a greedy algorithm that exists on web-based maps and makes encyclopedias to support descriptions of areas that have been colored. The greedy algorithm is a form of algorithm that is currently developing in optimization problems by forming a step-by-step solution. Greedy algorithm can be applied to resolve the problem of graph coloring. The best solution to answer a graph coloring problem is to determine the chromatic number $\chi(G)$ using the greedy algorithm. Graph coloring on a map is done using a greedy algorithm to obtain the minimum number of colors possible without adjacent regions using the same color. Based on the research results, the least number of colors used in coloring the village map in Central Jakarta, which consists of 44 areas, is 6 types of colors.

Keywords: Greedy Algorithm, Greedy Coloring Algorithm, Geographic Information System, Mapbox

PENDAHULUAN

Perkembangan teknologi informasi serta komunikasi yang pesat telah mempengaruhi berbagai macam hal seperti waktu, tenaga, serta biaya melalui kecepatan dan ketepatan penyampaian informasi, sehingga penggalian informasi dari tempat lain dapat dilakukan dengan cepat. Penggunaan situs web adalah salah satu cara penyampaian informasi. Kebutuhan masyarakat yang banyak menggunakan teknologi dilatarbelakangi oleh perkembangan teknologi. Penggunaan teknologi oleh masyarakat mempengaruhi pertumbuhan dan perkembangan teknologi sistem informasi yang tepat dan efisien.

Pada pembuatan map biasanya diperlukan pewarnaan yang dapat digunakan untuk membedakan wilayah satu dengan lainnya sehingga terlihat jelas batas wilayah yang ditentukan. Namun banyaknya wilayah akan mengakibatkan pemakaian warna yang banyak dan tidak efisien.

Pada umumnya masyarakat sudah menggunakan perangkat gawai, maupun komputer karena banyaknya perangkat lunak yang ada, salah satunya di dunia pendidikan yang banyak menyediakan perangkat lunak untuk media pembelajaran dan sebagainya. Namun hanya sedikit program perangkat lunak yang membahas tentang geografi, mulai dari profil maupun sejarah wilayah geografis tertentu untuk mendukung

pewarnaan wilayah yang telah ditentukan.

Pada penentuan banyaknya warna secara optimal diperlukan algoritma yang dapat menentukan wilayah mana yang bertetangga dengan wilayah lainnya, sehingga wilayah yang tidak bertetangga dapat menggunakan warna yang telah digunakan. Pada kartografi, peta dianggap akurat dan bagus jika mudah dikenali oleh pembaca dan mencakup semua elemen yang diperlukan seperti legenda, skala, judul, warna, dan simbol. Salah satu kelebihan warna adalah memudahkan untuk mengidentifikasi beberapa area dengan garis lurus. Peta yang menunjukkan wilayah milik masing-masing negara diwarnai dengan corak berbeda. Peta akan sangat tidak efektif jika menggunakan terlalu banyak warna. Maka dari itu, diperlukan teknik mewarnai peta dengan warna sesedikit mungkin [1].

Teori graf bisa diterapkan di beberapa bidang seperti dalam masalah pemetaan, transportasi, jaringan telekomunikasi, riset operasional, dan sebagainya. Teori tentang graf sudah diperluas dengan menggunakan beragam algoritma dengan kelebihan dan kekurangan dalam mengatasi kasus tersebut. Masalah pewarnaan graf adalah salah satu masalah yang terkenal di teori graf. [2]. Salah satu topik yaitu matematika diskrit yang sudah lama dikenal dan umum dipakai dalam berbagai bidang yaitu pewarnaan graf [3]. Masalah pewarnaan graf terkenal dalam optimisasi kombinatoral [4].

Pewarnaan graf adalah metode pelabelan

graf. Metode ini menetapkan warna ke simpul, sisi atau daerah. Suatu pewarnaan memiliki indeks kromatik atau angka kromatik yang dilambangkan dengan $\chi'(G)$, ini merupakan bilangan paling kecil yang digunakan untuk pewarnaan sisi pada graf G [5]. Pada pewarnaan titik, setiap titik pada graf G mendapat warna sehingga dua titik atau wilayah yang berdekatan atau bertetangga diberi warna berbeda dan menghasilkan jumlah warna yang sedikit mungkin [6].

Graf sederhana yang dilambangkan $G = (V, E)$ terdiri dari himpunan simpul $V = \{v_1, v_2, v_3, \dots, v_n\}$ dan himpunan sisi $E = \{e_1, e_2, e_3, \dots, e_m\}$ sehingga setiap e_i ($1 \leq i \leq m$) secara unik diasosiasikan dengan sepasang simpul tak beraturan (v_j, v_k) ($1 \leq j, k \leq n$) dan j, k [1,2]. Matriks ketetanggaan atau *adjacency matrix* ditunjukkan oleh $A(G)$, sebuah matriks simetris $n \times n$, dengan $A(j, k) = 1$ ($1 \leq j, k \leq n$) if $(v_j, v_k) \in E(G)$; dan $A(j, k) = 0$ ($1 \leq j, k \leq n$) sebaliknya [7]. Algoritma *greedy coloring* menjamin bahwa $(\Delta+1)$ warna benar-benar digunakan dan grafik menggunakan warna yang lebih sedikit daripada $(\Delta+1)$ [8] di mana Δ merupakan derajat maksimum berdasarkan grafik ketergantungan [9].

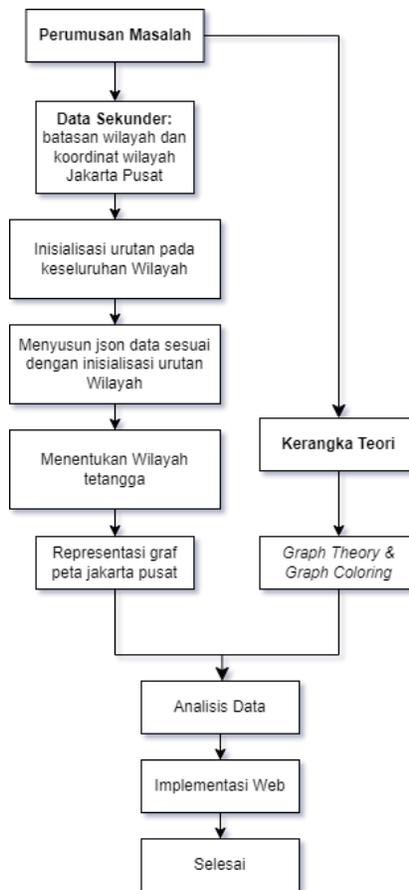
Terdapat Penelitian terdahulu yang telah merancang map medan dengan algoritma *greedy coloring*, dengan mendefinisikan 4 warna, menggunakan *adjacency matrix*, implementasi *greedy coloring* dengan Bahasa pemrograman python, dan pada penelitian terdahulu ini

terdapat 21 wilayah yang direpresentasikan sebagai vertex [1]. Adapun penelitian terdahulu yang lainnya membuat aplikasi pemesanan tiket dengan mengimplementasi algoritma *greedy* untuk mengolah data tata letak kursi penumpang serta mengoptimalkannya agar calon penumpang dapat memesan tiket dan menentukan kursi yang diinginkan [10]. Selain itu ada juga penelitian terdahulu yang membahas pengembangan algoritma pewarnaan graf yang cepat untuk mengatasi masalah pada metode yang ada, dan harus efisien untuk semua jenis instance graf, yaitu algoritma baru di mana algoritma Depth First Search digunakan untuk memberikan perintah ke simpul grafik [11].

Berdasarkan penelitian terdahulu untuk mendapatkan informasi titik yang bertetangga dengan titik lainnya secara optimal, maka perlu menggunakan algoritma *greedy*, dimana dalam kasus ini yaitu algoritma *greedy coloring*. Penelitian ini mengimplementasikan algoritma *greedy coloring* yang digunakan untuk membedakan warna wilayah atau desa yang bertetangga pada web ensiklopedia yang dibuat. Pada penelitian ini meneliti wilayah Jakarta Pusat yang di dalamnya terdapat 44 wilayah atau desa.

METODE PENELITIAN

Pada Gambar 1 ditunjukkan flowchart penelitian yang dilakukan yang diawali dari indentifikasi masalah, pembuatan web aplikasi, sampai presentasi hasil.



Gambar 1. Flowchart Tahapan Penelitian

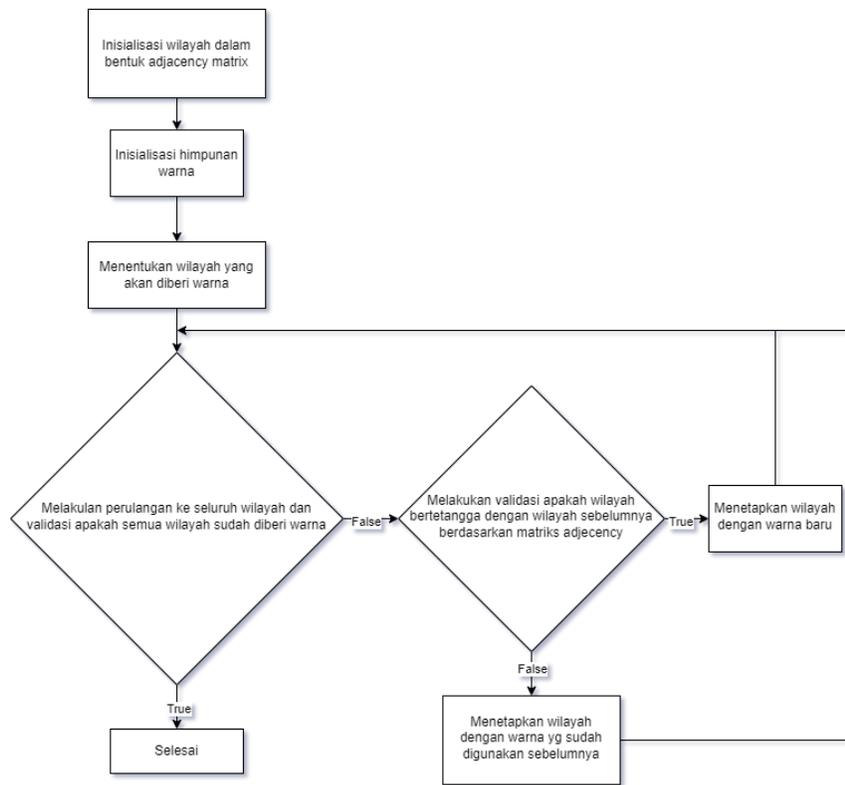
Data

Dalam penelitian ini digunakan data administrasi batas wilayah Jakarta Pusat yang berasal dari suatu manajemen projek/system versioning `https://github.com/pararawendy/border-desa-indonesia-geojson/blob/master/indonesia_villages_border_geojson.zip` dan artikel yang berisi data yang

dibutuhkan seperti informasi geografis wilayah dari mulai geometri, koordinat, dan properti lainnya seperti nama provinsi dan nama desa.

Algoritma Greedy Coloring

Algoritma *greedy coloring* pada penelitian ini menjadi salah satu pokok pembahasan. Pada Gambar 2 dapat dilihat *flowchart* algoritma *greedy coloring*.



Gambar 2. Flowchart Algoritma Greedy Coloring

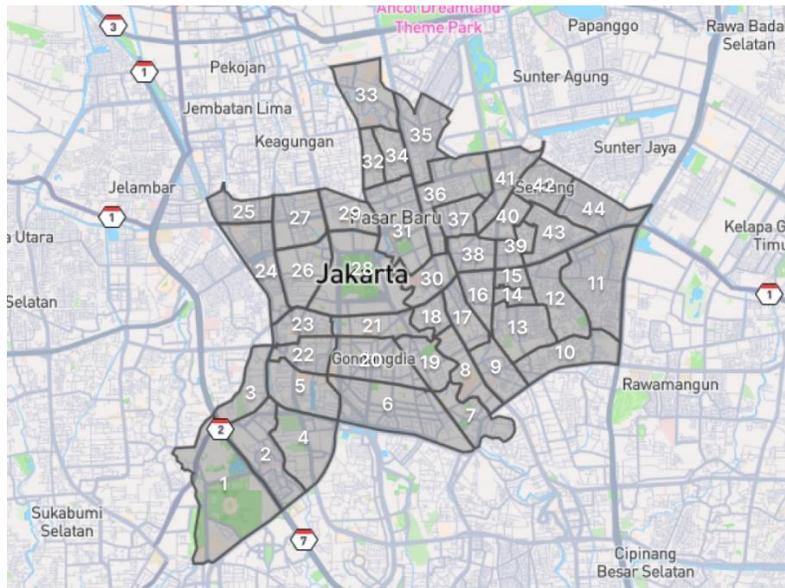
Pada Gambar 2 flowchart algoritma greedy coloring dijelaskan proses pertama yaitu membuat atau menginisialisasi wilayah atau titik pada peta dalam bentuk *adjacency matrix*. Tahap berikutnya diperlukan inisialisasi himpunan warna untuk pewarnaan nanti. Setelah itu dimulailah fungsi menentukan warna untuk menseleksi titik mana yang akan menggunakan warna tertentu. Jika masih terdapat wilayah atau titik yang belum diberi warna, maka fungsi menentukan warna akan dijalankan kembali. Jika seluruh wilayah atau titik sudah diberi warna, maka algoritma sudah

selesai.

HASIL DAN PEMBAHASAN

Adapun 44 desa atau kelurahan yang berada di Jakarta Pusat yang pada penelitian ini digunakan sebagai implementasi algoritma greedy coloring.

Berdasarkan Gambar 3, wilayah Jakarta Pusat dibagi menjadi 44 desa atau kelurahan. Tabel 1 merupakan penjelasan dari penomoran pada peta di Gambar 3 yang terdiri dari 44 wilayah.



Gambar 3. Peta Jakarta Pusat

Tabel 1. Daftar Desa Jakarta Pusat

No.	Nama Provinsi	No.	Nama Provinsi
1.	Gelora	23.	Kampung Bali
2.	Bendungan Hilir	24.	Cideng
3.	Petamburan	25.	Duri Pulo
4.	Karet Tengsin	26.	Petojo Selatan
5.	Kebon Melati	27.	Petojo Utara
6.	Menteng	28.	Gambir
7.	Pegangsaan	29.	Kebon Kelapa
8.	Kenari	30.	Senen
9.	Paseban	31.	Pasar Baru
10.	Rawasari	32.	Karang Anyar
11.	Cempaka Putih Timur	33.	Mangga Dua Selatan
12.	Cempaka Putih Barat	34.	Kartini
13.	Johar Baru	35.	Gunung Sahari Utara
14.	Kampung Rawa	36.	Gunung Sahari Selatan
15.	Galur	37.	Kemayoran
16.	Tanah Tinggi	38.	Bungur
17.	Kramat	39.	Harapan Mulia
18.	Kwitang	40.	Utan Panjang
19.	Cikini	41.	Kebon Kosong
20.	Gondangdia	42.	Serdang
21.	Kebon Sirih	43.	Cempaka Baru
22.	Kebon Kacang	44.	Sumur Batu

Berdasarkan Tabel 1, desa diurutkan untuk dapat dilakukan eksekusi awal dalam algoritma *greedy coloring* dalam inisialisasi *adjacency matrix*.

Representasi Wilayah Dalam Bentuk Graf

Setelah memberikan penomoran pada peta provinsi di Jakarta Pusat, selanjutnya menambahkan titik dan graf untuk merepresentasikan hubungan antara poligon satu dengan yang lainnya, sehingga dapat diketahui wilayah mana yang menempel dengan wilayah lainnya. Gambar 4 merupakan

gambaran representasi titik dan graf pada peta desa atau kelurahan di Jakarta Pusat.

Berdasarkan Gambar 4, wilayah Jakarta Pusat direpresentasikan sebagai graf dan dapat dilihat hubungan antara poligon satu dengan lainnya.

Representasi Graf Dalam Bentuk *Adjacency Matrix*

Berdasarkan Gambar 4, peta desa pada Jakarta Pusat telah direpresentasikan ke dalam titik dan graf. Berikut titik dan graf yang telah di implementasikan pada peta desa di Jakarta Pusat.



Gambar 4. Representasi Graf Pada Peta Jakarta Pusat

Gambar 6 merupakan bentuk matriks yang diperoleh dari graf yang ada pada Gambar 5. Dimana 0 merepresentasikan node satu dengan node lainnya tidak memiliki hubungan dan 1 merepresentasikan node satu dengan node lainnya terhubung, sehingga nantinya dapat menyesuaikan pewarnaan pada node-node yang ada.

Penerapan Algoritma

Di uraian sebelumnya sudah dilakukan pengurutan titik atau wilayah pada Jakarta Pusat dari 1 sampai dengan 44 dan juga telah dibuat *adjacency matrix* yang akan digunakan untuk mengidentifikasi hubungan antar setiap wilayah dalam implementasinya.

Berikut tahapan implementasi algoritma *greedy coloring* dalam pewarnaan peta wilayah Jakarta Pusat sebagai berikut:

1. Inisialisasi

Pada implementasi dimulai dengan inisialisasi *adjacency matrix* yang mewakili keterhubungan antar wilayah, inisialisasi variabel *color_class* yang akan menyimpan data *vertex* dengan warna yang ditentukan. Dan juga inisialisasi variabel *vertex* untuk digunakan dalam perulangan.

2. Perulangan pertama

Di sini dilakukan perulangan dengan kondisi *if vertex < matrix.length*, jika kondisi terpenuhi maka blok kode di dalamnya dijalankan dan lanjut ke tahap berikutnya, jika tidak terpenuhi maka

dilanjutkan dengan mengeksekusi kode blok dengan kondisi *if get_color == false*, maka jalankan kode *color_class.push([vertex])*.

3. set variabel *get_color = false*

Jika kondisi pada perulangan pertama terpenuhi, di tahap ini dilakukan inisialisasi variabel *get_color* dengan default value *false*.

4. Perulangan *color_class*

Pada tahap ini masuk ke dalam perulangan kedua. Untuk validasi setiap warna dengan dilakukan pengecekan dengan kondisi *if get_color == false && (ccindex < color_class.length)* jika kondisi terpenuhi maka blok kode di dalamnya dijalankan dan lanjut ke tahap berikutnya, jika tidak terpenuhi maka blok kode di dalamnya tidak dijalankan dan kembali ke perulangan pertama.

5. Pengecekan konflik

Pada perulangan ini dilakukan inisialisasi *found variable* dengan default value *false* dan cek kondisi *if matrix[vertex][current_vertex] != 0* jika konflik ditemukan maka *set found = true* yang berarti *vertex* sudah diberi warna dan tambah *ccindex variable* dengan 1.

6. Tambah *vertex* ke *color_class*

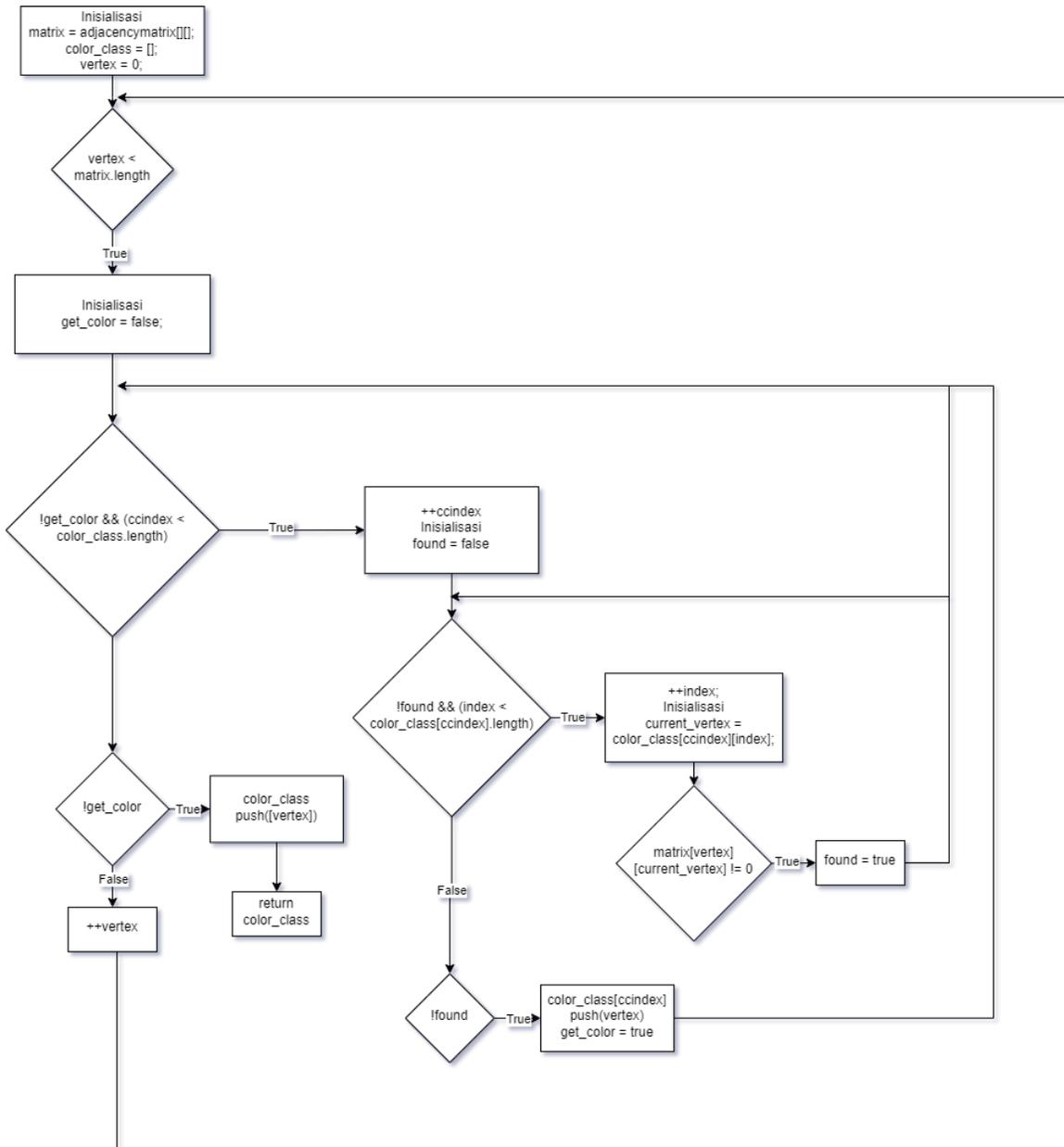
Pada tahap ini dilakukan penambahan *vertex* ke *color_class* dengan kondisi *if found == false* dan *get_color = true*.

1. Warna baru

Setelah pengecekan semua *color_class* dan melakukan validasi *get_color == false* maka gunakan warna baru.

2. Increment vertex

Jika semua baris kode sudah dilakukan maka tambah *vertex* dengan 1 lalu ulang perulangannya sampai semua *vertex* atau wilayah terpenuhi.



Gambar 7. Flowchart Implementasi Algoritma *Greedy Coloring*

Setelah semua algoritma dijalankan, maka akan menghasilkan satu variabel *color_class* dengan isi array 2 dimensi. Sebagai contoh variabel *color_class* akan terlihat pada Gambar 8.

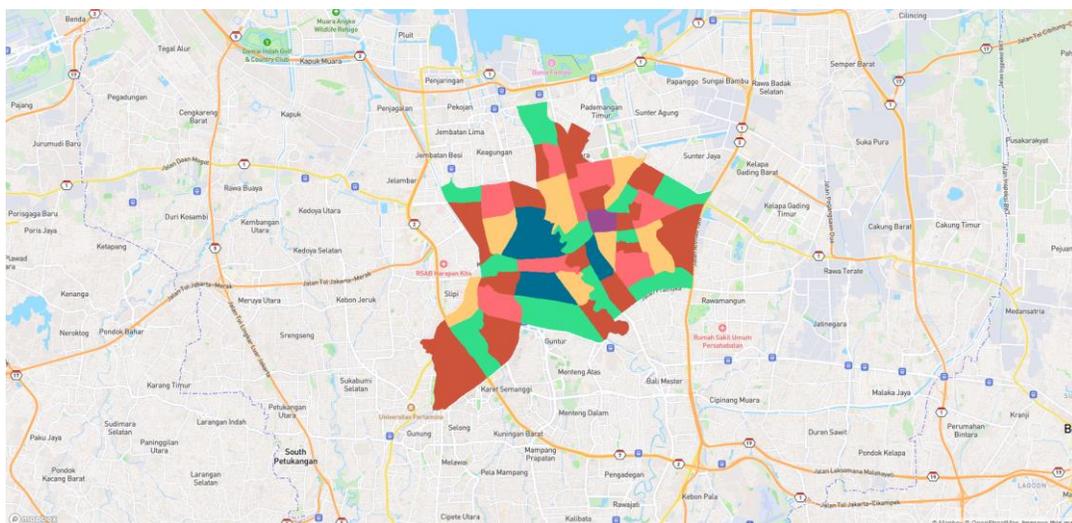
Seperti yang disebutkan di atas, *color_class variable* berisikan array 2 dimensi. Dimana dimensi pertama menentukan warna index yang akan digunakan dan dimensi kedua berisikan wilayah dengan kelompok warna yang

telah ditentukan.

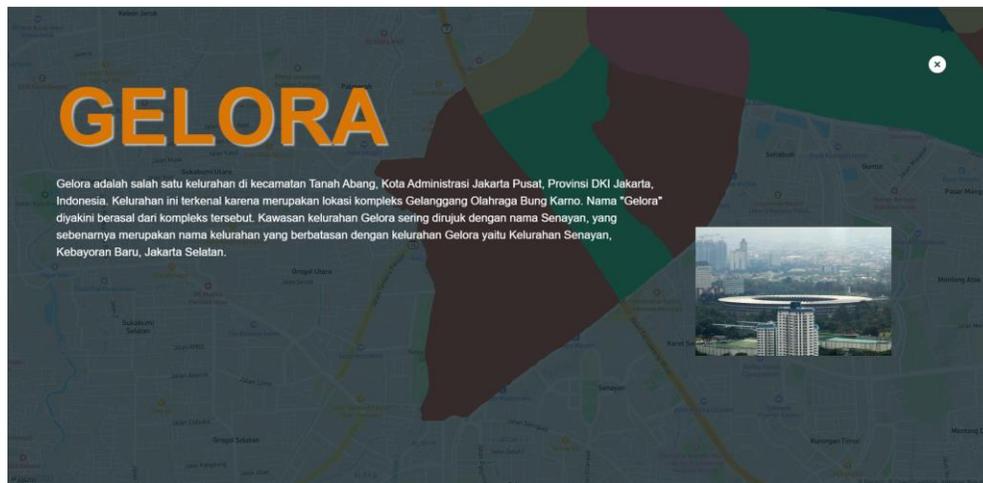
Gambar 9 merupakan halaman utama website map Jakarta Pusat yang telah diketahui batas-batas wilayah desa pada Jakarta Pusat yang dapat dilihat di <https://greedymap.netlify.app/>. Dimana terdapat 44 wilayah/desa yang telah dilakukan pewarnaan dengan menggunakan algoritma *greedy coloring* dengan 6 warna, yaitu orange, hijau, kuning, pink, biru, dan ungu.

```
1 var color_class = [  
2   [0, 3, 6, 8, 10, 13, 17, 21, 23, 28, 31, 34, 36, 38, 41],  
3   [1, 5, 7, 9, 14, 22, 24, 29, 32, 39, 43],  
4   [2, 11, 15, 18, 25, 30, 40],  
5   [4, 12, 20, 26, 33, 35, 42],  
6   [16, 19, 27],  
7   [37],  
8 ];
```

Gambar 8. Hasil Akhir *color_class variable*



Gambar 9. Implementasi Warna Hasil Algoritma Ke Dalam Web



Gambar 10. Halaman Artikel

Pada halaman ini berisi ensiklopedia yang berhubungan dengan wilayah yang dipilih. Gambar 10 menunjukkan artikel yang dipilih dari wilayah/desa yang berada di halaman utama website Gambar 9.

KESIMPULAN DAN SARAN

Berdasarkan hasil dan pembahasan yang telah diuraikan maka dapat diambil kesimpulan bahwa pewarnaan pada peta Jakarta Pusat dapat dilakukan menggunakan *greedy coloring algorithm* dengan cara membuat graf dual dari wilayah yang sudah ditentukan tersebut. Graf dual dari peta Jakarta Pusat terdiri dari 44 titik. Kemudian jumlah warna minimum pewarnaan peta Turki berdasarkan pada penelitian ini diperoleh 6 warna yaitu orange, hijau, kuning, pink, biru, dan ungu. Selain itu algoritma *greedy coloring* dapat diimplementasi ke dalam bentuk web untuk pembuatan ensiklopedia

pada map untuk membedakan warna satu wilayah dengan wilayah yang lain yang bertetangga dan dapat diakses dengan link <https://greedymap.netlify.app/>. Adapun saran untuk penelitian selanjutnya adalah mengoptimalkan kasus pewarnaan graf pada peta tersebut.

DAFTAR PUSTAKA

- [1] N. M. Surbakti and F. Ramadhani, "Implementation of the Greedy Algorithm for Coloring Graph Based on Four-Color Theorem," *sudo Jurnal Teknik Informatika*, vol. 1, no. 4, pp. 178–182, Dec. 2022, doi: 10.56211/sudo.v1i4.157.
- [2] K. Irsal, "PEWARNAAN GRAF PADA PETA MENGGUNAKAN ALGORITMA GREEDY (Studi Kasus : Peta Provinsi DKI Jakarta)."

- [3] M. Sopiyan, A. Gunaryati, and I. Fitri, "Graph coloring Sistem Pendaftaran dan Proses Penjadwalan Data Instruktur Berbasis Web dengan Algoritma Welch-powell," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 6, no. 1, p. 2022, 2022, doi: 10.35870/jti.
- [4] Y. Wang, S. Cai, S. Pan, X. Li, and M. Yin, "Reduction and Local Search for Weighted Graph Coloring Problem." [Online]. Available: www.aaai.org
- [5] M. Y. Fakhri, E. Harahap, and F. H. Badruzzaman, "Implementasi Algoritma Welch-Powell pada Pengaturan Lampu Lalu Lintas Pasteur Bandung," *Jurnal Riset Matematika*, vol. 1, no. 2, pp. 91–98, Dec. 2021, doi: 10.29313/jrm.v1i2.365.
- [6] S. Slamain, N. O. Adiwijaya, M. A. Hasan, D. Dafik, and K. Wijaya, "Local super antimagic total labeling for vertex coloring of graphs," *Symmetry (Basel)*, vol. 12, no. 11, pp. 1–17, Nov. 2020, doi: 10.3390/sym12111843.
- [7] R. Marappan and G. Sethumadhavan, "Complexity analysis and stochastic convergence of some well-known evolutionary operators for solving graph coloring problem," *Mathematics*, no. 3, 2020, doi: 10.3390/math8030303.
- [8] L. Dhulipala, G. E. Blelloch, and J. Shun, "Theoretically Efficient Parallel Graph Algorithms Can Be Fast and Scalable," *ACM Transactions on Parallel Computing*, vol. 8, no. 1, Apr. 2021, doi: 10.1145/3434393.
- [9] Y. Tian, K. Khosoussi, D. M. Rosen, and J. P. How, "Distributed Certifiably Correct Pose-Graph Optimization," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 2137–2156, Dec. 2021, doi: 10.1109/TRO.2021.3072346.
- [10] C. Angelina, N. Koban, and S. R. Nudin, "Penerapan Graph Coloring Menggunakan Algoritma Greedy Pada Aplikasi Pemesanan Tiket Kapal Penyeberangan," *Journal of Informatics and Computer Science*, vol. 02, 2020.
- [11] S. Kumar Gupta, G. & Singh, S. Gupta, and D. P. Singh, "Greedy Graph Coloring Algorithm Based on Depth First Search," *International Journal on Emerging Technologies*, vol. 11, no. 2, pp. 854–862, 2020, [Online]. Available: <https://www.researchgate.net/publication/341371807>