Jurnal EurekaMatika 13(2) (2025) 169-178

Jurnal EurekaMatika

Journal homepage: https://ejournal.upi.edu/index.php/JEM

Pewarnaan Simpul pada Hipergraf dengan Pendekatan Matriks

*Hanifa Nurshabira, Yaya Sukjaya Kusumah, dan Sumanang Muhtar Gozali**

Program Studi Matematika, Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam, Universitas Pendidikan Indonesia, Indonesia

*Corresponding author: gozali@upi.edu

A B S T R A K

Seiring berkembangnya ilmu pengetahuan, diperkenalkan konsep hipergraf yang merupakan generalisasi graf. Graf tersusun dari pasangan himpunan simpul dan himpunan sisi, sedangkan hipergraf tersusun dari pasangan himpunan simpul dan himpunan hyperedge, dimana hyperedge menghubungkan 2 atau lebih simpul pada hipergraf. Pewarnaan simpul pada hipergraf dapat digunakan untuk menyelesaikan berbagai masalah penugasan. Pada penelitian ini dilakukan pewarnaan simpul pada hipergraf dengan menggunakan algoritma pewarnaan simpul pada hipergraf dengan pendekatan matriks. Agar pewarnaan simpul pada hipergraf efisien, pada penelitian ini dibentuk program aplikasi komputer. Program ini disusun dengan menterjemahkan algoritma tersebut ke dalam bahasa tingkat tinggi Python. Program dijalankan dengan memberikan input himpunan hyperedge, himpunan simpul, dan juga banyak warna yang akan digunakan. Hasil dari program ini adalah visualisasi semua skema pewarnaan simpul pada hipergraf.

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima 1 Oktober 2025
Direvisi 20 Oktober 2025
Disetujui 28 Oktober 2025
Tersedia online 2 November 2025
Dipublikasikan 2 Novembe 2025

Kata Kunci:
Hipergraf,
Hyperedge,
Pewarnaan simpul,
Teori graf.

A B S T R A C T

As the science develops, the concept of hypergraphs is introduced, which is a generalization of graphs. Graphs are composed of pairs of vertices set and edges set, while hypergraphs are composed of pairs of vertices set and hyperedges set. Hyperedges connect 2 or more vertices in the hypergraph. Vertex coloring on hypergraphs can be used to solve various mapping problems. In this study, vertex coloring on hypergraphs is carried out using the vertex coloring algorithm on hypergraphs with a matrix approach. In order to make the vertex coloring on hypergraph efficient, a computer application program is formed in this research. This program is prepared by translating the algorithm into Python high-level language. The program is executed by giving input of hyperedge set, vertex set, and also many colors to be used. The result of this program is a visualization of all vertex coloring schemes on the hypergraph.

Keywords:
Graph theory,
Hyperedge,
Hypergraph,
Vertex coloring.

1. PENDAHULUAN

Pada tahun 1973, Claude Berge seorang ilmuwan matematika asal Perancis membahas konsep hipergraf dalam publikasinya yang merangkum berbagai penelitian sebelumnya terkait hipergraf. Pada konsep hipergraf, dikenal istilah *hyperedge* yang merupakan generalisasi sisi pada konsep graf. Sisi pada graf merupakan garis yang menghubungkan 2 simpul, sedangkan *hyperedge* merupakan himpunan bagian dari himpunan simpul pada hipergraf yang setiap anggotanya memiliki hubungan (Berge, 1973; Toft, 1974).

Hipergraf $H = (V, E)$ adalah pasangan himpunan simpul berhingga $V = \{v_i | i = 1,2, \dots, n\}$ dan himpunan *hyperedge* $E = \{e_i | i = 1,2, \dots, m\}$, dengan e_i bukan himpunan kosong dan merupakan himpunan bagian dari V, sedemikian sehingga $\bigcup_{i=1}^{m} e_i = V$ (Berge, 1973). Misalkan $H = (V, E)$ adalah sebuah hipergraf, dengan $|V| = n, |E| = m$. Jika $v_i, v_j \in e_i, e_j \in E$ dan terdapat $v \in V$ sedemikian sehingga $v \in e_i \cap e_j$, maka e_i, e_j dikatakan insiden (Bahmanian & Sajna, 2015). Insidensi dari setiap simpul dengan *hyperedge* dapat direpresentasikan ke dalam matriks insiden berukuran $m \times n$. Misalkan m_{ij} adalah elemen dari M pada baris ke-i dan kolom ke-j, m_{ij} akan bernilai 1 jika $v_j \in e_i$, dan bernilai 0 jika tidak (Putri *et al.*, 2020).

Seperti pada graf, terdapat juga pewarnaan simpul pada hipergraf. Sebuah pewarnaan simpul pada H adalah sebuah alokasi sebanyak k warna pada simpul-simpul tersebut, sehingga sebuah simpul memiliki tepat satu warna, menggunakan sebanyak k warna, dan tidak ada *hyperedge* dengan kardinalitas lebih dari satu yang monokromatik (Lovász, 1972; Bretto, 2013).

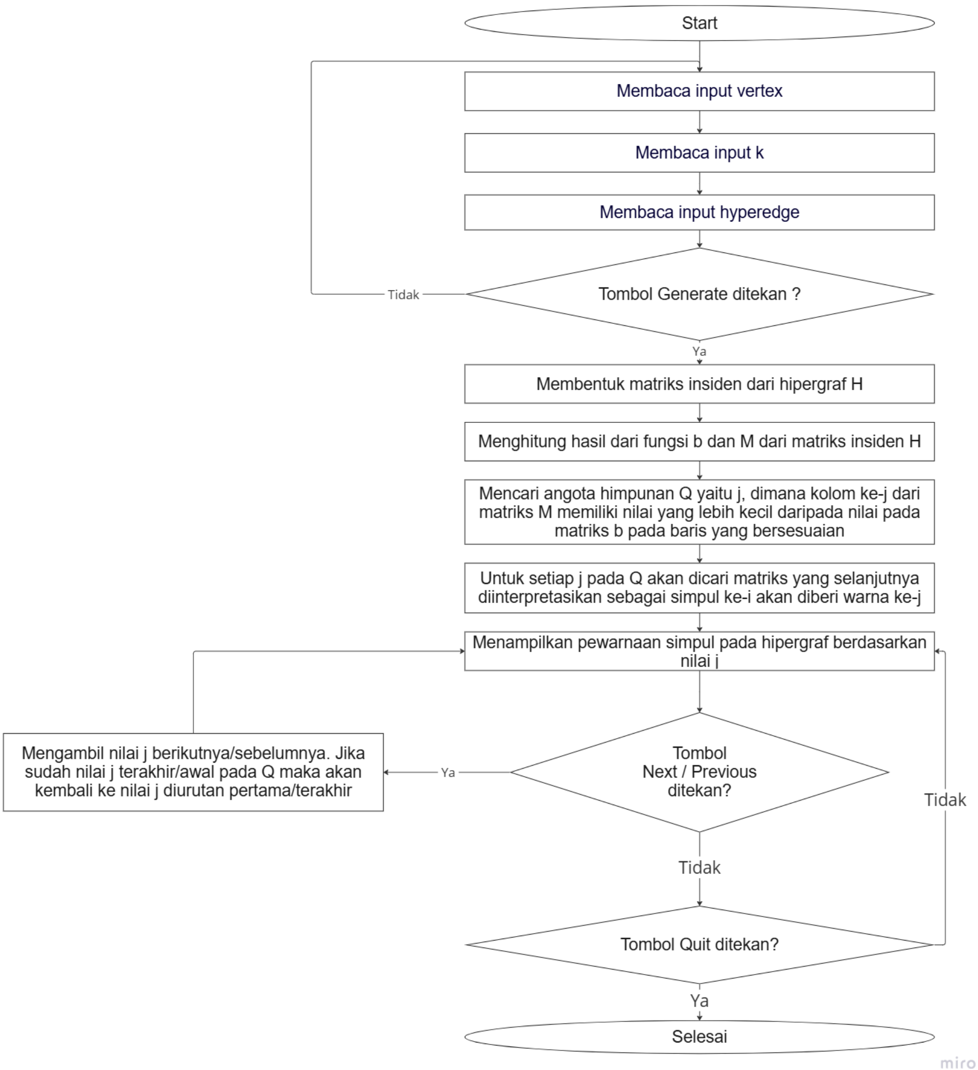
Pewarnaan simpul pada hipergraf dapat digunakan untuk menentukan jumlah frekuensi minimal untuk sebuah jaringan ponsel dan menentukan tempat pembuangan minimal untuk bahan kimia berbahaya (Bretto, 2013). Wanless & Wood (2022) menyebutkan bahwa pewarnaan simpul pada hipergraf dapat diaplikasikan untuk menjadwalkan pekerjaan pada mesin dengan alat yang digunakan bersama dan mendesain kode pengoreksi kesalahan yang mampu mendeteksi dan mengoreksi beberapa kesalahan dalam sebuah *codeword*.

Meng & Feng (2014) melakukan pewarnaan simpul hipergraf dengan pendekatan matriks, dengan menggunakan matriks insiden dari sebuah hipergraf, dicari skema-skema pewarnaan yang mungkin, dan kemudian hasil pewarnaan diaplikasikan untuk menyelesaikan masalah penyimpanan. Misalkan sebuah hipergraf $H = (V, E)$ dengan $V = \{v_1, v_2, \dots, v_n\}$ dan $E = \{e_1, e_2, \dots, e_m\}$. Matriks insiden dari H adalah A, elemen dari A pada baris ke-i dan kolom ke-j dinotasikan sebagai a_{ij}. Dipilih sebanyak q warna untuk melakukan pewarnaan.

Hipergraf karena memiliki sisi yang melibatkan beberapa simpul, maka visualisasinya menjadi lebih rumit dibandingkan visualisasi graf biasa. Dalam penelitian ini kami mengembangkan metode visualisasi graf dengan menggunakan matriks insiden yang dibangkitkan secara otomatis berdasarkan input. Program ini diimplementasikan menggunakan Python.

2. METODE

Program dibuat menggunakan bahasa tingkat tinggi Python dengan menggunakan library seperti numpy, mathplot, hypernetx. Alur kerja dari program aplikasi pewarnaan simpul pada hipergraf dapat dilihat pada Gambar 1.

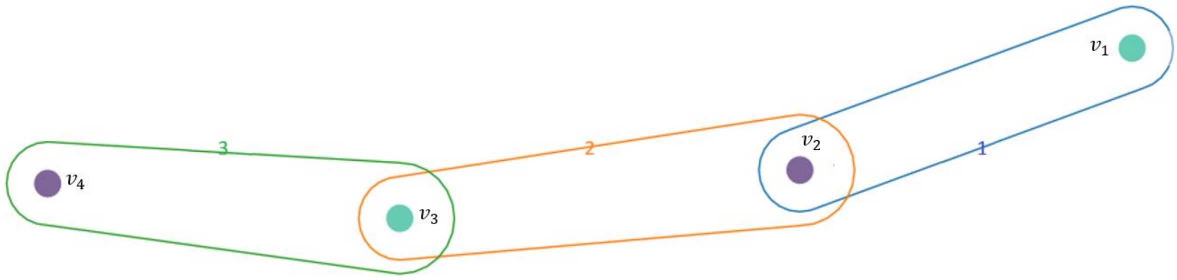


Gambar 1. Alur Kerja Program Aplikasi

2.1 Validasi program

Setelah pembuatan program, perlu pengujian program dengan menggunakan contoh hipergraf yang telah dilakukan perhitungan secara manual. Misalkan sebuah hipergraf $H = (V, E)$ dengan $V = \{v_1, v_2, v_3, v_4\}$ dan $E = \{e_1, e_2\}$ dimana $e_1 = \{v_1, v_2\}, e_2 = \{v_2, v_3\}, e_3 = \{v_3, v_4\}$. Gambar 2 menampilkan visualisasi hipergraf H. Matriks insiden dari H adalah

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$



Gambar 2. Visualisasi hipergraph H dengan matrik insiden A

Berikut ini hasil perhitungan b dan M untuk H

$$
\begin{aligned}
b &= \sum_{s=1}^{3}\sum_{t=s+1}^{4}\begin{bmatrix} a_{1s}a_{1t} \\ a_{2s}a_{2t} \\ a_{3s}a_{3t} \end{bmatrix} \\
&= \begin{bmatrix} a_{11}a_{12} \\ a_{21}a_{22} \\ a_{31}a_{32} \end{bmatrix} + \begin{bmatrix} a_{11}a_{13} \\ a_{21}a_{23} \\ a_{31}a_{33} \end{bmatrix} + \begin{bmatrix} a_{11}a_{14} \\ a_{21}a_{24} \\ a_{31}a_{34} \end{bmatrix} + \begin{bmatrix} a_{12}a_{13} \\ a_{22}a_{23} \\ a_{32}a_{33} \end{bmatrix} + \begin{bmatrix} a_{12}a_{14} \\ a_{22}a_{24} \\ a_{32}a_{34} \end{bmatrix} + \begin{bmatrix} a_{13}a_{14} \\ a_{23}a_{24} \\ a_{33}a_{34} \end{bmatrix} \\
&= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}
\end{aligned}
$$

Diperlukan nilai $S_{1,2}^4, S_{2,2}^4, S_{3,2}^4, S_{4,2}^4$ untuk menghitung M_{st}^H, $s \in \{1,2,3\}$, $t \in \{2,3,4\}$ yang akan digunakan untuk menghitung M.

$$S_{1,2}^4 = \delta_2[1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2]$$

$$S_{2,2}^4 = \delta_2[1\ \ 1\ \ 1\ \ 1\ \ 2\ \ 2\ \ 2\ \ 2\ \ 1\ \ 1\ \ 1\ \ 1\ \ 2\ \ 2\ \ 2\ \ 2]$$

$$S_{3,2}^4 = \delta_2[1\ \ 1\ \ 2\ \ 2\ \ 1\ \ 1\ \ 2\ \ 2\ \ 1\ \ 1\ \ 2\ \ 2\ \ 1\ \ 1\ \ 2\ \ 2]$$

$$S_{4,2}^4 = \delta_2[1\ \ 2\ \ 1\ \ 2\ \ 1\ \ 2\ \ 1\ \ 2\ \ 1\ \ 2\ \ 1\ \ 2\ \ 1\ \ 2\ \ 1\ \ 2]$$

$$M_{12}^H = \begin{bmatrix} 1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1 \end{bmatrix}$$

$$M_{13}^H = \begin{bmatrix} 1\,1\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,1\,1 \end{bmatrix}$$

$$M_{14}^H = \begin{bmatrix} 1\,0\,1\,0\,1\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,1\,0\,1 \end{bmatrix}$$

$$M_{23}^H = \begin{bmatrix} 1\,1\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,1\,1 \end{bmatrix}$$

$$M_{24}^H = \begin{bmatrix} 1\,0\,1\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,1\,0\,1 \end{bmatrix}$$

$$M_{34}^H = \begin{bmatrix} 1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1 \end{bmatrix}$$

$$
\begin{aligned}
M &= \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} J_2 M_{12}^H + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} J_2 M_{13}^H + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} J_2 M_{14}^H + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} J_2 M_{23}^H + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} J_2 M_{24}^H + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} J_2 M_{34}^H \\
&= \begin{bmatrix} 1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1 \\ 1\,1\,0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1 \\ 1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1 \end{bmatrix}
\end{aligned}
$$

Akan diperoleh $Q = \{6,11\}$. Untuk setiap $j \in Q$ akan dicari $x_i = S_{i,2}^4 \delta_{2^4}^j, i = 1,2, \dots ,4$

$$j = 6, [x_1, x_2, x_3, x_4] = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$j = 11, [x_1, x_2, x_3, x_4] = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

Skema-skema 2-*coloring* pada H adalah sebagai berikut :

1. Untuk $j = 6$, diperoleh skema 1 : $C_1 = \{v_1, v_3\}$ (Biru), $C_2 = \{v_2, v_4\}$ (Hitam), dan
2. Untuk $j = 11$, diperoleh skema 2 : $C_1 = \{v_2, v_4\}$ (Hitam), $C_2 = \{v_1, v_3\}$ (Biru).

Kedua skema ini akan menjadi bahan validasi kelayakan program yang dirancang menggunakan Python.

Terkait algoritma pewarnaan simpul pada hipergraf dengan pendekatan matriks, operasi matriks yang akan digunakan adalah sebagai berikut.

1. Block diagonal
   Matriks $A = Diag\{A_1, A_2, \dots, A_n\}$ adalah blok diagonal matriks dengan elemen a_{ii} adalah A_i untuk setiap $i \in 1,2,..,n$ (Meng & Feng, 2014).
2. Produk Kronecker

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1m}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}$$

   (Fernández, 2016).

3. Produk semi tensor kiri, misal n banyak kolom A, p banyak baris B, $\alpha = KPK(n, p)$

$$A \ltimes B = \left(A \otimes I_{\frac{\alpha}{n}}\right)\left(B \otimes I_{\frac{\alpha}{p}}\right)$$

   (Cheng & Zhang, 2003).

4. Produk Hadamard, misal $A,\ B \in M_{mxn}$

$$A \odot B = \begin{bmatrix} a_{11}b_{11} & \cdots & a_{1n}b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1}b_{m1} & \cdots & a_{mn}b_{mn} \end{bmatrix}$$

   (Pickard *et al.*, 2024)
5. Swap matriks
   Swap matriks $W_{[m,n]}$ berukuran $mn \times mn$ disebut sebagai *swap* matriks jika kolom-kolomnya dilabeli dengan $(11,\ 12, \dots,\ 1n, \dots ,\ m1,\ m2,\ \dots,\ mn)$ dan baris-barisnya dilabeli dengan $(11,\ 21,\ \dots,\ m1, \dots ,\ 1n,\ 2n,\ \dots ,\ mn)$, dengan elemen matriks

$$w_{(IJ),(ij)} = \delta_{i,j}^{I,J} = \begin{cases} 1, & I = i \ dan \ J = j \\ 0, & selainnya \end{cases}$$

   (Cheng & Zhang, 2003).

Selain itu, terdapat juga matriks yang didefinisikan yang akan digunakan dalam algoritma, berikut matriks-matriks berserta maknanya.

1. δ_q^n adalah kolom ke n dari matriks identitas berukuran $q \times q$. Untuk himpunan δ_q^n, dapat dituliskan sebagai $\Delta_n = \{\delta_n^i | i = 1,2, \dots, n\} = \delta_n[1 \ 2 \ \dots n]$.
2. $$S_{t,q}^n = \delta_q \left[\underbrace{\underbrace{1 \dots 1}_{q^{n-t}} \ \underbrace{2 \dots 2}_{q^{n-t}} \ \cdots \ \underbrace{q \dots q}_{q^{n-t}} \ \cdots \ \underbrace{1 \dots 1}_{q^{n-t}} \ \underbrace{2 \dots 2}_{q^{n-t}} \ \cdots \ \underbrace{q \dots q}_{q^{n-t}}}_{q} \right].$$
3. $M_{r,q^n} = Diag\{\delta_{q^n}^1, \delta_{q^n}^2, \dots, \delta_{q^n}^{q^n}\}$
4. $z_i = (\delta_q^i)^T$
5. $H_q = Diag\{z_1, z_2, \dots, z_q\}$
6. I_n adalah matriks identitas berukuran $n \times n$
7. $J_2 = [1,1, \dots, 1]_q$
8. $M_{st}^H = H_q S_{s,q}^n \left(I_{q^n} \otimes S_{t,q}^n\right) M_{r,q^n}$

Secara lengkap untuk algoritma pewarnaan simpul pada hipergraf dengan pendekatan matriks oleh Meng & Feng (2014) dapat dilakukan dengan mengikut langkah-langkah berikut ini.

1. Menghitung nilai M dan b.
$$b = \sum_{s=1}^{n-1} \sum_{t=s+1}^{n} \begin{bmatrix} a_{1s}a_{1t} \\ \vdots \\ a_{ms}a_{mt} \end{bmatrix}$$
$$M = \sum_{s=1}^{n-1} \sum_{t=s+1}^{n} \begin{bmatrix} a_{1s}a_{1t} \\ \vdots \\ a_{ms}a_{mt} \end{bmatrix} J_2 M_{st}^H$$
2. Menentukan apakah terdapat $j \in \{1,2, \dots, q^n\}$ sedemikian sehingga $Col_j(M) \ll b$. Jika kondisi tidak dipenuhi, proses dihentikan. Jika kondisi dipenuhi, maka tentukan $Q = \{j | Col_j(M) \ll b\}$.
3. Misalkan $\ltimes_{i=1}^n x_i = \delta_{p^n}^j, \forall j \in Q$. Perhatikan bahwa $x_i = S_{i,q}^n(\ltimes_{i=1}^n x_i) = S_{i,q}^n \delta_{p^n}^j, i = 1, 2, \dots, n$. Konstruksi $S_{c_k}^j$, yaitu himpunan simpul-simpul yang diwarnai warna $c_k, k \in 1,2, \dots, q$.
$$S_{c_1}^j = \{v_i | x_i = \delta_q^1, 1 \le i \le n\}$$
$$S_{c_2}^j = \{v_i | x_i = \delta_q^2, 1 \le i \le n\}$$
$$\vdots$$
$$S_{c_q}^j = \{v_i | x_i = \delta_q^q, 1 \le i \le n\}$$

Karena melibatkan operasi matriks yang menghasilkan ukuran matriks berkali-kali lipat, algoritma ini akan sulit dilakukan jika ukuran hipergraf cukup besar. Pada penelitian ini akan dibuat program aplikasi pewarnaan simpul pada hipergraf dengan pendekatan matrik. Untuk memperoleh hasil pewarnaan dengan lebih cepat.

3. HASIL DAN PEMBAHASAN

Algoritma pewarnaan simpul pada hipergraf memiliki langkah-langkah yang terlihat abstrak. Sebenarnya, setiap langkahnya berkaitan dengan proses pewarnaan. Berikut ini makna dari tiap langkan perhitungan matriks pada algoritma.

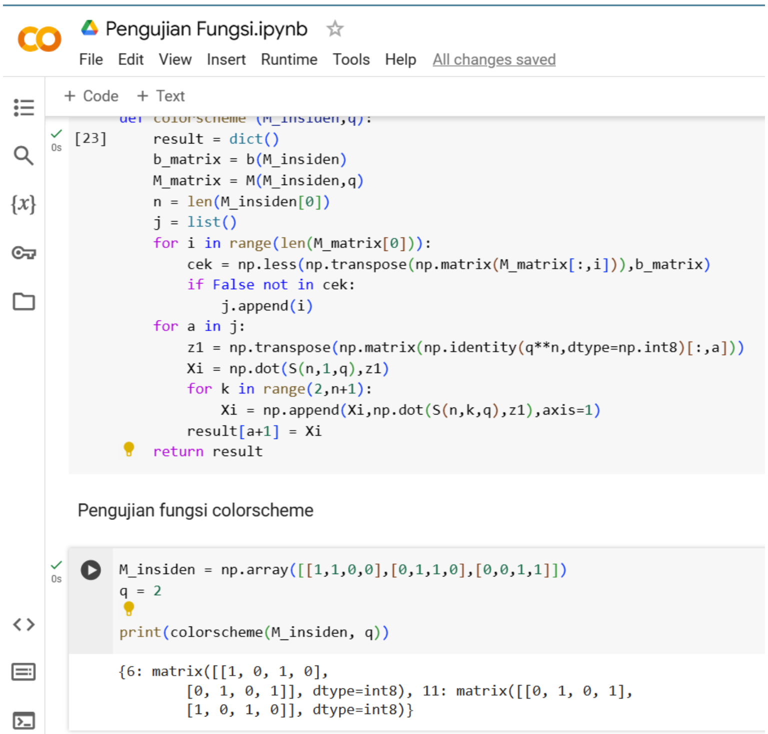
1. Membentuk matriks insiden dari hipergraf. Hal ini bertujuan untuk memberikan informasi setiap simpul dimuat pada *hyperedge* apa saja.
2. Menghitung matriks b. Perhatikan bahwa martiks b memberikan informasi berapa hubungan yang terjadi untuk setiap *hyperedge*. Misalkan jika di dalam $e_1 = \{v_1, v_2\}$, maka hanya ada satu hubungan dalam e_1. Hal ini terlihat pada nilai $Col_1(b) = 1$.
3. Menghitung matriks M, dilakukan untuk mengelompokkan setiap simpul menjadi k kelompok. Matriks M selalu memiliki banyak kolom sebanyak anggota himpunan *hyperedge* dan banyak kolom sebanyak banyak warna dipangkatkan banyak anggota himpunan simpul. Maknanya adalah ada kelompok pewarnaan sejumlah banyak kolom pada matriks M.
4. Mendapatkan himpunan $Q = \{j | Col_j(M) \ll b\}$. Nilai-nilai pada himpunan ini memberikan informasi hasil pengelompokan mana saja yang memenuhi definisi pewarnaan simpul dengan baik.
5. Menghitung matriks $x_i = S^n_{i,q}\delta^j_{p^n}, i = 1, 2, \ldots, n,$ dilakukan untuk memberikan informasi untuk skema dengan j tertentu, simpul diberikan warna ke berapa.

Untuk program aplikasinya telah dirancang fungsi utama yang menjalankan algortima pewarnaan. Fungsi pewarnaan adalah sebagai berikut.

```
def colorscheme(matriks_insiden, q):
    result = dict()
    len_kolom = matriks_insiden.shape[1]
    b_matriks = b(matriks_insiden)
    m_matriks = f_m(matriks_insiden, q)
    indeks = np.where((m_matriks < b_matriks).all(axis=0) == 1)[0]
    temp = np.zeros((q, len_kolom), dtype=np.int8)
    for a in indeks:
        z1 = np.identity(q**len_kolom, dtype=np.int8)[:,a].reshape(-1,1)
        xi = temp
        for i in range(len_kolom):
            xi_col = np.dot(f_s(len_kolom, i+1, q), z1).transpose()
            xi[:,i] = xi_col
        result[a+1] = xi
    return result
```

Adapun untuk fungsi-fungsi lainnya seperti, left_semi_tensor, $S^n_{t,q}, M_{r,q^n}$ H_q, M^H_{st}, a_{st}, b, M dan get_color juga dibuat ke dalam fungsi Pyhton, tetapi penulis tidak menampilkannya. Sebagai bentuk pengujian fungsi diatas, dari hasil perhitungan pada Contoh

1 akan dibandingkan dengan hasil perhitungan dari fungsi Python yang telah dibentuk. Gambar 3 adalah hasil perhitungan program.



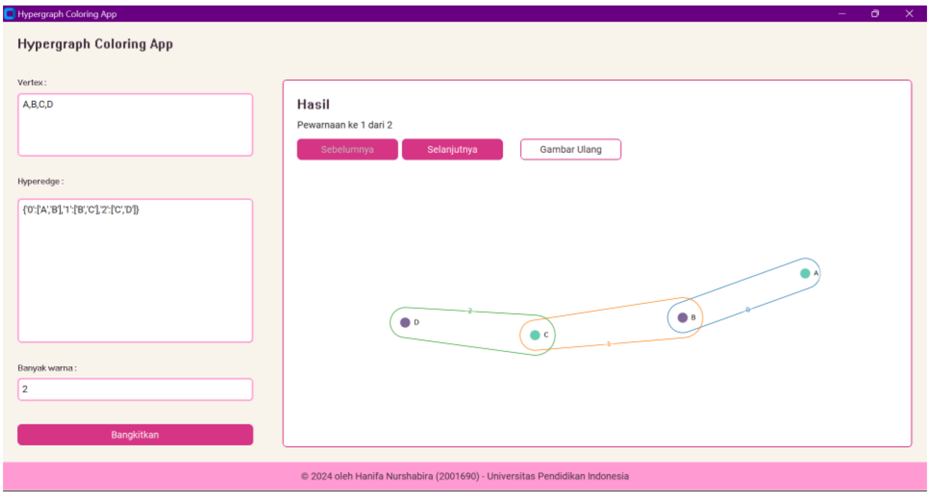
Gambar 3. Hasil Pewarnaan Simpul Hipergraf H dengan Program

Hasil keluaran program dapat dilihat pada Gambar 3. Pada perhitungan program hasilnya berupa dictionary yang kuncinya adalah nilai j dan nilanya adalah matriks yang merepresentasikan hubungan antara simpul (sebagai index kolom) dengan warna (sebagai index baris). Maknanya, skema pewarnaan simpul pada hipergraf yang mungkin berdasarkan hasil perhitungan program adalah sebgaai berikut.

Untuk $j = 6$, diperoleh skema 1 : $C_1 = \{v_1, v_3\}$,$C_2 = \{v_2, v_4\}$

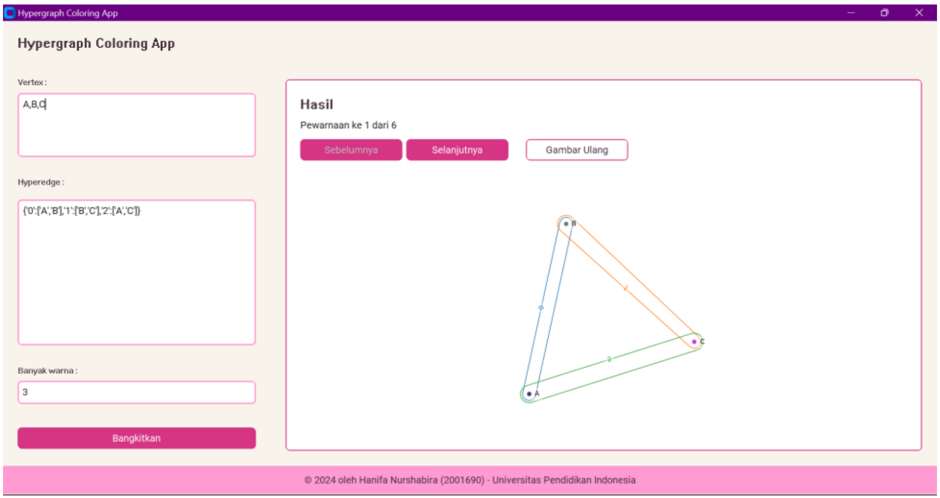
Untuk $j = 11$, diperoleh skema 2 : $C_1 = \{v_2, v_4\}$, $C_2 = \{v_1, v_3\}$

Jika dibandingkan dengan hasil perhitungan pada Contoh 1, skema yang dihasilkan sama. Selanjutnya dibentuk juga GUI untuk tampilan program yang memanfaatkan library Python seperti Customtkinter, HypernetX, dan Mathplotlib yang ditampilkan pada Gambar 4.



Gambar 4. Tampilan GUI dari Program Aplikasi

1. Masukan :
   Vertex : `A,B,C`
   Hyperedge :`{'0':['A','B'],'1':['B','C'],'2':['A','C']}`
   Banyak Warna : `2`
   Keluaran :



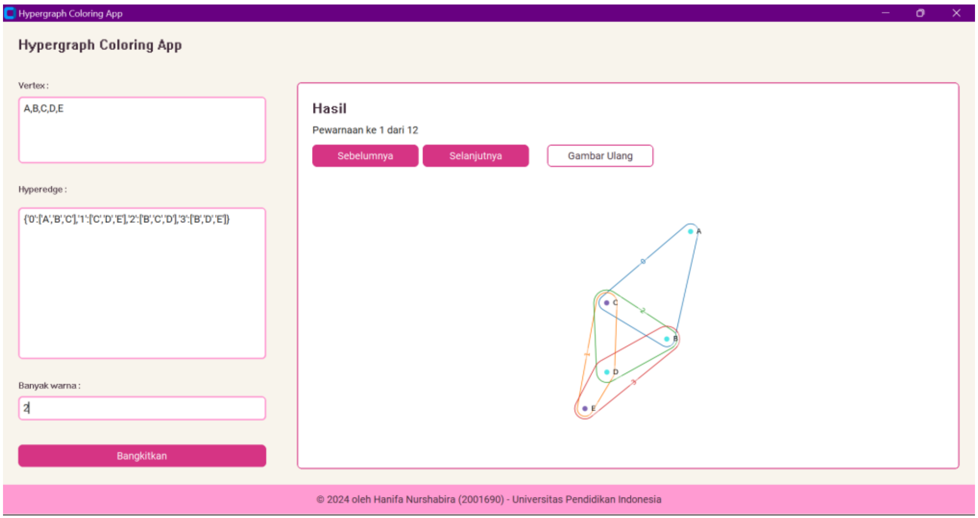
Gambar 5. Hasil pewarnaan hipergraf

2. Masukan :
   Vertex : `A,B,C,D,E`

   Hyperedge : `{'0':['A','B','C'],'1':['C','D','E'],`

   `'2':['B','C','D'],'3':['B','D','E']}`

   Banyak Warna : `2`
   Keluaran :



Gambar 6. Hasil pewarnaan hipergraf

4. KESIMPULAN

Algoritma pewarnaan simpul pada hipergraf memiliki 3 tahapan. Pertama, Menghitung nilai M dan b. Kedua, menentukan apakah terdapat $j \in \{1,2,\ldots,q^n\}$ sedemikian sehingga $Col_j(M) \ll b$. Jika kondisi tidak dipenuhi, proses dihentikan. Jika kondisi dipenuhi, maka tentukan $Q = \{j | Col_j(M) \ll b\}$. Ketiga, konstruksi $S_{c_k}^j$, yaitu himpunan simpul-simpul yang diwarnai warna $c_k, k \in 1,2,\ldots,q$. Maka akan diperoleh semua kemungkinan pewarnaan simpul untuk sebuah hipergraf.

Program pewarnaan simpul pada hipergraf dikonstruksi dengan menggunakan Python dan memanfaatkan library Python seperti *numpy, hypernetx, matplolib*, dan *customtkinter*. Untuk menjalankan program terlebih dahulu perlu memasukan daftar simpul, *dictionary hyperedge*, dan banyak warna yang akan digunakan. Setelah itu, jika tombol "bangkitkan" ditekan, maka gambar hasil pewarnaan simpul akan ditampilkan. Jika tombol selanjutnya atau sebelumnya ditekan, akan ditampilkan skema pewarnaan simpul selanjutnya atau sebelumnya. Skema pewarnaan simpul ini diperoleh dengan menggunakan algoritma pewarnaan simpul yang sudah diterjemahkan kedalam bahasa pemrograman Python.

Hasil pewarnaan pada program sama dengan perhitungan manualnya, dengan demikian program ini dapat membantu pekerjaan pewarnaan simpul pada hipergraf dengan lebih cepat. Program ini tentu saja bisa digunakan untuk masalah penugasan dengan merepresentasikan simpul dan *hyperedge* dengan makna tertentu. Kemudian, dari skema-skema yang ada ditentukan penugasan apa yang dipilih berdasarkan kebutuhan. Ide ini bisa digunakan untuk penelitian selanjutnya.

5. DAFTAR PUSTAKA

Bahmanian, M. A., & Šajna, M. (2015). Hypergraphs: connection and separation. *Theory and Applications of Graphs.*

Berge, C. (1973). Balanced hypergraphs and some applications to graph theory. In *A survey of combinatorial theory* (pp. 15-23). North-Holland.

Bretto, A. (2013). Hypergraph theory. *An introduction. Mathematical Engineering. Cham: Springer*, *1*, 209-216. Cheng, D. Z., & Zhang, L. J. (2003). On semi-tensor product of matrices and its applications. *Acta Mathematicae Applicatae Sinica*, *19*(2), 219-228.

Cheng, D. Z., & Zhang, L. J. (2003). On semi-tensor product of matrices and its applications. *Acta Mathematicae Applicatae Sinica*, *19*(2), 219-228.

Fernández, F. M. (2016). The Kronecker product and some of its physical applications. *European Journal of Physics*, *37*(6), 065403(1-11).

Lovász, L. (1972). Normal hypergraphs and the perfect graph conjecture. *discrete Mathematics*, *2*(3), 253-267.

Meng, M., & Feng, J. E. (2014). A matrix approach to hypergraph stable set and coloring problems with its application to storing problem. *Journal of Applied Mathematics*, *2014*(1), 783784 (1-9).

Pickard, J., Chen, C., Stansbury, C., Surana, A., Bloch, A. M., & Rajapakse, I. (2024). Kronecker product of tensors and hypergraphs: structure and dynamics. *SIAM Journal on Matrix Analysis and Applications*, *45*(3), 1621-1642.

Putri, F., F., Triyani & Wardani, A. (2020). Konsep dasar hipergraf dan sifat-sifatnya. *Jurnal Ilmiah Matematika dan Pendidikan Matematika (JMP). 12*(2), 49-62.

Toft, B. (1974). Color-critical graphs and hypergraphs. *Journal of Combinatorial Theory, Series B*, *16*(2), 145-161.

Wanless, I. M., & Wood, D. R. (2022). A general framework for hypergraph coloring. *SIAM Journal on Discrete Mathematics*, *36*(3), 1663-1677.