

PENGLASTERAN GRAF DAN APLIKASINYA

Mokhammad Ridwan Yudhanegara
Universitas Singaperbangsa Karawang
mridwan.yudhanegara@staff.unsika.ac.id

Abstrak

Walaupun penemuannya sudah lama, graf merupakan cabang ilmu terbaru dari matematika. Pokok materi awal dari graf itu sendiri adalah sekumpulan titik yang dihubungkan oleh sisi. Semakin canggihnya teknologi, graf digunakan oleh berbagai cabang ilmu, baik matematika itu sendiri, dan di luar matematika. Di luar matematika, pada bidang teknik, seperti teknik informatika, teknik industri, dll., istilah lain graf adalah jaringan. Pada aplikasinya, lokasi atau tempat sering direpresentasikan oleh titik dan sisi merepresentasikan jalur atau jalan penghubung antar lokasi. Artikel ini membahas contoh kasus pengklasteran graf yang digunakan untuk mempartisi graf menjadi dua subgraf/ kluster, empat kluster, dan delapan kluster.

Kata kunci: pengklasteran, nilai eigen, matriks Laplacian

Pendahuluan

Analisis kluster, selanjutnya disebut pengklasteran bertujuan untuk mengklasifikasikan objek berdasarkan karakteristik antar objek tersebut, sehingga dapat diidentifikasi dengan karakteristik masing-masing kelompok/ kluster (Yudhanegara, dkk., 2020). Pengklasteran graf berguna untuk mengelompokkan titik dalam graf ke dalam beberapa kluster dengan memperhatikan struktur sisi dari jaringan tersebut. Pada umumnya terdapat dua pendekatan dalam mencari subgraf/kluster dari suatu graf/jaringan, yaitu pendekatan dengan informasi geometris, dan pendekatan tanpa informasi geometris (Newman, 2004).

Banyak sekali metode pencarian kluster dari sebuah jaringan menggunakan pendekatan tersebut. Kemudian ada beberapa metode untuk mencari kluster dengan cara membagi jaringan menjadi dua kluster (*bisection methods*), selanjutnya dilakukan secara rekursif untuk menghasilkan lebih banyak kluster yang diinginkan. Beberapa metode *bisection* yang menggunakan pendekatan geometris diantaranya metode dengan *coordinate bisection*, *inertial bisection*, dan *geometric partitioning*.

Metode *spectral bisection*, yaitu metode pencarian kluster dengan cara membagi jaringan menjadi dua kluster. Metode ini tergolong pada pendekatan tanpa informasi geometris, yang tidak dilakukan pada metode-metode *bisection* lainnya yang pada umumnya menggunakan informasi geometris. Pembagian klasternya mengacu pada nilai eigen matriks Laplacian \mathbf{L} dari graf $G = (V, E,)$ dengan $\{e_{ij} = (v_i, v_j): v_i, v_j \in V, i \neq j\}$ $V = \{v_1, v_2, \dots, v_n\}$ dan yaitu

$$\mathbf{L} = \mathbf{Q}\mathbf{Q}^T = \mathbf{D} - \mathbf{A}$$

dengan \mathbf{Q} adalah matriks insidensi yang diberikan orientasi, \mathbf{D} adalah matriks diagonal yang entri diagonalnya adalah derajat titik-titik dan entri yang lainnya nol, dan \mathbf{A} adalah matriks ketetanggan dari graf G .

$$\mathbf{Q} = q_{ij} = \begin{cases} +1, v_i \text{ akhir dari } e_j \\ -1, v_i \text{ akhir dari } e_j \\ 0, \text{ yang lainnya} \end{cases}$$

Sehingga entri matriks

$$\mathbf{L} = l_{ij} = \begin{cases} -1, \text{ jika } v_i \neq v_j \text{ dan } e_{ij} \in E \\ 0, \text{ jika } v_i \neq v_j \text{ dan } e_{ij} \notin E \\ \text{deg}(v_i), \text{ jika } i = j \end{cases}$$

Untuk graf berbobot $G = (V, E, w)$, entri matriks

$$\mathbf{L} = l_{ij} = \begin{cases} -w_e(e_{ij}), \text{ jika } v_i \neq v_j \text{ dan } e_{ij} \in E \\ 0, \text{ jika } v_i \neq v_j \text{ dan } e_{ij} \notin E \\ \sum_{k=1}^n w_e(e_{ik}), \text{ jika } i = j \end{cases}$$

Adapun sifat dari matriks \mathbf{L} (Elsner, 1997) yaitu:

- simetris dan nilai eigen $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ adalah anggota bilangan real dan vektor eigennya anggota bilangan real dan ortogonal,
- semua nilai eigen $\lambda_j, j = 1, 2, \dots, n$ adalah tak negatif
- $\mathbf{1} = (1, 1, \dots, 1)^T$ merupakan vektor eigen dengan nilai eigen $\lambda_1 = 0$,
- multiplisitas aljabar dari nilai eigen $\lambda_1 = 0$ adalah banyaknya komponen dari graf G , sehingga secara khusus:

$$\lambda_2 \neq 0 \leftrightarrow G \text{ terhubung.}$$

Bukti c)

$$\text{Karena } \mathbf{L} \mathbf{1} = \begin{bmatrix} \sum_{i=1}^n l_{ij} \\ \vdots \\ \sum_{i=1}^n l_{ij} \end{bmatrix} = \mathbf{0} \text{ maka } \mathbf{L} \mathbf{1} = \mathbf{0} \mathbf{1}$$

Sehingga $\mathbf{L} \mathbf{1} = \lambda_1 \mathbf{1}, \lambda_1 = 0$

Bukti d)

$\lambda_2 \neq 0 \leftrightarrow G$ terhubung

(\leftarrow) jika G terhubung maka $\lambda_2 \neq 0$

G terhubung jika dan hanya jika \mathbf{L} *irreducible*, dan jika \mathbf{L} *irreducible* maka dapat dibentuk matriks tak negatif $\hat{\mathbf{L}} = 2n\mathbf{I} - \mathbf{L}$, dengan λ_i adalah nilai eigen dari \mathbf{L} dan $\hat{\lambda}_i = 2n - \lambda_i, i = 1, 2, \dots, n$ adalah nilai eigen dari $\hat{\mathbf{L}}$ dan semua nilai eigen dari $\hat{\mathbf{L}}$ adalah positif.

$$\begin{aligned} \hat{\mathbf{L}} \mathbf{x} &= (2n\mathbf{I} - \mathbf{L}) \mathbf{x} \\ &= 2n\mathbf{x} - \mathbf{L} \mathbf{x} = 2n\mathbf{x} - \lambda_i \mathbf{x} \\ &= (2n - \lambda_i) \mathbf{x} \end{aligned}$$

Teorema Peron-Frobenius (Elsner, 1997):

- If \mathbf{B} is positive, then $\rho(\mathbf{B})$ is a simple eigenvalue, greater than the magnitude of any other eigenvalue.
- If $\mathbf{B} \geq \mathbf{0}$ is irreducible, then $\rho(\mathbf{B})$ is a simple eigenvalue, any eigenvalue of \mathbf{B} the same modulus is also simple. \mathbf{B} has a positive eigen vector \mathbf{x} corresponding to $\rho(\mathbf{B})$, and any nonnegative eigenvector of \mathbf{B} is a multiple of \mathbf{x} .



Berdasarkan teorema tersebut $\hat{\mathbf{L}}$ terhubung, sehingga untuk $\hat{\lambda}_n$ adalah *simple eigenvalue* begitu juga dengan λ_1 sehingga $\lambda_2 > \lambda_1 = 0$.

(\rightarrow) jika $\lambda_2 \neq 0$ maka G terhubung,

kontraposisinya jika G tidak terhubung maka $\lambda_2 = 0$.

G tidak terhubung, artinya ada $G_1, G_2, \dots, G_k, k \leq n$ komponen yang terhubung. Sehingga \mathbf{L} mempunyai blok-blok diagonal dari $\mathbf{L}_1, \mathbf{L}_2, \dots, \mathbf{L}_k$. Sehingga

$$\sigma(\mathbf{L}) = \bigcup_{i=1}^k \sigma(\mathbf{L}_i).$$

Karena setiap \mathbf{L}_i pasti memuat nilai eigen yang bernilai 0, maka setidaknya pada \mathbf{L} terdapat k buah nilai eigen bernilai 0. $\lambda_1 = \lambda_2 = \dots = \lambda_k = 0$.

Untuk memperoleh kluster dari suatu jaringan (dalam kasus ini graf G) dilakukan dengan menggunakan algoritma *spectral bisection* yaitu:

- menentukan matrik Laplacian \mathbf{L} dari graf G ,
- mencari vektor eigen \mathbf{u} dari nilai eigen terkecil kedua λ_2 ,
- menghitung median $m_{\mathbf{u}}$ dari semua elemen-elemen \mathbf{u} ,
- memilih $V_1 = \{v_i \in V: \mathbf{u}_i < m_{\mathbf{u}}\}$ dan $V_2 = \{v_i \in V: \mathbf{u}_i > m_{\mathbf{u}}\}$ dan jika beberapa elemen \mathbf{u} sama dengan $m_{\mathbf{u}}$, maka distribusikan titik yang sesuai sehingga seimbang. Langkah d) didasarkan pada Teorema Elsner (1997).

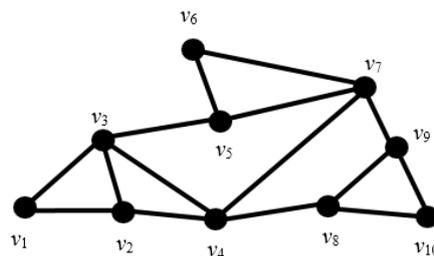
Teorema Elsner (1997)

Misal $G = (V, E), V = \{v_1, v_2, \dots, v_n\}$ adalah graf terhubung dan \mathbf{u} adalah vektor eigen yang bersesuaian. Untuk $r \geq 0, r \in \mathbb{R}$, definisikan $V_1 = \{v_i \in V: \mathbf{u}_i \geq -r\}$ maka V_1 subgraf terhubung, dan Untuk $r \leq 0, r \in \mathbb{R}$, definisikan $V_2 = \{v_i \in V: \mathbf{u}_i \leq -r\}$ maka V_2 subgraf terhubung.

Kelemahan dari metode *spectral bisection* adalah hanya membagi jaringan menjadi dua kluster. Pembagian menjadi lebih banyak kluster biasanya dilakukan dengan melakukan pembagian berulang, tetapi ini tidak selalu memberikan hasil yang memuaskan. Jika hal tersebut terjadi, secara umum kita tidak tahu sebelumnya berapa banyak kluster peroleh. Selain itu, jika jenis grafnya bukan graf komplit, memungkinkan terjadi kluster yang tidak sesuai harapan.

Contoh:

Misalkan kita punya graf G yang akan dicari klasternya



Gambar 1. Graf G

Matriks **A** dari graf G yaitu

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

sedangkan matriks **D** dari graf G yaitu

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

dan matriks **L** sebagai berikut

$$\begin{bmatrix} 2 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 4 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 4 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 3 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 4 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}$$

Kemudian diperoleh polinomial karakteristik dari matriks **L** dengan mencari determinan $\lambda \mathbf{I} - \mathbf{A}$, yaitu

$$\det(\lambda \mathbf{I} - \mathbf{A}) = 0$$

$$\lambda^{10} - 30\lambda^9 - 387\lambda^8 - 2806\lambda^7 + 12530\lambda^6 - 35444\lambda^5 + 62767\lambda^4 - 65964\lambda^3 + 36488\lambda^2 - 7970\lambda = 0.$$

Sehingga diperoleh nilai eigen

$$\lambda_1 = 0.0000, \lambda_2 = 0.6046, \lambda_3 = 0.9345, \lambda_4 = 2.5849, \lambda_5 = 3.2364, \lambda_6 = 3.3138, \lambda_7 = 3.7249, \lambda_8 = 4.4494, \lambda_9 = 4.9537, \lambda_{10} = 6.1978,$$



dan vektor eigen bersesuaian

| | u | | | | | | | | |
|---|----------------|---------|---------|---------|---------|---------|---------|---------|---------|
| 1 | -0.4448 | -0.3756 | 0.5629 | -0.3303 | 0.1012 | 0.2453 | 0.1954 | -0.1131 | -0.1172 |
| 1 | -0.333 | -0.2842 | -0.2071 | 0.1179 | -0.3127 | -0.4888 | -0.4057 | -0.376 | 0.094 |
| 1 | -0.2877 | -0.116 | -0.1221 | 0.2904 | 0.1797 | 0.0658 | -0.073 | 0.7101 | 0.3979 |
| 1 | -0.0651 | -0.0954 | -0.5268 | 0.012 | -0.1828 | 0.0433 | 0.4655 | 0.1377 | -0.5814 |
| 1 | -0.134 | 0.3996 | -0.0018 | 0.4222 | 0.5176 | 0.2183 | -0.2225 | -0.3257 | -0.27 |
| 1 | -0.0699 | 0.6492 | 0.2879 | -0.1351 | -0.5591 | 0.0078 | -0.1194 | 0.2045 | -0.0611 |
| 1 | 0.0365 | 0.2921 | -0.1666 | -0.2552 | 0.2169 | -0.2319 | 0.5148 | -0.2782 | 0.5265 |
| 1 | 0.3631 | -0.1844 | -0.2496 | -0.1441 | -0.2094 | 0.6669 | -0.2454 | -0.1872 | 0.2594 |
| 1 | 0.3931 | -0.058 | 0.005 | -0.4939 | 0.3732 | -0.3333 | -0.3551 | 0.2489 | -0.2447 |
| 1 | 0.5419 | -0.2275 | 0.4182 | 0.516 | -0.1246 | -0.1934 | 0.2451 | -0.0209 | -0.0035 |

Untuk $\lambda_2 = 0.6046$

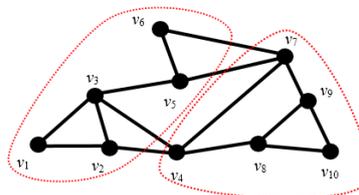
$$\mathbf{u} = [-0.4448 \quad -0.3330 \quad -0.2877 \quad -0.0651 \quad -0.134 \\ -0.0699 \quad 0.0365 \quad 0.3631 \quad 0.3931 \quad 0.5419]^T$$

$$m_{\mathbf{u}} = -0.0675$$

$$V_1 = \{v_1, v_2, v_3, v_5, v_6\}$$

$$V_2 = \{v_4, v_7, v_8, v_9, v_{10}\}.$$

Kalster yang dihasilkan terlihat pada Gambar 2.



Gambar 2. Graf G dengan dua kluster

1. Simulasi

Misalkan kita menganggap bahwa graf adalah sebuah jaringan pendistribusian barang yang merupakan peta lokasi yang dimiliki oleh perusahaan jasa pengiriman barang. Simulasi yang dilakukan yaitu pengklasteran jaringan pendistribusian dengan terlebih dahulu menentukan algoritma pengklasteran dengan metode *spectral bisection* secara rekursif. Setelah itu, menentukan banyaknya kluster yang dihasilkan dari algoritma tersebut sebanyak $2^k, k = 1, 2, 3$, dan membuat programnya dengan perangkat lunak Python. Selanjutnya dilakukan pengklasteran lokasi pada jaringan pendistribusian.

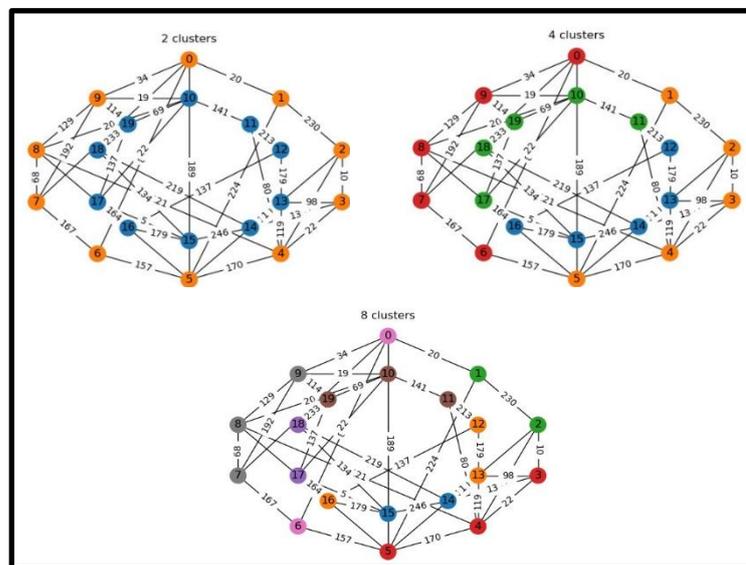
Pada suatu jaringan pendistribusian, diketahui bahwa setiap ruas jalan memiliki volum kendaraan yang berbeda. Kondisi volum kendaraan selalu berubah tiap waktunya. Hal ini memberikan gambaran bahwa jaringan pendistribusian tersebut memiliki bobot sisi yang berubah-ubah. Kemudian, diasumsikan ada 5000 unit kendaraan yang melewati ruas-ruas jalan pada tiap waktu t di kota A. Misal \mathbf{X}_t adalah vektor acak yang menyatakan banyaknya kendaraan yang melewati ruas-ruas jalan pada tiap waktu t , dengan X_{it} adalah peubah acak yang menyatakan banyaknya kendaraan yang melewati ruas jalan ke- i pada

waktu ke- t . Pada jaringan pendistribusian, lokasi tujuan direpresentasikan sebagai titik dan ruas jalan direpresentasikan sebagai sisi.

Selanjutnya, dilakukan pengklasteran lokasi pada jaringan pendistribusian tersebut dengan menggunakan metode *spectral bisection* secara rekursif. Pada simulasi ini, jaringan terdiri dari 20 titik ($V = \{v_0 = 0, v_1 = 1, v_2 = 2, \dots, v_{19} = 19\}$) dan sisi terdiri dari 45 sisi ($E = \{e_1, e_2, \dots, e_{45}\}$). Pada kasus ini tampilan titik diberi label 0, 1, 2, ..., 19, dan sisi diberi label banyaknya mobil yang melewati sisi pada waktu t seperti pada Gambar 3. Adapun fungsi kepadatan peluang dari $\mathbf{X}_t \sim \text{Multinomial}(\theta_1, \theta_2, \dots, \theta_{45}, 5000)$ yaitu

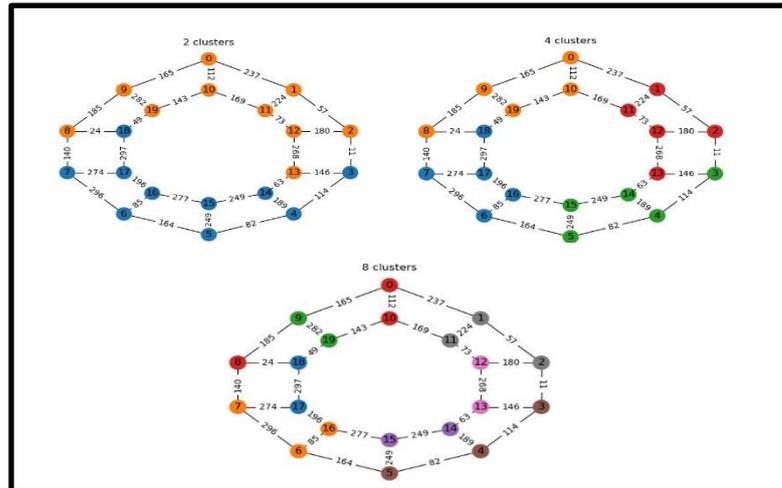
$$p(\mathbf{x}; \boldsymbol{\theta}) = \frac{5000!}{\prod_{i=1}^{45} x_i!} \prod_{i=1}^{45} \theta_i^{x_i}.$$

Dengan menentukan nilai $\boldsymbol{\theta}$ secara acak, hasil kluster dapat dilihat pada Gambar 4.



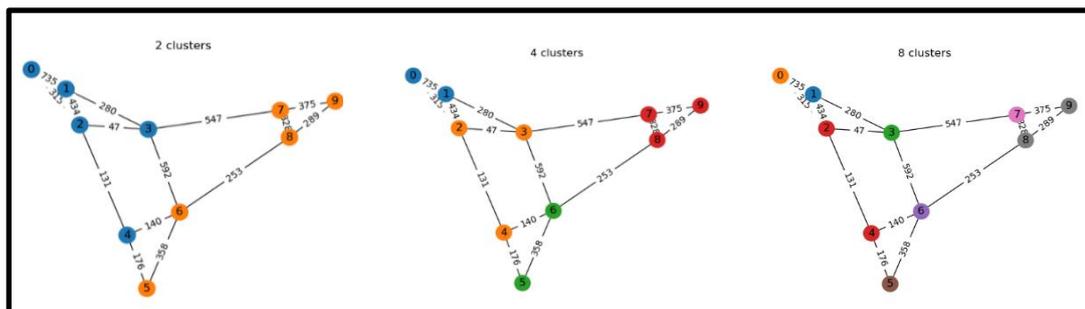
Gambar 4. Hasil pengklasteran jaringan

Misalkan jika jaringan pendistribusian lain terdiri dari 20 titik ($V = \{v_0 = 0, v_1 = 1, v_2 = 2, \dots, v_{19} = 19\}$) dan sisi terdiri dari 30 sisi ($E = \{e_1, e_2, \dots, e_{30}\}$), dengan $\mathbf{X}_t \sim \text{Multinomial}(\theta_1, \theta_2, \dots, \theta_{30}, 5000)$ dan menentukan nilai $\boldsymbol{\theta}$ secara acak, maka hasilnya dapat dilihat pada Gambar 5.



Gambar 5. Hasil pengklasteran jaringan

Kemudian, jika jaringan pendistribusian lain terdiri dari 10 titik ($V = \{v_0 = 0, v_1 = 1, v_2 = 2, \dots, v_{10} = 9\}$) dan sisi terdiri dari 15 sisi ($E = \{e_1, e_2, \dots, e_{15}\}$), dengan $\mathbf{X}_t \sim \text{Multinomial}(\theta_1, \theta_2, \dots, \theta_{15}, 5000)$ dan dengan menentukan nilai θ secara acak, maka hasilnya dapat dilihat pada Gambar 6.



Gambar 6. Hasil pengklasteran jaringan

Kita melakukan pengklasteran pada jaringan pendistribusian barang pada Gambar 4, Gambar 5, dan Gambar 6 dengan tujuan bahwa barang-barang dengan tujuan alamat pada lokasi kluster yang sama akan dikirim oleh kendaraan pengantar yang sama.

2. Kesimpulan

Pengklasteran graf menggunakan metode *spectral bisection* secara rekursif dapat memudahkan kita dalam mengklaster graf menjadi $2^k, k = 1, 2, \dots, n$ subgraf/kluster. Sebagai aplikasinya, kita dapat mengklaster sebuah peta lokasi secara nyata. Pengklasteran graf itu sendiri membutuhkan bidang ilmu matematika lainnya. Seperti aljabar, yang telah diuraikan pada artikel ini yaitu peranan nilai eigen terkecil kedua dari matriks Laplacian dalam menentukan. Kemudian statistika, yaitu peranan sebuah distribusi data dalam membantu dalam proses simulasi untuk menentukan bobot graf dan pemrograman komputer untuk mempermudah proses simulasi. Adapun salah satu manfaat dari pengklasteran graf itu sendiri pada bidang transportasi/jasa pengiriman barang yaitu untuk menentukan kelompok lokasi dari jaringan pendistribusian barang.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Sapto Wahyu Indratno, Ph.D (dosen Matematika Institut Teknologi Bandung) dan Dr. R. R. Kurnia Novita Sari (dosen Matematika Institut Teknologi Bandung) yang memberikan banyak kontribusi pada artikel ini.

Daftar Pustaka

- Elsner, U. 1997. *Graph partitioning*. Chemnitz: Technische Universität Chemnitz
- Newman, M. E. J. 2004. Detecting community structure in networks. *The European Physical Journal B*, 38, 321-30.
- Yudhanegara, M. R., Indratno, S. W. dan Sari, R. R. K. N. 2019. Pengaruh Perubahan Bobot Busur pada Pengklasteran Jaringan Pendistribusian. *Prosiding Seminar Nasional Statistika Universitas Padjajaran VIII (SNS) 2019*, 8, 1-10.
- Yudhanegara, M. R., Indratno, S. W. dan Sari, R. R. K. N. 2020. Clustering for items delivery using rule-k-means. *Journal of the Indonesian Mathematical Society*, 26(2), 185-191.