

## KLASTERISASI GRAF MENGGUNAKAN METODE MARKOV CLUSTER ALGORITHM (MCL)

*Desti Riminarsih*<sup>1</sup>  
*Ilmiyati Sari*<sup>2</sup>  
*Feni Andriani*<sup>3</sup>

<sup>1,2,3</sup> *Fakultas Teknologi Industri, Universitas Gunadarma  
Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat*

<sup>1</sup>[destimath@staff.gunadarma.ac.id](mailto:destimath@staff.gunadarma.ac.id)

<sup>2</sup>[ilmiyati@staff.gunadarma.ac.id](mailto:ilmiyati@staff.gunadarma.ac.id)

<sup>3</sup>[feni.andriani@staff.gunadarma.ac.id](mailto:feni.andriani@staff.gunadarma.ac.id)

### Abstrak

*Analisis kluster merupakan salah satu teknik yang banyak digunakan untuk mengenali kelompok alami dalam suatu kelas entitas. Salah satu metode analisis kluster pada graf adalah metode MCL. Pada penelitian ini dijelaskan algoritma MCL dan contoh penerapan pada klusterisasi graf. MCL merupakan suatu simulasi aliran pada random walk hingga ditemukannya kluster. Algoritma diawali dengan menentukan representasi graf dalam bentuk associated matrix. Langkah berikutnya adalah menentukan matriks Markov berdasarkan pada associated matrix. Tahap berikutnya adalah operasi ekspansi pada matriks Markov yang dilanjutkan dengan operasi inflasi pada matriks hasil ekspansi. Langkah berikutnya adalah mengecek apakah sudah sesuai dengan kriteria konvergensi. Jika kondisi konvergensi belum tercapai maka proses diulang kembali dari tahap operasi ekspansi matriks.*

**Kata Kunci:** *Graf, Klusterisasi, Markov Chain, Matriks, Operator Inflasi, Random Walk.*

## GRAPH CLUSTERING WITH MARKOV CLUSTER ALGORITHM (MCL) METHOD

### Abstract

*Cluster analysis is one of the most widely used techniques for recognizing natural groups within an entity class. One method of cluster analysis in graph is the MCL method. In this study, the MCL algorithm was described and the examples of application in clustering graph. MCL is a flow simulation on a random walk until the discovery of the cluster. The algorithm begins by determining the representation of the graph in the form of an associated matrix. The next step is to determine the Markov matrix based on the associated matrix. The next stage is the expansion operation on the Markov matrix followed by the inflation operation on the expansion matrix. The next step is to check whether it is in line with the convergence criteria. If the convergence condition has not been reached then the process is repeated from the phase of expansion operation of the matrix.*

**Keywords:** *Clustering, Graph, Inflation Operator, Markov Chain, Matrix, Random Walk.*

## PENDAHULUAN

Data *mining* adalah proses untuk menemukan informasi yang berguna secara otomatis di repositori data yang besar. Teknik penambangan data dikerahkan untuk menjelajahi *database* besar untuk menemukan pola baru dan berguna yang mungkin tidak diketahui [1]. Salah satu teknik dalam data *mining* adalah analisis *cluster* yang merupakan studi matematis mengenai metode untuk mengenali kelompok alami dalam suatu kelas entitas [2]. Analisis *cluster* mengelompokkan objek data hanya berdasarkan informasi yang terdapat pada data yang menggambarkan objek dan hubungannya. Tujuannya adalah bahwa objek dalam suatu kelompok serupa satu sama lain dan berbeda dari objek dalam kelompok lain. Semakin besar kesamaan (atau homogenitas) dalam suatu kelompok dan semakin besar perbedaan antar kelompok, semakin baik atau lebih berbeda pengelompokannya. Seluruh koleksi *cluster* biasanya disebut sebagai *clustering* [1].

Analisis *cluster* memberikan abstraksi dari objek data individual ke kelompok tempat objek data berada. Selain itu, beberapa teknik pengelompokan mengkarakterisasi setiap cluster dalam hal prototipe cluster yang berupa objek data yang mewakili objek lain di cluster. Prototip cluster ini dapat digunakan sebagai dasar untuk sejumlah analisis data atau teknik pengolahan data. Oleh karena itu, dalam konteks utilitas, analisis cluster adalah studi teknik untuk menemukan prototip cluster yang paling representative terhadap data [1].

Aplikasi-aplikasi dalam analisis cluster cukup banyak dan telah menghasilkan banyak metode yang membingungkan [2]. Pada umumnya aplikasi klasik memiliki satu kesamaan yaitu adanya anggapan bahwa entitas diwakili oleh vektor, yang menggambarkan bagai-

mana masing-masing entitas mendapatkan nilai pada serangkaian karakteristik atau fitur. Perbedaan antara dua entitas dihitung sebagai jarak antara masing-masing vektor yang menggambarkannya. Ketidaksamaan dari dua vector sesuai dengan jarak dalam geometri Euclidean yang sangat eksplisit [2].

Graf adalah objek yang memiliki sifat kombinatorial yang jauh lebih banyak yang tercermin pada beberapa istilah seperti simpul, ruas derajat (jumlah tetangga), jalur, siklus, keterhubungan, dan sebagainya. Bobot ruas biasanya tidak sesuai dengan jarak atau kedekatan yang dapat disematkan dalam geometri Euclidean, juga tidak memerlukannya menyerupai metrik [2]. Beberapa algoritma *clustering* telah dikembangkan salah satunya adalah algoritma *clustering* berbasis graf yang terdiri dari 4 macam yaitu MCODE, RNSC (*Restricted Neighborhood Search Clustering*), SPC (*Super Paramagnetic Clustering*) dan MCL (*Markov Clustering*) [3].

Analisis *cluster* berbasis graf erat kaitannya dengan bidang partisi graf, dimana metode ini digunakan untuk menemukan partisi optimal dari sebuah graf dengan batasan tertentu. Partisi didefinisikan secara ketat sebagai pembagian beberapa  $S$  menjadi himpunan bagian yang memenuhi asumsi semua pasang himpunan bagian dipisahkan dan penyatuan semua subset menghasilkan  $S$ .

Pada partisi graf, ukuran partisi yang dibutuhkan ditentukan terlebih dahulu. Tujuannya adalah untuk meminimalkan beberapa fungsi biaya yang terkait dengan hubungan yang menghubungkan elemen partisi yang berbeda [2]. Hal ini merupakan perbedaan mendasar antara graf partisi dan *graph clustering* dimana pada graf *clustering* penemuan kelompok-kelompok dilakukan secara alami. Pada penelitian ini

dijelaskan metode clustering pada graf menggunakan algoritma MCL.

## METODE PENELITIAN

Pada bagian ini dijelaskan tahapan klasterisasi graf dengan menggunakan metode MCL. Konsep utama klasterisasi pada graf adalah simpul terhubung tinggi bisa berada dalam satu klaster sedangkan simpul terhubung rendah bisa berada dalam kelompok yang berbeda [2]. Jumlah jalur  $u-v$  dengan panjang  $k$  lebih besar jika  $u, v$  berada pada cluster padat yang sama, dan lebih kecil jika berada pada kelompok yang berbeda. Suatu *random walk* pada graf akan meninggalkan klaster sampai simpul-simpulnya telah dikunjungi.

### Random Walk

*Random walk* pada graf terdiri dari urutan simpul yang dihasilkan dari simpul awal dengan memilih ruas, melewati ruas menuju simpul baru dan mengulangi prosesnya [4]. *Random walk* bisa dimulai dari simpul manapun. Dimulai pada simpul  $r$ , jika *random walk* akan mencapai simpul  $t$  dengan probabilitas tinggi, maka  $r$  dan  $t$  harus dikelompokkan bersama. Berdasarkan sebuah graf, terdapat banyak tautan dalam sebuah klaster dan lebih sedikit hubungan antar klaster. Jika dimulai dari sebuah simpul, dan kemudian melakukan *random walk* ke simpul yang terhubung, kemungkinan lebih cenderung untuk tinggal di dalam sebuah klaster daripada melakukan *walk* di antara keduanya. Hal ini merupakan dasar dari algoritma MCL. *Random walk* pada graf dapat digunakan untuk menentukan keberadaan suatu kelompok yang ditandai dengan kecenderungan berkumpulnya suatu aliran. *Random walk* pada graf dihitung dengan menggunakan "*Markov Chains*"[5].

### Markov Chain

*Markov Chain* merupakan suatu

barisan variable  $x_1, x_2, x_3$  dan seterusnya dimana diberikan stata saat ini, stata yang sebelumnya dan stata berikutnya yang bersifat independen. Probabilitas untuk langkah waktu berikutnya hanya bergantung pada probabilitas saat ini. *Random walk* merupakan suatu contoh *Markov Chain* dengan menggunakan matriks transisi probabilitas [5].

### Operator Inflasi

Definisi 1.

Diberikan matriks  $M \in R^{n \times 1}$ ,  $M \geq 0$ , bilangan real nonnegative  $r$ , matriks hasil penskalaan kembali setiap kolom  $M$  dengan koefisien pangkat  $r$  disebut  $\Gamma_r M$ , dan  $\Gamma_r$  disebut operator inflasi dengan koefisien pangkat  $r$ . Secara formal,  $\Gamma_r : R^{n \times 1} \rightarrow R^{n \times 1}$  didefinisikan dengan

$$(\Gamma_r M)_{pq} = (M_{pq})^r / \sum_{i=1}^k (M_{iq})^r \quad (1)$$

Jika subscript dihilangkan maka pada persamaan (1) nilai koefisien pangkat ( $r$ ) adalah 2.

### MCL

Pada metode MCL simulasi aliran pada *random walk* merupakan konsep inti hingga ditemukannya klaster. Aliran lebih mudah berada di dalam wilayah padat daripada di batas yang jarang, namun dalam jangka panjang efek ini akan lenyap. Selama kekuatan sebelumnya dari *Markov Chain*, bobot ruas akan lebih tinggi pada mata rantai yang berada di dalam cluster, dan lebih rendah di antara kelompok-kelompok. Hal ini berarti terdapat korespondensi antara distribusi bobot di seluruh kolom dan *clustering*. MCL dengan sengaja meningkatkan pengaruh dengan cara menghentikan sebagian dalam *Markov Chain* kemudian menyesuaikan transisi menurut kolom. Untuk setiap simpul pada graf, nilai transisi diubah sehingga mengakibatkan tetangga yang kuat diperkuat lebih lanjut sedangkan tetangga yang kurang populer diturunkan. Penyesuaian ini bisa dila-



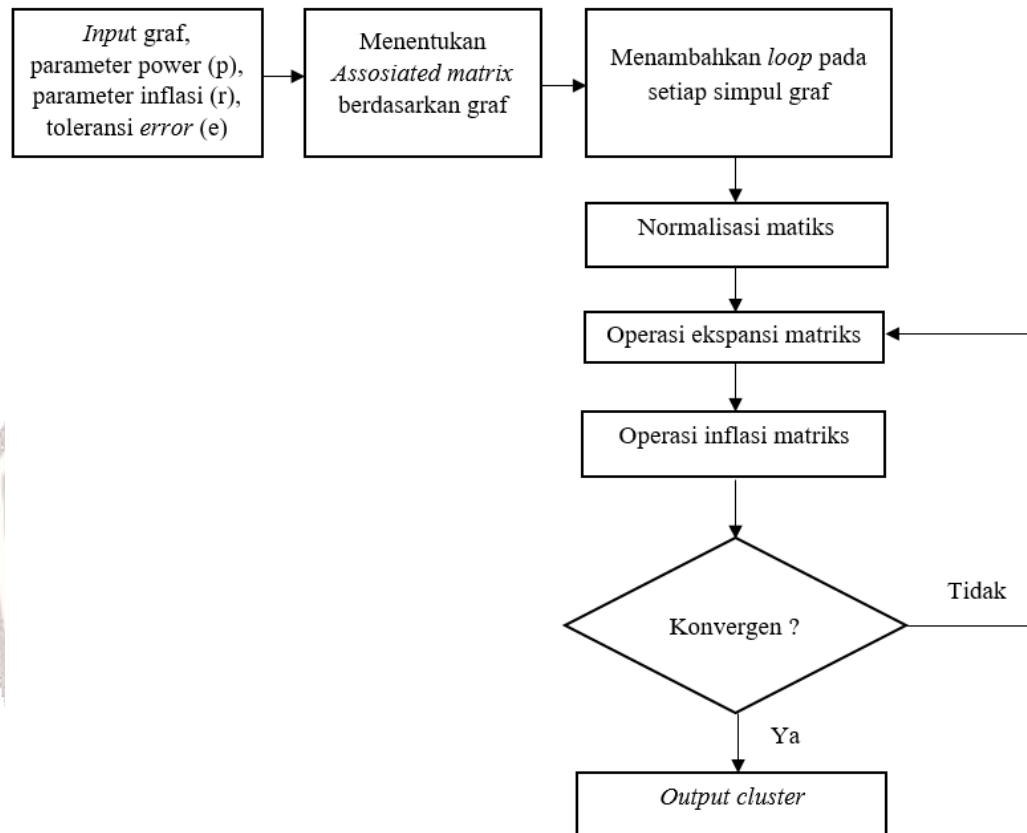
kukan dengan mengangkat satu kolom ke kekuatan non-negatif, dan kemudian kembali normal [5].

### Algoritma MCL

*Markov Clustering Algorithm* (MCL) adalah algoritma cluster yang

cepat untuk graf yang didasarkan pada simulasi aliran (*flows*) pada graf. MCL dapat pula dilakukan pada graf yang terboboti.

Tahapan dalam melakukan klasterisasi graf disajikan pada Gambar 1.



Gambar 1. Tahapan Algoritma MCL

Berdasarkan Gambar 1, penjelasan tahapan algoritma MCL [6, 7] sebagai berikut:

1. *input* graf yang akan diklasterisasi, nilai parameter *power* yaitu  $e$ , nilai *parameter inflasi* yaitu  $r$  dan nilai toleransi *error*.
2. Buatlah *associated matrix* ( $M_G$ ) yang bersesuaian dengan graf pada langkah 1. *Associated matrix* disebut juga dengan matriks adjasensi yaitu matriks yang kolom dan barisnya merepresentasikan simpul pada graf, misalnya baris dan kolom 1 merepsentasikan simpul 1, se-

dangkan busurnya direpresentasikan oleh entri-entri pada *associated matrix* tersebut, dimana simpul-simpul yang dihubungkan dengan sebuah busur diberi entri 1 dan yang tidak berhubungan diberi entri 0.

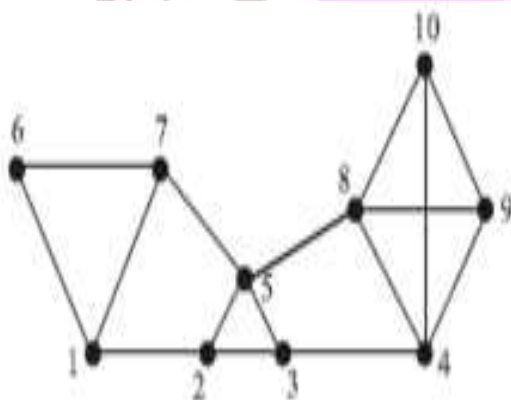
3. Tambahkan *loop* pada setiap simpul atau dengan kata lain *associated matrix* yang diperoleh pada langkah 2 ditambahkan dengan matriks identitas  $I_n$ .
4. Normalisasi matriks hasil pada langkah 3. Pada langkah ini, setiap entri matriks dibagi dengan jumlah entri matriks di kolom

tersebut sehingga diperoleh matriks dengan jumlah entri dari setiap kolom adalah 1.

5. Pada tahap ini dilakukan operasi ekspansi matriks yang dilakukan dengan memangkatkan matriks yang diperoleh pada langkah 4 dengan pangkat  $e$ , dalam hal ini  $e = 2$ .
6. Pada tahap ini dilakukan operasi inflasi matriks hasil proses ekspansi (matriks yang diperoleh pada langkah 5) dengan parameter inflasi  $r$ .
7. Suatu matriks hasil dari langkah 6 disebut konvergen jika nilai residual energy lebih kecil daripada toleransi *error* maka proses berhenti dan diperoleh matriks yang menunjukkan klusterisasi. Jika kondisi tersebut belum dicapai maka ulangi kembali langkah 5 dan 6 hingga diperoleh kondisi konvergen.
8. Interpretasikan matriks pada langkah 7 sehingga ditemukan *cluster-cluster* dari graf.

### HASIL DAN PEMBAHASAN

Pada bagian ini dijelaskan implementasi metode MCL pada graf berikut:



Gambar 2. Graf

Langkah pertama adalah membaca graf, menentukan parameter ekspansi ( $p$ ), parameter inflasi ( $r$ ), dan toleransi *error* ( $e$ ). Graf yang diklusterisasi adalah graf

yang ditunjukkan pada Gambar 2. Nilai parameter yang diambil adalah  $p = 2$ ,  $r = 2$ , dan  $e = 0.001$ .

Berdasarkan pada Gambar 2, *associated matrix* ( $M_G$ ) adalah

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

Langkah selanjutnya adalah menambahkan *loop* pada graf dengan cara menambahkan nilai 1 pada setiap entri diagonal dari  $M_G$  sehingga diperoleh matriks berikut:

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

Pada proses normalisasi matriks terlebih dahulu ditentukan jumlah tiap kolom. Misal pada kolom 1 dari matriks diatas,

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

Jumlah kolom 1 = 4, sehingga hasil normalisasi kolom 1 adalah

$$\frac{1}{4} \times \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.25 \\ 0.25 \\ 0 \\ 0 \\ 0 \\ 0.25 \\ 0.25 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Sehingga matriks hasil proses normalisa-

si tertera pada gambar 3.

0.25	0.25	0	0	0	0.33	0.25	0	0	0
0.25	0.25	0.25	0	0.2	0	0	0	0	0
0	0.25	0.25	0.2	0.2	0	0	0	0	0
0	0	0.25	0.2	0	0	0	0.17	0.25	0.25
0	0.25	0.25	0	0.2	0	0.25	0.17	0	0
0.25	0	0	0	0	0.33	0.25	0.17	0	0
0.25	0	0	0	0.2	0.33	0.25	0	0	0
0	0	0	0.2	0.2	0	0	0.17	0.25	0.25
0	0	0	0.2	0	0	0	0.17	0.25	0.25
0	0	0	0.2	0	0	0	0.17	0.25	0.25

Gambar 3. Hasil Poses Normalisasi

Langkah berikutnya adalah melakukan operasi ekspansi matriks. Matriks hasil normalisasi diekspansi dengan cara

dipangkatkan 2, sehingga diperoleh hasil seperti pada gambar 4.

0.27	0.125	0.0625	0	0.1	0.2739	0.2075	0.0561	0	0
0.125	0.2375	0.175	0.05	0.14	0.0825	0.1125	0.034	0	0
0.0625	0.175	0.225	0.09	0.14	0	0.05	0.068	0.05	0.05
0	0.0625	0.1125	0.224	0.084	0	0	0.1479	0.2175	0.2175
0.125	0.175	0.175	0.084	0.224	0.0825	0.1125	0.0629	0.0425	0.0425
0.2075	0.0625	0	0.034	0.084	0.2739	0.2075	0.085	0.0425	0.0425
0.2075	0.1125	0.05	0	0.09	0.2739	0.2575	0.0901	0	0
0	0.05	0.1	0.174	0.074	0	0.05	0.1819	0.2175	0.2175
0	0	0.05	0.174	0.034	0	0	0.1479	0.2175	0.2175
0	0	0.05	0.174	0.034	0	0	0.1479	0.2175	0.2175

Gambar 4. Hasil operasi Ekspansi Matriks

Langkah selanjutnya adalah dilakukan operasi inflasi matriks hasil proses ekspansi dengan parameter inflasi r

= 2. Gambar 5 adalah matriks hasil operasi inflasi.

0.383155	0.107066	0	0	0.100628	0.333333	0.242255	0	0	0
0.082123	0.38651	0.227642	0	0.197231	0	0.07121	0	0	0
0	0.20985	0.376307	0	0.197231	0	0	0	0	0
0	0	0.094077	0.355848	0	0	0	0.221601	0.25	0.25
0.082123	0.20985	0.227642	0	0.504911	0	0.07121	0	0	0
0.226299	0	0	0	0	0.333333	0.242255	0	0	0
0.226299	0.086724	0	0	0	0.333333	0.37307	0	0	0
0	0	0.074332	0.214717	0	0	0	0.335197	0.25	0.25
0	0	0	0.214717	0	0	0	0.221601	0.25	0.25
0	0	0	0.214717	0	0	0	0.221601	0.25	0.25

Gambar 5. Matriks hasil operasi inflasi.

Selanjutnya menentukan apakah matriks hasil operasi inflasi sudah memenuhi kriteria konvergen. Pada iterasi pertama ini diperoleh nilai residual energi: 0.162050 sehingga proses diulangi lagi

dari tahap 5 hingga diperoleh kondisi stabil.

Berdasarkan Tabel 1 nilai residual energi mulai menurun pada iterasi ke-3 hingga ke-9. Pada iterasi ke-9 nilai

residual energi < 0.001 sehingga iterasi berhenti pada iterasi ke-9 merupakan matriks yang menunjukkan klaster-klaster dari graf.

Iterasi ke-i	Residual energi
5	0.127801
6	0.124785
7	0.085941
8	0.014665
9	0.000000

**Tabel 1. Residual Energi**

Iterasi ke-i	Residual energi
1	0.162050
2	0.156611
3	0.161669
4	0.137281

Klasterisasi berhasil ditentukan dengan banyak iterasi sebanyak 9. Gambar 6 adalah matriks hasil operasi inflasi dari iterasi ke-2 hingga ke-9.

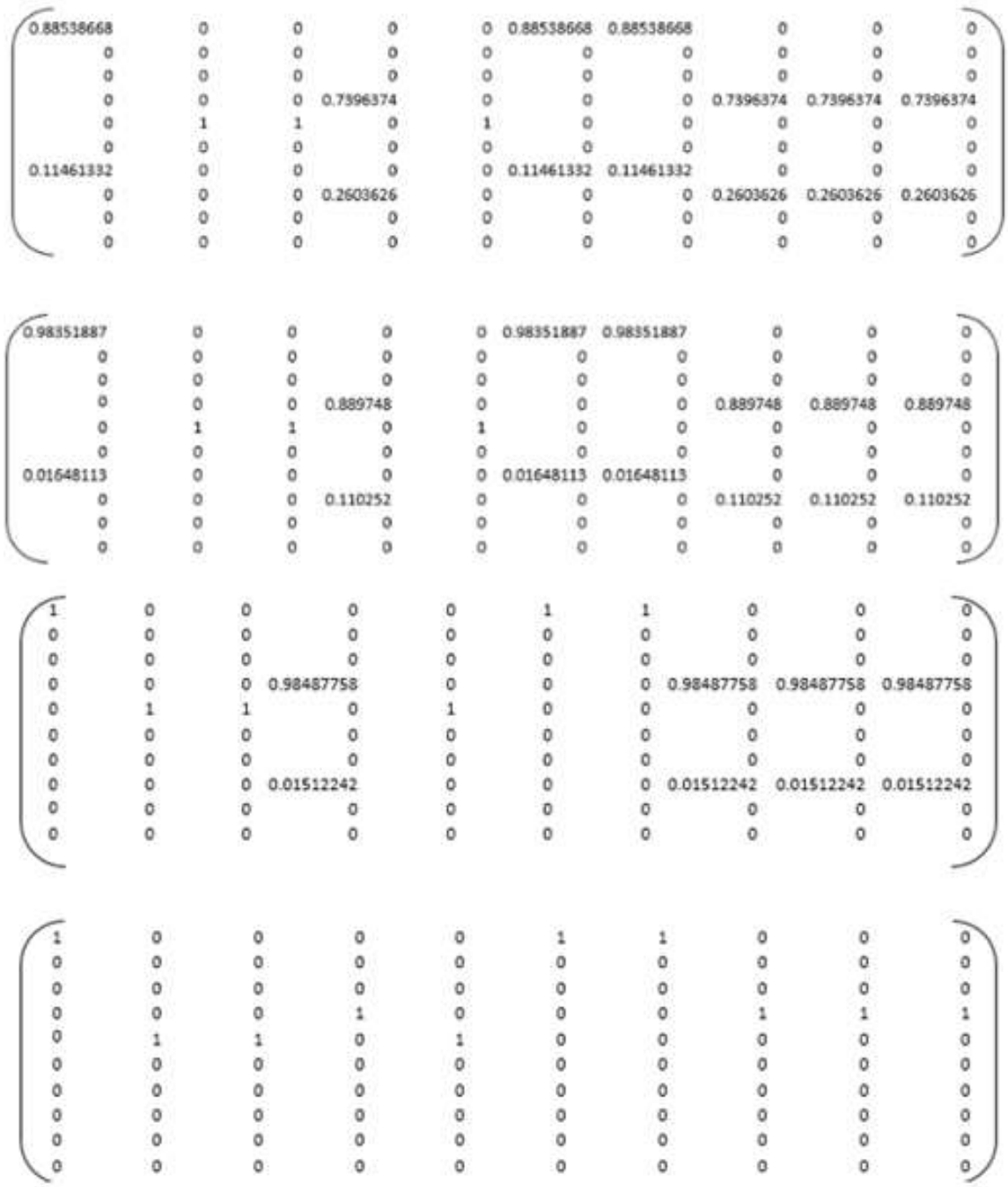
$$\begin{pmatrix} 0.41344614 & 0.08485014 & 0 & 0 & 0.05236746 & 0.37793296 & 0.37412911 & 0 & 0 & 0 \\ 0 & 0.35165134 & 0.29063989 & 0 & 0.22494272 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.22207486 & 0.33397817 & 0 & 0.19867302 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.31494869 & 0 & 0 & 0 & 0.27729102 & 0.28902457 & 0.28902457 \\ 0.05398453 & 0.34142366 & 0.37538195 & 0 & 0.5240168 & 0 & 0 & 0 & 0 & 0 \\ 0.22497912 & 0 & 0 & 0 & 0 & 0.26438516 & 0.24583975 & 0 & 0 & 0 \\ 0.30759022 & 0 & 0 & 0 & 0 & 0.35768188 & 0.38003114 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25981476 & 0 & 0 & 0 & 0.29176863 & 0.27444122 & 0.27444122 \\ 0 & 0 & 0 & 0.21261828 & 0 & 0 & 0 & 0.21547017 & 0.21826711 & 0.21826711 \\ 0 & 0 & 0 & 0.21261828 & 0 & 0 & 0 & 0.21547017 & 0.21826711 & 0.21826711 \end{pmatrix}$$

$$\begin{pmatrix} 0.47041459 & 0 & 0 & 0 & 0 & 0.45907591 & 0.45831035 & 0 & 0 & 0 \\ 0 & 0.26167624 & 0.25089099 & 0 & 0.21821006 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.18049841 & 0.19584648 & 0 & 0.16339467 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.33944217 & 0 & 0 & 0 & 0.33643552 & 0.33757323 & 0.33757323 \\ 0 & 0.55782534 & 0.55326253 & 0 & 0.61839527 & 0 & 0 & 0 & 0 & 0 \\ 0.17512564 & 0 & 0 & 0 & 0 & 0.17802334 & 0.17753977 & 0 & 0 & 0 \\ 0.35445976 & 0 & 0 & 0 & 0 & 0.36290075 & 0.36414988 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.29527226 & 0 & 0 & 0 & 0.29774726 & 0.29670198 & 0.29670198 \\ 0 & 0 & 0 & 0.18264279 & 0 & 0 & 0 & 0.18290861 & 0.18286239 & 0.18286239 \\ 0 & 0 & 0 & 0.18264279 & 0 & 0 & 0 & 0.18290861 & 0.18286239 & 0.18286239 \end{pmatrix}$$

$$\begin{pmatrix} 0.57336205 & 0 & 0 & 0 & 0 & 0.57308765 & 0.57306782 & 0 & 0 & 0 \\ 0 & 0.12757451 & 0.12769368 & 0 & 0.12412326 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.06943582 & 0.0697324 & 0 & 0.06795701 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.42450699 & 0 & 0 & 0 & 0.42448745 & 0.42449513 & 0.42449513 \\ 0 & 0.80298967 & 0.80257392 & 0 & 0.80791973 & 0 & 0 & 0 & 0 & 0 \\ 0.08290252 & 0 & 0 & 0 & 0 & 0.08293822 & 0.08294031 & 0 & 0 & 0 \\ 0.34373533 & 0 & 0 & 0 & 0 & 0.34397413 & 0.34399187 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.32696765 & 0 & 0 & 0 & 0.32698411 & 0.32697762 & 0.32697762 \\ 0 & 0 & 0 & 0.12426268 & 0 & 0 & 0 & 0.12426422 & 0.12426363 & 0.12426363 \\ 0 & 0 & 0 & 0.12426268 & 0 & 0 & 0 & 0.12426422 & 0.12426363 & 0.12426363 \end{pmatrix}$$

$$\begin{pmatrix} 0.73540683 & 0 & 0 & 0 & 0 & 0.73540669 & 0.73540668 & 0 & 0 & 0 \\ 0 & 0.02336579 & 0.02336645 & 0 & 0.02335809 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.56667181 & 0 & 0 & 0 & 0.56667181 & 0.56667181 & 0.56667181 \\ 0 & 0.97663421 & 0.97663355 & 0 & 0.97664191 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.26459317 & 0 & 0 & 0 & 0 & 0.26459331 & 0.26459332 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.33621066 & 0 & 0 & 0 & 0.33621066 & 0.33621066 & 0.33621066 \\ 0 & 0 & 0 & 0.04855877 & 0 & 0 & 0 & 0.04855877 & 0.04855877 & 0.04855877 \\ 0 & 0 & 0 & 0.04855877 & 0 & 0 & 0 & 0.04855877 & 0.04855877 & 0.04855877 \end{pmatrix}$$





Gambar 6. Matriks Hasil Operasi Inflasi dari Iterasi ke-2 Hingga ke-9

Matriks pada iterasi ke-9 merupakan matriks yang menunjukkan hasil klasterisasi graf. Klaster ditandai dengan baris yang memiliki entri tak nol. Entri matriks yang bernilai 1 pada baris yang sama menunjukkan bahwa entri-entri tersebut berada dalam klaster yang sama. Kolom matriks menunjukkan simpul graf.

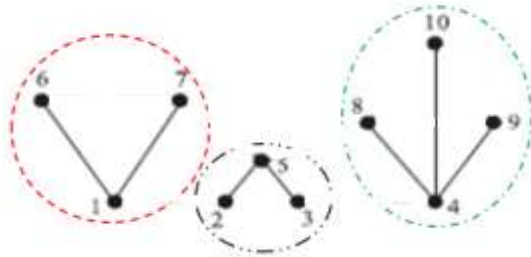
Pada matriks hasil iterasi ke-9 diperoleh bahwa pada baris pertama, entri yang bernilai 1 ada di kolom ke-1, ke-6,

dan ke-7. Pada baris ke-4 diperoleh bahwa entri yang bernilai 1 ada di kolom ke-4, ke-8, ke-9, dan ke-10. Pada baris ke-5 diperoleh bahwa entri yang bernilai 1 ada di kolom ke-2, ke-3, dan ke-5.

Kolom pada matriks merepresentasikan simpul pada graf. Berdasarkan matriks hasil iterasi ke-9 maka dapat disimpulkan bahwa simpul-simpul pada graf dikelompokkan ke dalam 3 klaster. Ketiga klaster tersebut adalah (1,6,7),



(4,8,9,10) dan (2,3,5). Klaster (1,6,7) dengan simpul 1 sebagai pusatnya, klaster {4, 8, 9, 10} dengan simpul 4 sebagai pusatnya, dan klaster {2, 3, 5} dengan simpul 5 sebagai pusatnya.



**Gambar 3. Graf Hasil MCL**

Berdasarkan pada Gambar 3 dapat dilihat hasil klusterisasi metode MCL yaitu graf memiliki 3 klaster yang berbeda.

### **SIMPULAN DAN SARAN**

Algoritma MCL merupakan algoritma yang mudah untuk diterapkan dan memiliki konsep sederhana dengan dua operasi utama yaitu operasi ekspansi dan operasi inflasi. Penentuan klaster berdasarkan algoritma ini dilakukan secara alami. Setiap permasalahan yang dapat direpresentasikan dalam sebuah graf maka proses klusterisasinya dapat menggunakan algoritma MCL.

### **DAFTAR PUSTAKA**

- [1] [Van Dongen, 2000] Van Dongen, S. 2000. Graph Clustering by Flow Simulation. *PhD Thesis*. University of Utrecht, The Netherlands.
- [2] [Tan, Steinbach, dan Kumar, 2006] Tan, P. N., Steinbach, M., dan Kumar, V. 2006. Introduction to Data Mining. Boston: Pearson Education.
- [3] [Brohee dan van Helden, 2006] Brohee, S. dan van Helden, J. 2006. Evaluation of Clustering Algorithms For Protein-Protein Interaction Networks. *BMC Bioinformatics*, Vol. 7, pp. 488.
- [4] [Blum, Hopcroft, dan Kannan, 2015] Blum, A., Hopcroft, J., dan Kannan, R. 2015. Foundations of Data Science. Retrieved from <https://www.cs.cornell.edu/jeh/bookMay2015.pdf>.
- [5] [Macropol, 2009] Macropol, K. 2009. Clustering on Graphs: The Markov Cluster Algorithm (MCL). Retrieved from [https://www.cs.ucsb.edu/%7Eexyan/classes/CS595D2009winter/MCL\\_Presentation2.pdf](https://www.cs.ucsb.edu/%7Eexyan/classes/CS595D2009winter/MCL_Presentation2.pdf).
- [6] [Stijn van Dongen] Stijn van Dongen, MCL - a cluster algorithm for graphs. Retrieved from <http://www.micans.org/mcl/>.
- [7] [Enright, van Dongen, Ouzounis, 2002] Enright, A. J., van Dongen, S., dan Ouzounis, C. A. 2002. An Efficient Algorithm for Large-scale Detection of Protein Families. *Nucleic Acids Research*, Vol. 30, No. 7, pp. 1575 – 1584.