**LAPORAN TUGAS KULIAH COMPUTATION II**



**Penerapan Adjacency Matrix dari Jejaring Sosial**

**untuk Penentuan Kedekatan Profil User**

**DISUSUN OLEH**

**Eryca Dhamma Shanty (535230071)**

**Georgia Sugisandhea (535230080)**

**Brenda Abigail Han’s H (535230194)**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS TARUMANAGARA**

**JUNI 2024**

**DAFTAR ISI**

[**DAFTAR GAMBAR** 4](#_Toc170067315)

[**DAFTAR TABEL** 5](#_Toc170067316)

[**BAB I**  **PENDAHULUAN** 1](#_Toc170067317)

[**1.1.** **Latar Belakang** 1](#_Toc170067318)

[**1.2.** **Tujuan dan Kegunaan Penerapan Teori Matrix Pada Analisis Jejaring Sosial** 2](#_Toc170067320)

[**BAB II**  **KERANGKA TEORI** 3](#_Toc170067321)

[**2.1.** **Graf** 3](#_Toc170067322)

[**BAB III SocioMatrix** 5](#_Toc170067327)

[**3.1.** **Algoritma** 5](#_Toc170067328)

[**3.2.** **Contoh Perhitungan** 6](#_Toc170067329)

[**3.3.** **Implementasi Adjacency Matrix** 14](#_Toc170067341)

[**BAB IV PENUTUP** 20](#_Toc170067353)

[**DAFTAR PUSTAKA** 21](#_Toc170067354)

[**LAMPIRAN A LISTING PROGRAM** 22](#_Toc170067355)

# **DAFTAR GAMBAR**

[**DAFTAR GAMBAR**](#_gjdgxs)

[**BAB I   
PENDAHULUAN**](#_2et92p0) 1

[1.1. Latar Belakang](#_tyjcwt) 1

[Gambar 1.1Contoh Ilustrasi jaringan sosial](#_3dy6vkm) 2

[**BAB II   
KERANGKA TEORI**](#_2s8eyo1) 3

2.1. Graf

[Gambar 2.1 Contoh Ilustrasi simple graph dan unsimple graph.](#_26in1rg) 3

[Gambar 2.2 Contoh Ilustrasi graf dan digraf.](#_lnxbz9) 4

[Gambar 2.3 Contoh Ilustrasi Weighted graf dan Unweighted graf.](#_35nkun2) 4

[**BAB III   
SocioMatrix**](#_1ksv4uv) 5

[3.2. Contoh Perhitungan](#_2jxsxqh) 6

[Gambar 3.1 Graf Hubungan Sosial yang dipakai dalam contoh perhitungan](#_z337ya) 6

[3.3. Implementasi Adjacency Matrix](#_vx1227) 14

[Gambar 3.2 Class MyVertex](#_yb0d8cr22wkk) 15

[Gambar 3.3 Library dan constructor class WeightedRelation](#_kowwphupotdd) 15

[Gambar 3.4 Fungsi addEdge](#_5gi9ghdjcrpi) 16

[Gambar 3.5 Iterasi perhitungan hubungan dalam fungsi Density](#_v0uzj0f2dwuj) 17

[Gambar 3.6 Perhitungan density menggunakan program dan menampilkan hasilnya](#_5j5lobr6blp7) 18

[Gambar 3.7 Inisialisasi individu dalam jejaring sosial](#_f4vpam57qejx) 18

[Gambar 3.8 Inisialisasi jaringan sosial dan memasukkan hubungan antar individu](#_tgx2905ybwu1) 19

[Gambar 3.9 Pemanggilan fungsi ClosenessCentrality dan Density](#_b94p8rxpyf0n) 19

[Gambar 3.10 Output dari fungsi Density terhadap jejaring sosial](#_xu6wyx9xaf2g) 19

# 

# **DAFTAR TABEL**

[**BAB III SocioMatrix** 5](#_Toc170067327)

[**3.2.** **Contoh Perhitungan** 6](#_Toc170067329)

[Tabel 3.1 Adjacency Matrix individu A sampai Z kepada individu A sampai I 7](#_Toc170067331)

[Tabel 3.2 Adjacency Matrix individu A sampai Z kepada individu J sampai R 7](#_Toc170067332)

[Tabel 3.3 *Adjacency Matrix* individu A sampai Z kepada individu S sampai Z 8](#_Toc170067333)

[Tabel 3.4 Jarak *shortest path* dari semua individu A sampai I 9](#_Toc170067334)

[Tabel 3.5 Jarak *shortest path* dari semua individu A sampai Z kepada individu J sampai R 9](#_Toc170067335)

[Tabel 3.6 Jarak *shortest path* dari semua individu A sampai Z kepada individu S sampai Z 10](#_Toc170067336)

[Tabel 3.7 Total jarak *shortest path* masing masing individu 10](#_Toc170067337)

[Tabel 3.8 Perhitungan *closeness* antar individu 11](#_Toc170067338)

[Tabel 3.9 Perhitungan *normalized closeness* antar individu 12](#_Toc170067339)

[Tabel 3.10 Jumlah relasi masing masing individu 13](#_Toc170067340)

**KATA PENGANTAR**

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, kami dapat menyelesaikan laporan tugas yang berjudul “Penerapan Adjacency Matrix dari Jejaring Sosial untuk Penentuan Kedekatan Profil User” untuk meningkatkan akurasi dan efisiensi dalam analisis relasi sosial.Laporan ini disusun sebagai salah satu bentuk tanggung jawab akademik dari tugas akhir dalam mata kuliah Computation II di Universitas Tarumanagara.

Laporan ini bertujuan untuk menjelaskan proses implementasi struktur data Adjacency Matrix dalam menganalisis dan menentukan kedekatan profil pengguna pada jejaring sosial. Adjacency Matrix digunakan untuk merepresentasikan bagaimana hubungan antar pengguna dalam jejaring sosial sehingga dapat identifikasi seberapa dekat pengguna dengan pengguna lainnya. Melalui penggunaan *Adjacency Matrix* juga dapat digunakan untuk mengidentifikasi keterhubungan antar profil,komunitas berdasarkan kedekatan profil.

Kami ingin mengucapkan terimakasih untuk semua pihak yang telah membantu membimbing selama proses penyusunan laporan ini. Ucapan terima kasih khusus kami sampaikan kepada: Lely Hiryanto ST.,MSc.,Ph.D dan kepada rekan-rekan satu tim yang telah bekerja sama dengan baik, saling membantu dan berdedikasi tinggi dalam menyelesaikan tugas ini.

Kami sangat terbuka terhadap saran dan kritik yang membangun demi perbaikan yang akan datang. Semoga laporan ini dapat memberikan manfaat dan kontribusi positif bagi semua pihak yang berkepentingan. Akhir kata, kami berharap laporan ini dapat memberikan wawasan yang bermanfaat bagi pembaca dan dapat dipahami dan diterima dengan baik bagi pembaca.

# **BAB I** **PENDAHULUAN**

## **Latar Belakang**

Pada era digital saat ini, jejaring sosial telah menjadi bagian yang penting dari kehidupan sehari hari. Melalui banyak platform, bersosialisasi kini adalah suatu kebutuhan setiap individu . Dahulu orang-orang harus bertemu secara langsung, dijaman sekarang orang dengan mudah tinggal menyalakan handphone atau pesan online dengan menggunakan gelombang internet dengan begitu sosialisasi terjalin [1].

Jejaring Sosial mewadahi media interaksi yang dibentuk dari sebuah simpul-simpul (*vertex/nodes*) dan *edge* sebagai jembatan yang menghubungkan satu atau lebih tipe relasi simpul spesifik seperti kerabat, komunitas yang disebut jejaring sosial. Dalam konteks ini, kita memahami kedekatan antar profil pengguna menjadi sangat penting untuk berbagai aplikasi, mulai dari identifikasi ikatan teman hingga analisis jejaring sosial.

Metode yang kami gunakan efektif untuk menemukan kedekatan antar profil pengguna adalah dengan menggunakan metode *matrix adjacency* dan *graph*. Graf adalah struktur yang terdiri dari simpul (*nodes/vertex*) untuk merepresentasikan pengguna dan *edge* merepresentasikan untuk hubungan antar pengguna. Dalam *matrix adjacency*, hubungan ini diwakili dalam bentuk matriks. Kami juga menggunakan metode pendekatan lainnya seperti mencari nilai *closeness* mewakili kedekatan seorang individu yang lain dan nilai *density* atau kerapatan dari suatu jejaring sosial [1]. Metode ini membantu mengidentifikasi peran penting individu dalam jejaring sosial.

Penerapan *matrix adjacency* pada jejaring sosial memungkinkan analisis yang lebih mendalam dan efisien terhadap pola interaksi dan kedekatan antar pengguna yang lebih mendalam dan efisien. *Matrix adjacency* menyediakan cara yang sistematis untuk memvisualisasikan dan mengukur hubungan antar pengguna dalam jaringan.

Pendekatan ini memungkinkankita melihat seberapa dekat dua profil pengguna dalam jaringan sosial dan dapat digunakan untuk berbagai tujuan seperti mengidentifikasi keterhubungan antar profil, komunitas berdasarkan kedekatan profil, analisis komunitas, dan manajemen jaringan.

Penelitian ini ditujukan untuk menjelaskan penerapan *matrix adjacency* dalam menentukan kedekatan profil pengguna di jejaring sosial. Kami berharap dengan memahami prinsip dasar dan penerapannya, anda akan memberikan kontribusi yang signifikan terhadap perkembangan teknologi jejaring sosial dan penerapan analisis grafik dalam berbagai bidang aplikasi praktik.

visualisasi jaringan sosial 


### Gambar 1.1 Contoh Ilustrasi jaringan sosial

### 

Penelitian ini ditujukan untuk menjelaskan penerapan *matrix adjacency* dalam menentukan kedekatan profil pengguna di jejaring sosial. Kami berharap dengan memahami prinsip dasar dan penerapannya, anda akan memberikan kontribusi yang signifikan terhadap perkembangan teknologi jejaring sosial dan penerapan analisis grafik dalam berbagai bidang aplikasi praktik.

## **Tujuan dan Kegunaan Penerapan Teori Matrix Pada Analisis Jejaring Sosial**

Analisis jejaring Sosial adalah bagian penting dari era digital yang terus berkembang pesat untuk memahami dinamika interaksi sosial dan pengaruh individu dalam suatu jaringan. Jejaring sosial mencakup banyak hal, mulai dari komunikasi antar individu hingga interaksi dalam kelompok besar, serta platform online seperti sosial media dan jaringan profesional.

Tujuan dari laporan ini adalah untuk mengeksplorasi penerapan teori *matrix* dalam analisis jejaring sosial untuk membuat graf berarah yang dapat menunjukkan hubungan yang ada antara elemen dalam jaringan. Grafis ini membantu dalam menganalisis struktur kepengurusan suatu organisasi berdasarkan garis komando. Peneliti dapat menghitung berbagai fitur dan properti graf yang digunakan untuk menganalisis jejaring sosial yang terbentuk dengan menggunakan matriks pengukuran AJS.

Penerapan teori matriks dalam analisis jejaring sosial memiliki berbagai kegunaan praktis. Memanfaatkan teori matriks dalam analisis jejaring sosial memungkinkan untuk mengevaluasi seberapa efektif komunikasi yang dibangun dalam organisasi. Peneliti dapat menggunakan program dengan bahasa Java untuk menghitung kedekatan individu dan kerapatan jejaring sosial untuk mengidentifikasi komponen jejaring sosial media yang penting, seperti broker informasi yang berperan strategis dalam penyebaran informasi dan aliran pengetahuan. Analisis jejaring sosial juga memungkinkan untuk mengevaluasi kinerja dan efektivitas organisasi berdasarkan struktur jejaring yang terbentuk dan membantu dalam menentukan kegunaan jaringan untuk setiap anggota organisasi.

# **BAB II** **KERANGKA TEORI**

## **Graf**

Graf merupakan cabang matematika yang merepresentasikan objek diskrit dan hubungan antara objek-objek sederhananya graf adalah sekumpulan titik yang dihubungkan oleh garis. Objek yang disebut dengan simpul (*nodes/vertex*) dan garis yang menghubungkan simpul (*nodes*/*vertex*) dan penghubung suatu simpul disebut sisi/garis (*edge*). Dalam Jejaring sosial , nodes dapat mewakili entitas seperti individu sedangkan edge merepresentasikan sebagai garis menghubungan satu *nodes* dengan *nodes* lain atau interaksi antara *nodes*. Teori ini pertama kali muncul di abad ke-18 ketika seorang matematikawan Swiss, Leonhard Euler, memecahkan “Masalah Tujuh Jembatan Königsberg.” [4].

**2.2 Jenis - Jenis Graf** [2]

Graf dikelompokkan menjadi dua jenis:

* Graf sederhana (*simple graf*)

Graf yang tidak memiliki sisi rangkap atau tidak memiliki sisi ganda(duplikat).

* Graf tak-sederhana (*unsimple-graph*)

Graf yang memiliki sisi ganda atau sisi yang dihubungkan dua nodes yang sama.

### 

### 

### Gambar 2.1 Contoh Ilustrasi *simple graph* dan *unsimple graph*.

Berdasarkan orientasi arah Graf terbagi menjadi dua:

* Graf berarah (*directed graph atau digraph*)

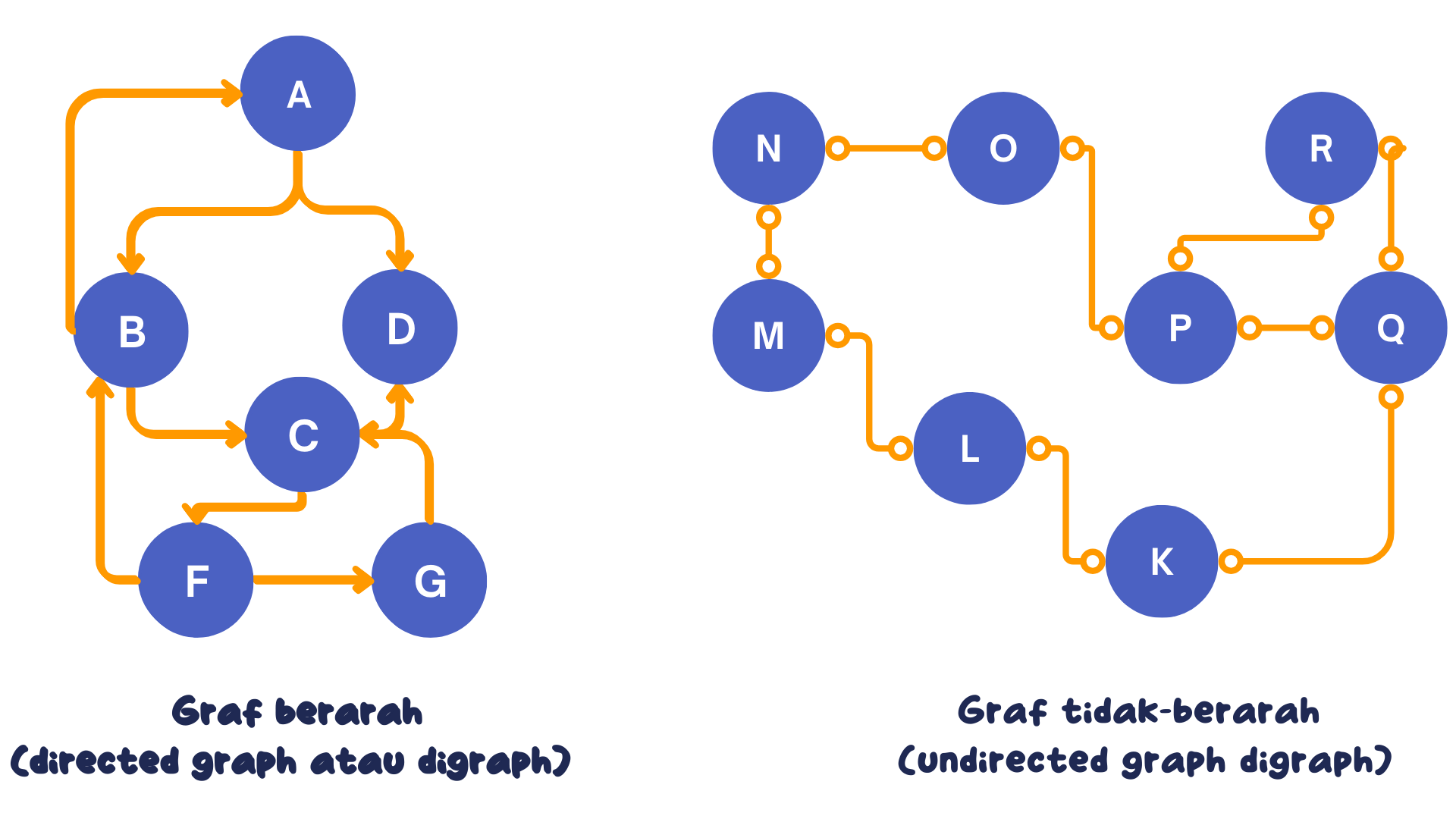
Graf disetiap sisinya berisi arah disebut graf berarah.

Contoh graf pada gambar 2.2 merupakan contoh graf berarah.

* Graf tidak-berarah (*undirected graph digraph*)

Graf di setiap sisinya tidak berisi arah disebut graf tidak-berarah.

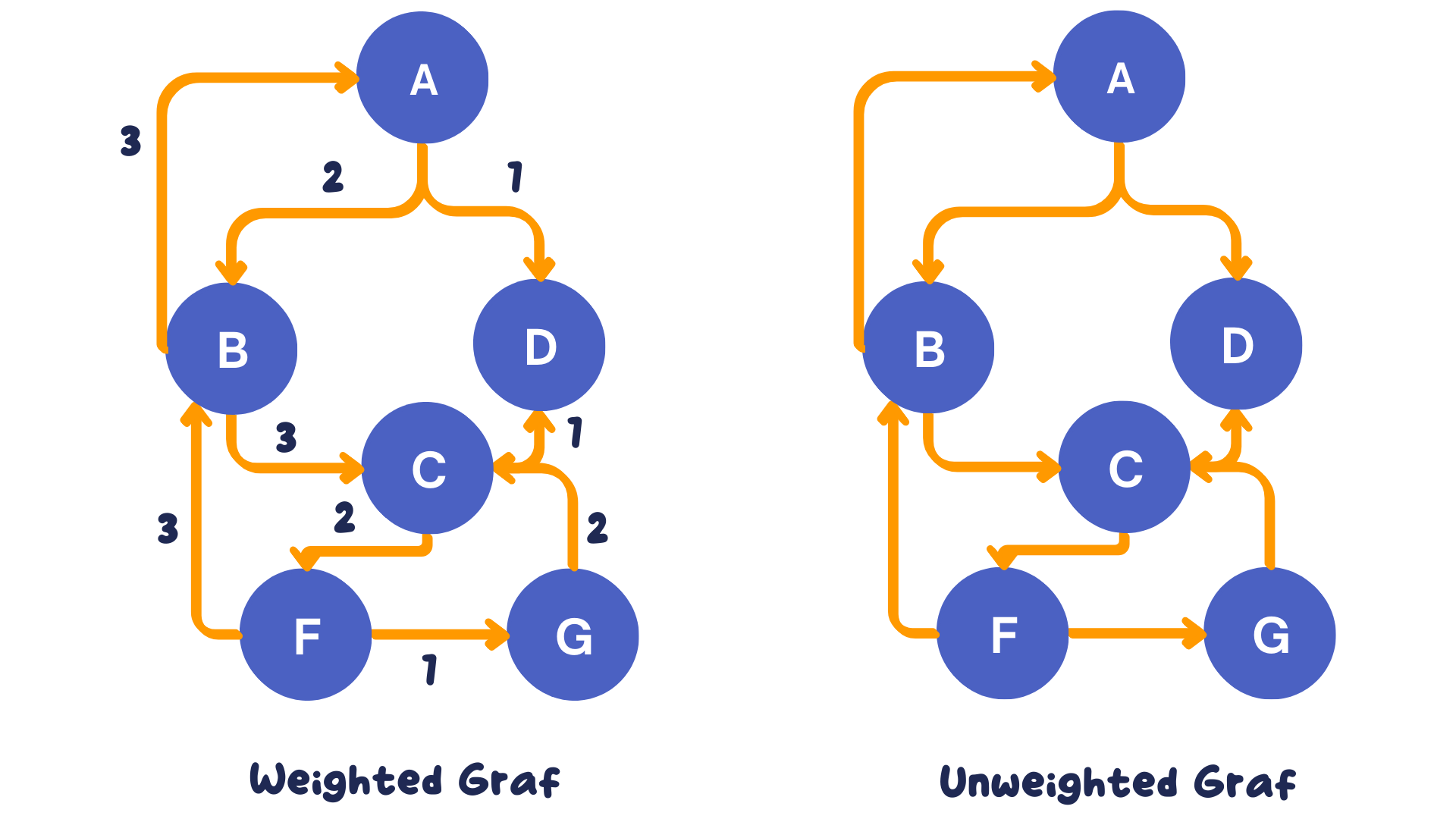
Contoh digraf pada gambar 2.2 merupakan contoh digraf tak berarah .



### Gambar 2.2 Contoh Ilustrasi graf dan digraf.

Berdasarkan nilai atau bobot dalam graf terbagi menjadi dua:

* bagian yaitu graf setiap sisi memiliki bobot menunjukkan kapasitas nodes disebut (*Weighted Graph*) sedangkan graf yang tidak memiliki bobot pada nodes tetapi hanya memiliki edge disebut (*Unweighted Graph*)



### Gambar 2.3 Contoh Ilustrasi Weighted graf dan Unweighted graf.

Representasi Graf terbagi menjadi dua matriks [1]:

* (*Adjacency Matrix*) berisi matriks dua dimensi dimana baris dan kolom mewakili nodes atau disebut vertex dan elemen simpul berisi nilai keterhubungan nodes dengan nodes lain. Contohnya jika ada hubungan nodes i dan nodes j, maka elemen matriks A[i],[j] bernilai 1, sebaliknya jika tidak ada hubungan maka bernilai 0. Adjacency Matrix pada Gambar 3.2.2 Contoh Ilustrasi graf dan digraf. merupakan contoh graf berarah.
* (*Adjacency List*) berisi setiap *nodes* memiliki daftar hubungan antara nodes lain.

# **BAB III SocioMatrix**

## **Algoritma**

Dalam era internet saat ini, jejaring sosial telah menjadi bagian penting dari kehidupan sehari-hari, baik untuk tujuan profesional maupun pribadi. Analisis kedekatan profil menjadi penting untuk berbagai aplikasi seperti pemasaran, keamanan *cyber* dan rekomendasi teman karena pentingnya hubungan antar pengguna di jejaring sosial. Akibatnya, kami memfokuskan diri pada penggunaan *Social Network Analysis* yang berfokus pada penggunaan *SocioMatrix* sebagai alat analisis untuk mengukur kedekatan antara profil pengguna dalam laporan ini.

Untuk mempermudah pemahaman, proses yang dijelaskan dalam algoritma penerapan *SocioMatrix* adalah sebagai berikut:

1. Pengumpulan Data: Pertama, data jejaring sosial dikumpulkan, yang dapat mencakup informasi tentang hubungan dan teman pengguna. Contoh data ini termasuk pengikut, daftar teman, atau interaksi antar pengguna.
2. Pembentukan *Adjacency Matrix*: Kami membentuk *Adjacency Matrix* setelah data dikumpulkan. Hubungan antara dua pengguna ditunjukkan pada setiap elemen matriks bujur sangkar dalam representasi graf ini. Elemen matriks di posisi tersebut diberi nilai 1 jika ada hubungan, seperti pertemanan atau pengikut, tetapi jika tidak, diberi nilai 0.
3. Analisis Kedekatan: Kami menganalisis kedekatan antar profil pengguna dengan *Adjacency Matrix.* Salah satu cara untuk melakukannya adalah menghitung jarak yang ada antara dua pengguna dalam graf, atau jumlah hubungan minimal yang harus dilakukan untuk mencapai satu pengguna dari pengguna lain. Metode lain adalah menganalisis komponen terhubung untuk menemukan cluster atau kelompok pengguna yang saling terhubung.
4. Analisis Kerapatan: Dengan penggunaan Adjacency Matrix, kerapatan dari suatu hubungan sosial dapat dihitung. Dimana perhitungannya dilaksanakan dengan membagi jumlah hubungan yang ada dengan jumlah hubungan yang mungkin (berpeluang) terjadi dalam keseluruhan hubungan sosial.
5. Implementasi Algoritma: Langkah terakhir adalah menerapkan algoritma dalam kode program. Kami melakukan ini dengan menggunakan bahasa pemrograman tertentu seperti Java, untuk menerapkan matriks kedekatan dan melakukan analisis kedekatan berdasarkan data yang telah dikumpulkan.

## **Contoh Perhitungan**

### Gambar 3.1 Graf Hubungan Sosial yang dipakai dalam contoh perhitungan

* + 1. **Perhitungan Closeness Centrality**

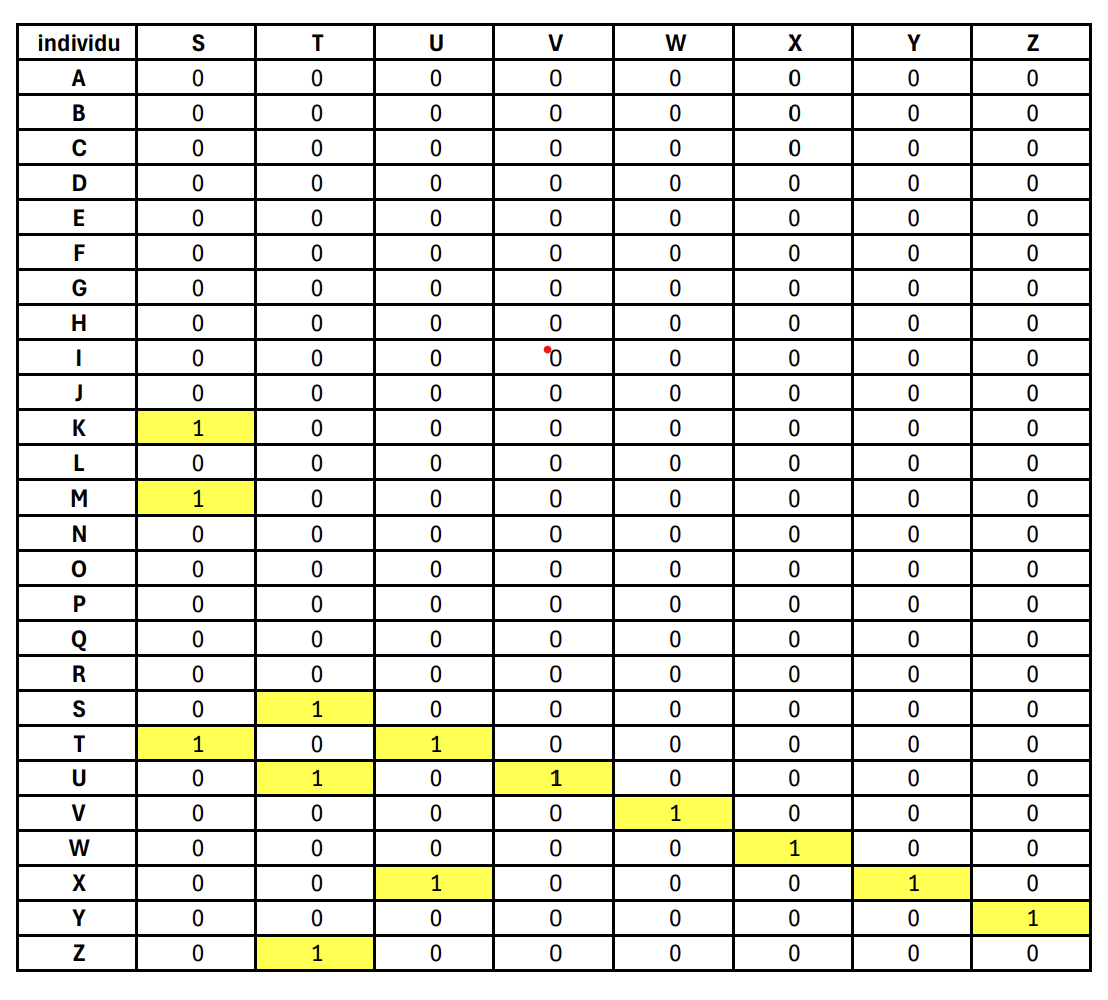
Closeness Centrality adalah *inverse* dari keseluruhan jumlah jarak satu individu pada individu lainnya. Karena terdapat kemungkinan bahwa walau suatu individu tidak secara langsung terhubung ke individu tertentu, namun bisa terhubung melalui individu lain. Melalui closeness centrality, terlihat bagaimana masing masing individu dekat dengan semua *vertex* lain, dan seberapa mudah suatu informasi sampai kepada individu tersebut, berkenaan dengan *geodesic* *distance* individu tersebut kepada semua individu lain.

Untuk memulai perhitungan, langkah pertama adalah mengubah graf yang dipakai menjadi *adjacency matrix*. Dalam contoh perhitungan ini, masing masing *edge* berbobot 1. Ketika suatu individu memiliki *outdegree* (sisi keluar vertex) kepada individu lain, maka dalam *matrix* tersebut, di baris dan kolom yang sesuai berisi nilai 1, dan ketika tidak ada outdegree yang tertuju pada individu tersebut, maka baris dan kolom tersebut berisi nilai 0.

### Tabel 3.1 Adjacency Matrix [individu](#_2w5ecyt) A sampai Z kepada [individu](#_2w5ecyt) A sampai I

### Tabel 3.2 Adjacency Matrix [individu](#_2w5ecyt) A sampai Z kepada [individu](#_2w5ecyt) J sampai R

### Tabel 3.3 *Adjacency Matrix* [individu](https://docs.google.com/document/d/1CUZDH9uU1OdrOo8j2-Z5kuZ6-V7Rc6hC/edit?pli=1#heading=h.nnk4r3y5fzqm) A sampai Z kepada [individu](https://docs.google.com/document/d/1CUZDH9uU1OdrOo8j2-Z5kuZ6-V7Rc6hC/edit?pli=1#heading=h.nnk4r3y5fzqm) S sampai Z



### 

Setelah mendapatkan *outdegree* dari masing masing [individu](#_2w5ecyt), langkah selanjutnya adalah mencari *Shortest Path* atau jarak paling dekat dari tiap [individu](#_2w5ecyt) ke semua [individu](#_2w5ecyt) yang ada dalam lingkup hubungan sosial yang terdapat pada matrix dengan menggunakan *algoritma Dijkstra*. Dan kemudian jarak dari [individu](#_2w5ecyt) tersebut kepada [individu](#_2w5ecyt) lain dimasukkan kedalam matrix kedua.

*Algoritma Dijkstra* adalah sebuah algoritma untuk pemecahan komputasi pada kasus menemukan jarak terpendek (*shortest path problem*) untuk sebuah graf berarah (*directed graph*) [3].

### Tabel 3.4 Jarak *shortest path* dari semua [individu](https://docs.google.com/document/d/1CUZDH9uU1OdrOo8j2-Z5kuZ6-V7Rc6hC/edit?pli=1#heading=h.nnk4r3y5fzqm) A sampai I

### Tabel 3.5 Jarak *shortest path* dari semua individu A sampai Z kepada individu J sampai R

### 

### 

### Tabel 3.6 Jarak *shortest path* dari semua individu A sampai Z kepada individu S sampai Z

### 

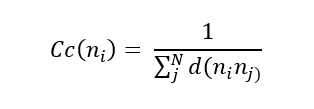
Langkah selanjutnya adalah menjumlahkan semua jarak *shortest path* yang sudah dicari sebelumnya untuk masing masing [individu](#_2w5ecyt), yang kemudian akan didapatkan:

### Tabel 3.7 Total jarak *shortest path* masing masing [individu](https://docs.google.com/document/d/1CUZDH9uU1OdrOo8j2-Z5kuZ6-V7Rc6hC/edit?pli=1#heading=h.nnk4r3y5fzqm)

### 

### 

Setelah menemukan jumlah masing masing jarak shortest path, masukkan jarak tersebut kedalam rumus Closeness Centrality, yaitu:

 (3.1)

dimana,

\

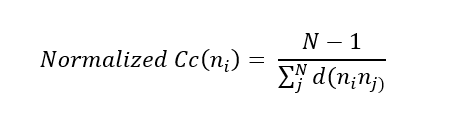
Maka kita dapatkan hasilnya sebagai berikut:

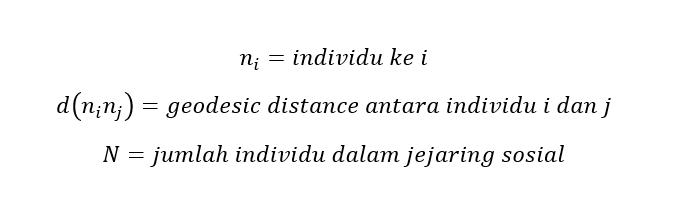
### Tabel 3.8 Perhitungan *closeness* antar [individu](#_2w5ecyt)

### 

Dengan perhitungan closeness centrality ini, dapat dilihat seberapa central dan seberapa terjangkau [individu](#_2w5ecyt) tersebut dalam jejaring sosial itu. Semakin tinggi nilai *closeness centrality* yang dimiliki, maka semakin *central* [individu](#_2w5ecyt) tersebut, semakin dekat [individu](#_2w5ecyt) tersebut kepada seluruh [individu](#_2w5ecyt) pada *matrix* [7].

Terdapat juga rumus normalized closeness dimana rumusnya adalah invers dari rata rata jumlah shortest path [individu](#_2w5ecyt) tersebut.

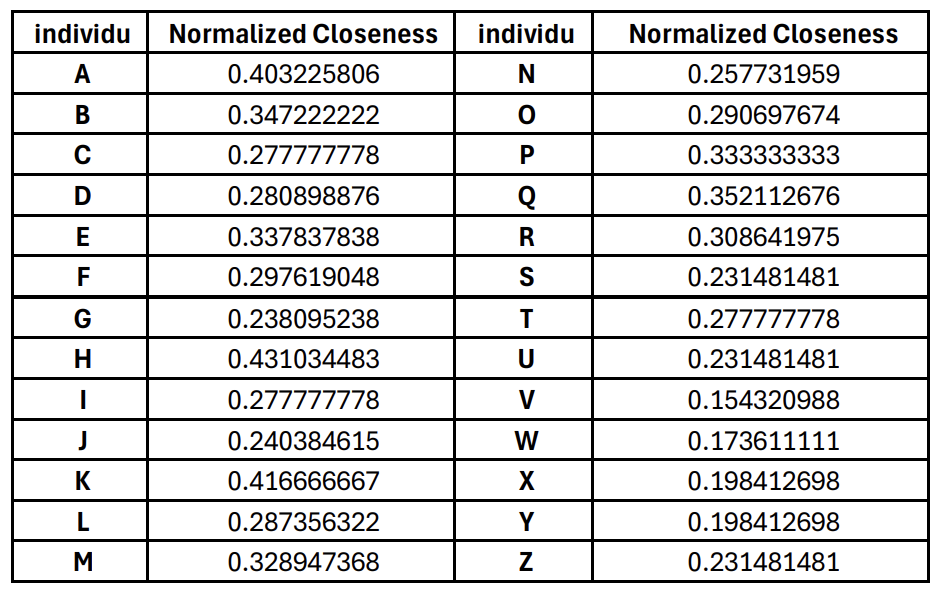
 (3.2)

dimana,

### 

Secara hasil, normalized closeness juga menganut aturan bahwa semakin tinggi value dari normalized closeness [individu](#_2w5ecyt) tersebut, semakin dekat dan terjangkau [individu](#_2w5ecyt) tersebut. Namun normalized closeness ini memperbolehkan perbandingan value normalized closeness antar jejaring sosial yang berbeda [7].

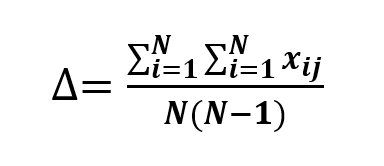
### Tabel 3.9 Perhitungan *normalized closeness* antar [individu](https://docs.google.com/document/d/1CUZDH9uU1OdrOo8j2-Z5kuZ6-V7Rc6hC/edit?pli=1#heading=h.nnk4r3y5fzqm)

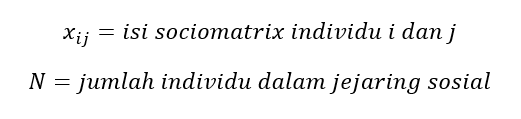


Pada matrix yang dipakai, [individu](#_2w5ecyt) K memiliki angka *centrality* yang paling tinggi, yang berarti [individu](#_2w5ecyt) tersebut adalah [individu](#_2w5ecyt) yang paling sentral dan paling dekat dengan [individu](#_2w5ecyt) lain. Individu dengan angka centrality paling tinggi adalah individu yang lebih efisien dalam penyebaran informasi dalam jejaring sosial [6].

**3.3.2. Perhitungan Density**

Selain perhitungan closeness centrality, sociomatrix dari suatu jejaring sosial dapat digunakan untuk menghitung kepadatan (*density*). Rumus dari perhitungan ini adalah pembagian jumlah hubungan yang ada dalam jejaring sosial tersebut dibagi dengan jumlah hubungan yang mungkin terjadi (peluang) [5].

(3.3)

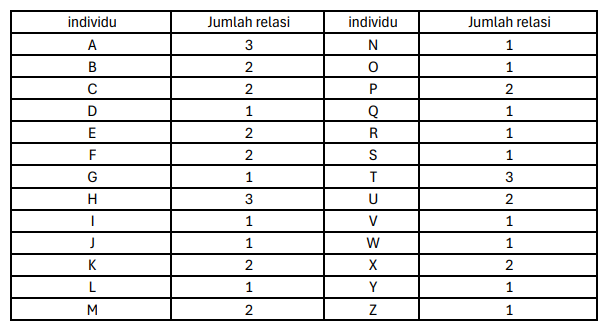
dimana,

### 

Maka semakin banyak hubungan yang ada atau semakin dekat angka hubungan dengan peluang hubungan yang terjadi, semakin padat jejaring sosial tersebut, dan juga sebaliknya [8]. Sebuah graf dinyatakan sangat padat jika memiliki nilai kepadatannya berupa 1.

Dengan menggunakan adjacency list yang telah ditampilkan pada tabel 3.2.2 sampai 3.2.4, jumlah semua hubungan yang ada dapat dihitung dengan penjumlahan kolom yang berisi nilai 1.

### Tabel 3.10 Jumlah relasi masing masing individu



Jumlah hubungan dalam contoh jejaring sosial berupa 41, dan jejaring sosial tersebut memiliki jumlah vertex 26. Maka setelah dikalkulasikan ke dalam rumus density, maka hasil density dari jejaring sosial yang kita pakai sebesar 0.063076923, yang berarti kepadatannya cukup kecil, mengingat graf sangat padat berada di nilai 1 dan tidak padat berada di nilai 0.

## **Implementasi Adjacency Matrix**

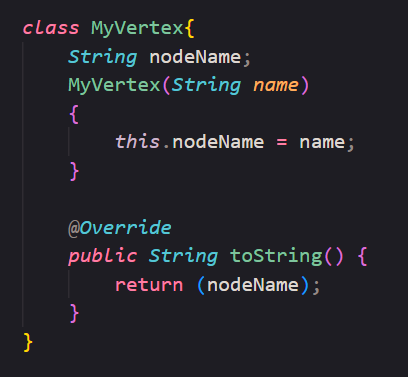
Pada penerapan perhitungan closeness centrality, normalized closeness centrality, dan density, bahasa yang digunakan adalah Java. *Library* yang digunakan adalah *library* Java Util untuk penggunaan *Map, Hash Map, Priority Queue, Arraylist, dan List.*

Dalam pembuatan dan pengujian program ini, perangkat komputer yang digunakan adalah Laptop Lenovo dengan spesifikasi perangkat dan Windows sebagai berikut:

* Processor 11th Gen Intel(R) Core(TM) i3-1115G4 @ 3.00GHz 3.00 GHz
* Installed RAM 8.00 GB (7.79 GB usable)
* System type 64-bit operating system, x64-based processor
* Edition Windows 11 Home Single Language
* Version 23H2
* Installed on ‎4/‎7/‎2023
* OS build 22631.3737
* Experience Windows Feature Experience Pack 1000.22700.1009.0

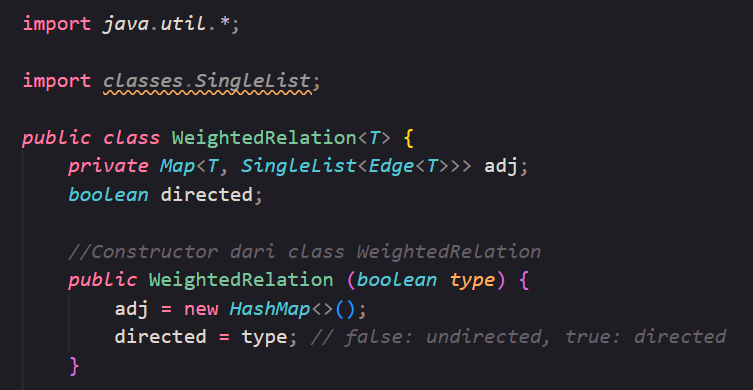
Pengujian program ini dilaksanakan dengan 10 individu yang saling berhubungan, yaitu A sampai J dari contoh jejaring sosial yang dipakai dalam contoh perhitungan. Terdapat 5 *class* yang digunakan dalam pembuatan program ini; *class Edge* yang menyimpan hubungan/*edge* antar individu dalam jejaring sosial, *class Node* yang membantu dalam iterasi tetangga/individu yang terhubung langsung dari suatu individu, *class SingleList* yang menyimpan semua hubungan yang ada dalam jejaring sosial, *class MyVertex* yang menyimpan individu dalam struktur jejaring sosial, dan *class WeightedRelation* yang merupakan struktur dan fungsi fungsi dari suatu jejaring sosial sebagai *class* yang difokuskan utama. Kode *class Edge, Node,* dan *SingleList* terlampir pada halaman lampiran.

*Class MyVertex* memiliki 1 variabel yaitu *String* nodeName yang menyimpan nama dari individu dalam jejaring sosial.



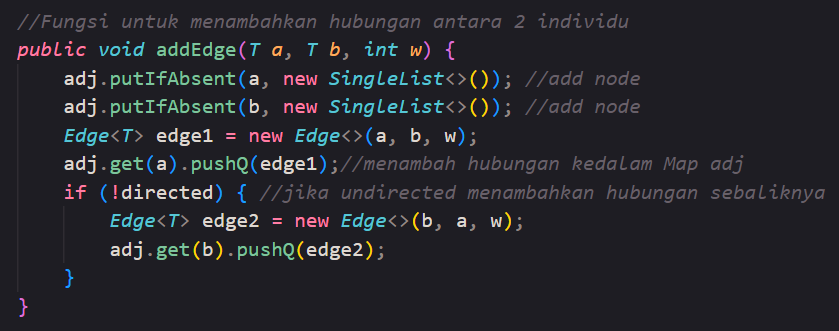
### Gambar 3.2 *Class MyVertex*

*Class WeightedRelation* memiliki 2 variabel, *Map*<*T*, *SingleList*<*Edge*<*T*>>> adj yang menyimpan individu dan hubungan yang ada dalam relasi tersebut, *boolean* directed menyatakan jika jejaring sosial tersebut berupa berarah atau tidak berarah.



### Gambar 3.3 *Library* dan *constructor class WeightedRelation*

Selanjutnya, terdapat fungsi untuk menambah hubungan antara 2 individu, yaitu addEdge. Ketika jejaring sosial berupa berarah, maka hubungan hanya akan ditambahkan sebagai hubungan antara individu a ke individu b, namun ketika jejaring sosial berupa tidak berarah, maka hubungan akan ditambahkan sebagai hubungan antara individu a ke individu b dan sebaliknya.



### Gambar 3.4 Fungsi addEdge

Selanjutnya terdapat fungsi untuk perhitungan *closeness centrality* dan *normalized closeness centrality,* yaitu fungsi *ClosenessCentrality.* Sesuai dengan gambar 1 yang terlampir pada halaman lampiran, terdapat 5 variabel yang diinisialisasi pada awal fungsi, yaitu *Map*<*T*, *Integer*> res yang menyimpan jarak shortest path terpendek sepanjang iterasi perhitungan, kemudian terdapat *PriorityQueue*<*Map*.*Entry*<*T*, *Integer*>> pq

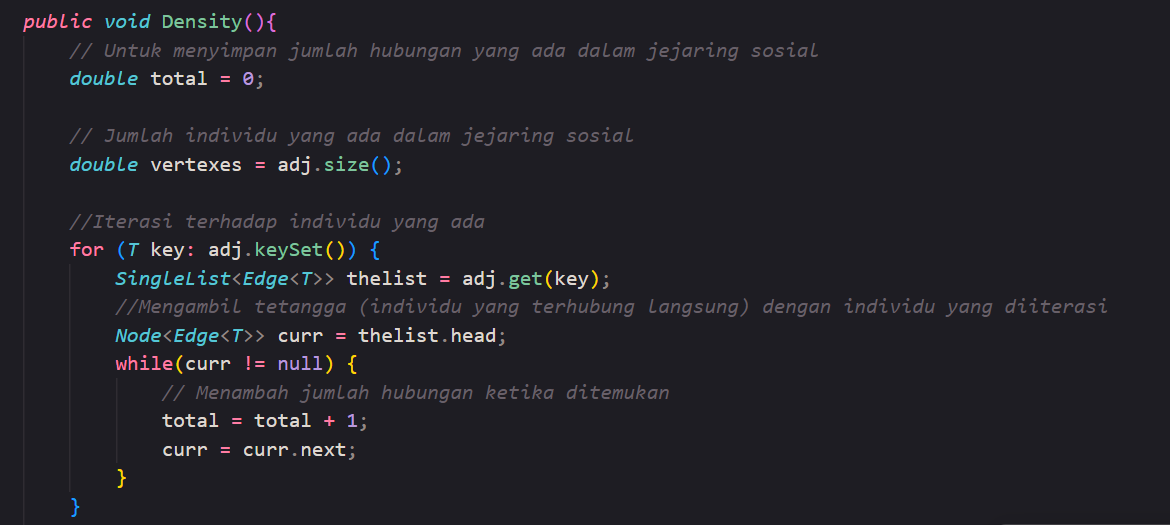
yang menyimpan priority queue sepanjang iterasi algoritma dijkstra. Selanjutnya terdapat *Map*<*T*,*T*> prev yang menyimpan path dari shortest path individu ke semua individu lain, lalu terdapat *double* vertexes yang menyimpan jumlah individu dalam jejaring sosial dengan mengambil size Map adj struktur jejaring sosial. Terakhir, terdapat *ArrayList*<*T*> path yang menyimpan path dari shortest path dalam perhitungan jumlah geodesic distance suatu individu.

Perhitungan dimulai dengan iterasi algoritma Dijkstra. Pertama, individu *start* dipilih dari individu yang ada dalam Map adj, kemudian jarak shortest path ke masing masing individu dibuat menjadi Integer.MAX\_VALUE untuk mendapatkan angka tertinggi yang memungkinkan, lalu memasukkan individu *start* ke dalam priority queue. Lalu iterasi sampai *priority queue* habis dimulai, dimana individu teratas dalam *priority queue* di *poll* lalu tetangga dari individu tersebut diambil dan dihitung kembali. Jika ternyata jarak dari individu *start* ke individu tetangga tersebut lebih pendek menggunakan path individu yang baru dikeluarkan dari priority queue, maka jarak dalam res diganti dengan jarak yang lebih pendek tersebut dan *path-*nya di masukkan kedalam prev. Proses tersebut diulang sampai semua individu telah masuk dan dikeluarkan dari priority queue. Proses algoritma ini dilampirkan pada gambar 2 pada halaman lampiran.

Setelah shortest path dari individu *start* sampai semua individu telah ditemukan, iterasi kedua dimulai untuk mendapatkan angka jumlah total geodesic distance dari individu *start.* Sesuai dengan gambar 3 pada halaman lampiran, iterasi dilaksanakan kecuali ketika individunya sama dengan individu *start.* Kemudian, dari individu yang teriterasi, mulai dari individu tersebut, *path* dalam prev akan ditelusuri sampai berakhir pada individu *start.* Selagi iterasi tersebut dilaksanakan, setiap sampai ke individu baru yang bukan individu *start*, maka total akan ditambah 1. Iterasi tersebut akan dilaksanakan sampai semua individu kecuali individu *start* sudah dikunjungi.

Setelah total dari geodesic distance individu *start* didapatkan, maka variabel yang dibutuhkan dimasukkan kepada rumus (3.1) untuk closeness centrality, dan rumus (3.2) untuk normalized closeness centrality. Setelah dihitung, maka hasilnya ditampilkan. Program tersebut dibuat sesuai dengan gambar 4 pada halaman lampiran.

Selain perhitungan closeness centrality, terdapat function Density untuk menghitung kerapatan dari jaringan sosial yang ada. Fungsi ini mengiterasi terhadap semua individu yang ada dan menghitung jumlah hubungan yang terhubung pada individu tersebut. Iterasi perhitungan tersebut dilaksanakan seperti gambar 3.5 yang dilampirkan berikut.



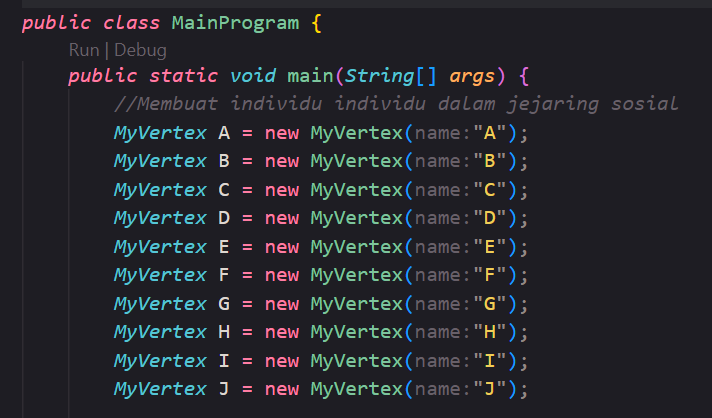
### Gambar 3.5 Iterasi perhitungan hubungan dalam fungsi Density

Setelah jumlah hubungan dalam jejaring sosial didapatkan, dan juga jumlah individu dalam jejaring sosial didapatkan melalui fungsi size pada adj, kedua variabel tersebut dimasukkan dan dihitung menggunakan rumus 3.3. Dan hasilnya kemudian ditampilkan.

### 

### Gambar 3.6 Perhitungan density menggunakan program dan menampilkan hasilnya

Maka *class* dan fungsi tersebut dapat digunakan. Pertama, setiap individu di inisialisasi menggunakan *class MyVertex.*



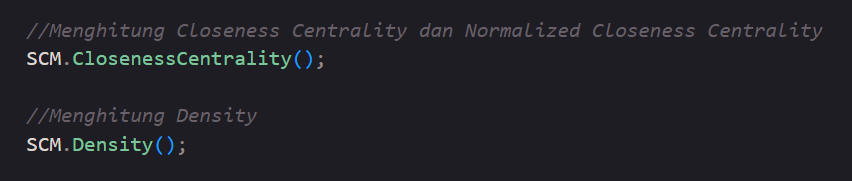
### Gambar 3.7 Inisialisasi individu dalam jejaring sosial

Setelah itu, struktur jejaring sosial diinisialisasi menggunakan *class WeightedRelation*, lalu masukkan masing masing hubungan antar individu menggunakan fungsi *addEdge*.

### 

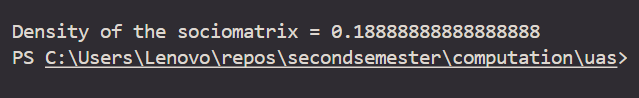
### Gambar 3.8 Inisialisasi jaringan sosial dan memasukkan hubungan antar individu

Dengan menggunakan fungsi *ClosenessCentrality* dan fungsi *Density,* hasilperhitungan closeness centrality, normalized closeness centrality, dan density ditampilkan.



### Gambar 3.9 Pemanggilan fungsi *ClosenessCentrality* dan *Density*

Hasil dari closeness centrality dan normalized closeness centrality dari semua individu ditampilkan secara bersamaan, maka output dari fungsi tersebut akan dilampirkan pada gambar 5 sampai 9 pada halaman lampiran. Hasil dari fungsi perhitungan density pada jejaring sosial yang ada akan menampilkan *output* seperti berikut.



### Gambar 3.10 *Output* dari fungsi Density terhadap jejaring sosial

# **BAB IV PENUTUP**

Pada Bab I hingga Bab III membahas tentang representasi metode matematis graf dalam konteks jejaring sosial. Graf digunakan untuk merepresentasikan objek dan hubungan objek. Dalam jejaring sosial, graf terdiri dari simpul yang mewakili entitas seperti individu dan *edge*, untuk merepresentasikan interaksi antara profil. Penerapan *Adjacency Matrix* dari Jejaring Sosial untuk Penentuan Kedekatan Profil User merupakan upaya kami melakukan pencarian tentang pentingnya jejaring sosial dalam kehidupan sehari-hari di era digital, sehingga interaksi antar individu atau komunitas dipermudah melalui platform-platform online. Kami menganalisis kedekatan profil pengguna dalam jejaring sosial dengan metode pendekatan *adjacency matrix dan graph* untuk melihat hubungan antar profil dengan menggunakan metode pendekatan lainnya seperti mencari nilai (*closeness*) mewakili kedekatan seorang individu yang lain dan nilai *density* atau kerapatan dari suatu jejaring sosial [1]. Metode ini membantu mengidentifikasi peran penting individu dalam jejaring sosial.

Tujuan dari laporan ini adalah untuk menjelaskan implementasi dan penerapan *adjacency matrix* dalam analisis jejaring sosial untuk menentukan kedekatan profil serta memahami konsep dasar graf dalam konteks jejaring sosial serta mengenal metode representasinya menggunakan *adjacency matrix*.

# **DAFTAR PUSTAKA**

[1]<https://ejournal.undip.ac.id/index.php/transmisi/article/view/5043/4577>

[2]<https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Makalah-Matdis-2015/Makalah-IF2120-2015-106.pdf>

[3]<https://mti.binus.ac.id/2017/11/28/algoritma-dijkstra/>

[4] <https://surabaya.telkomuniversity.ac.id/teori-graf-sejarah-manfaat-dan-aplikasinya/>

[5]<https://www.the-vital-edge.com/knowledge-base/what-is-network-density/>

[6]<https://cambridge-intelligence.com/social-network-analysis/>

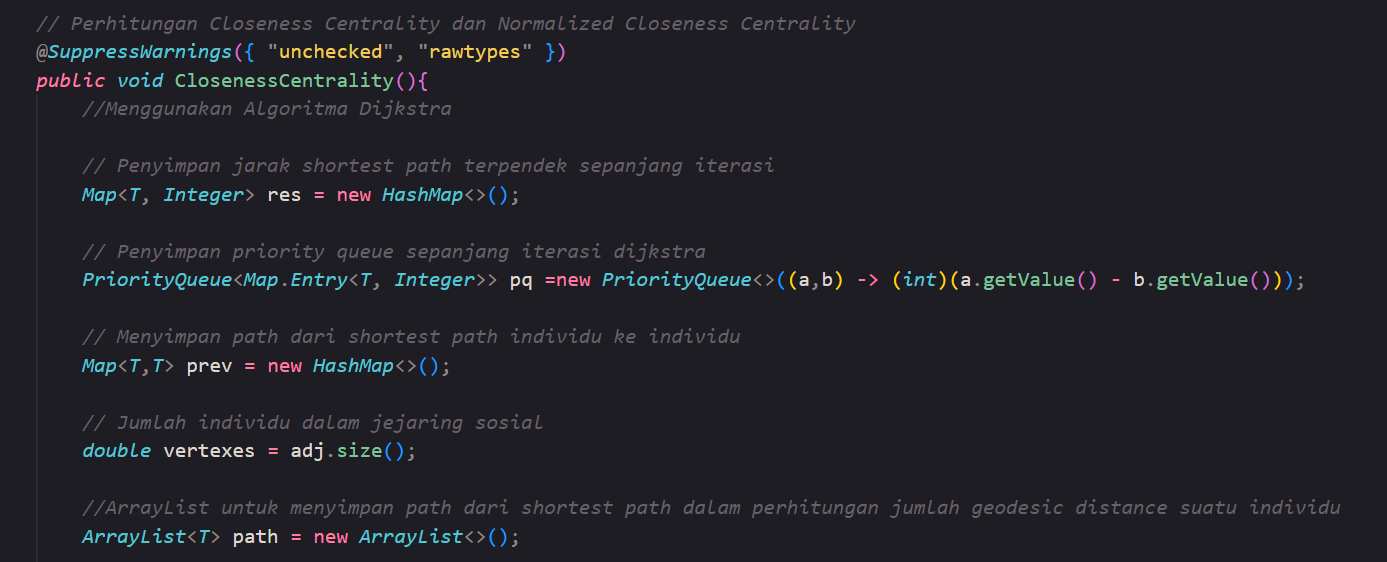
[7]<https://neo4j.com/docs/graph-data-science/current/algorithms/closeness-centrality/>

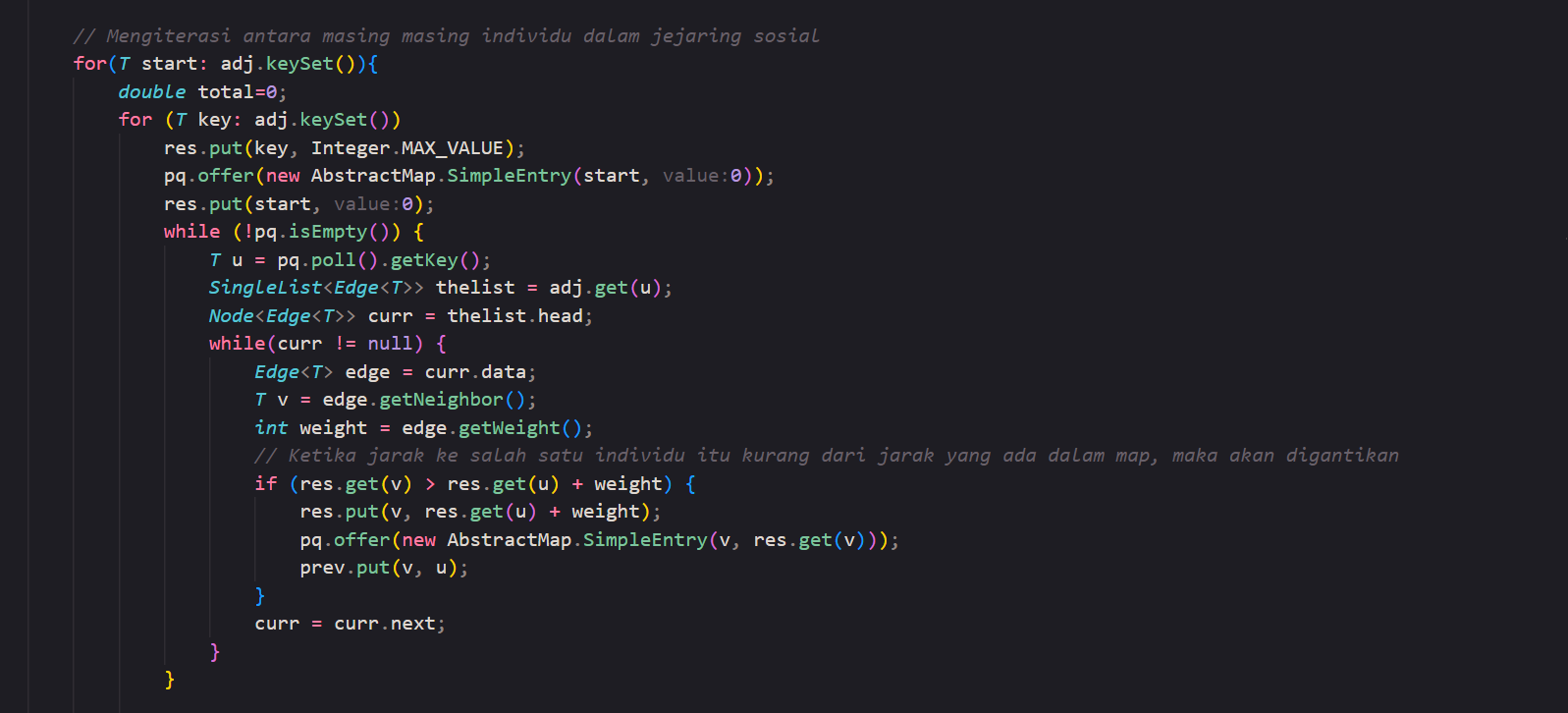
[8] <https://scholarhub.uny.ac.id/cgi/viewcontent.cgi?article=1219&context=pythagoras>

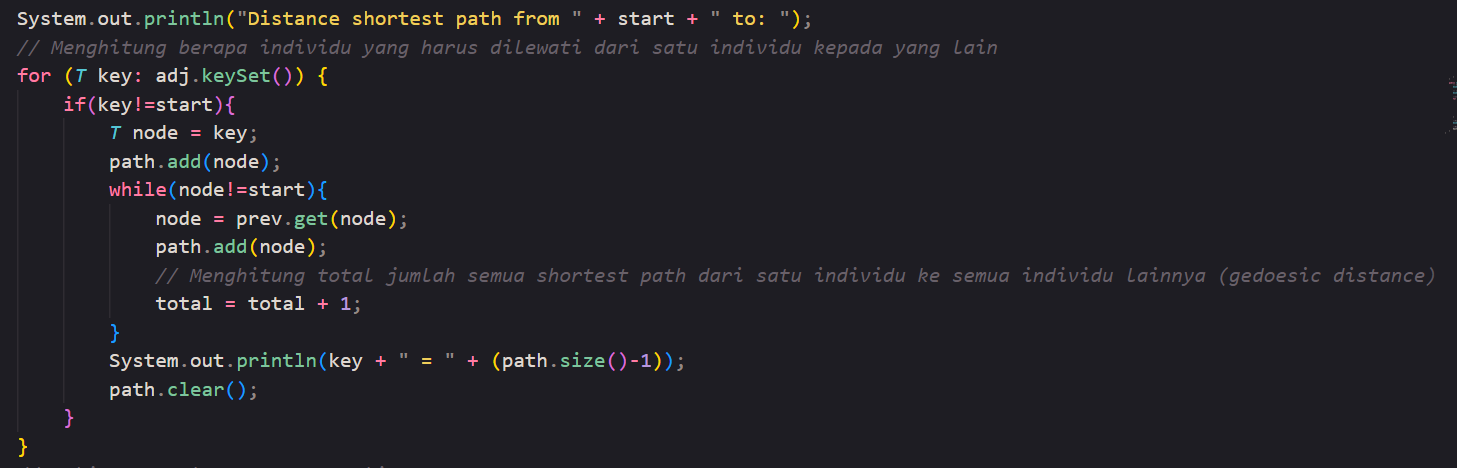
[9]<https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.centrality.closeness_centrality.html>

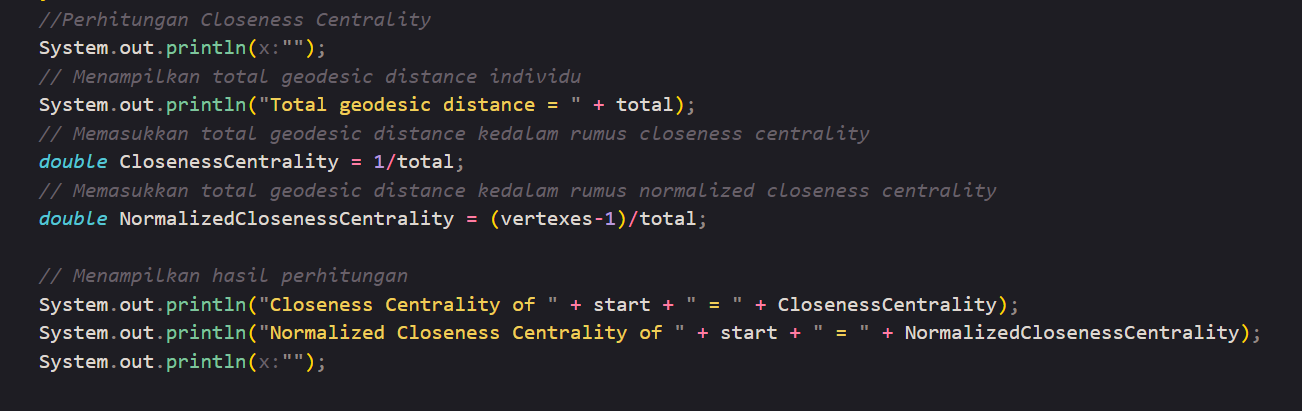
[10] <https://www.cs.cornell.edu/home/kleinber/networks-book/networks-book.pdf>

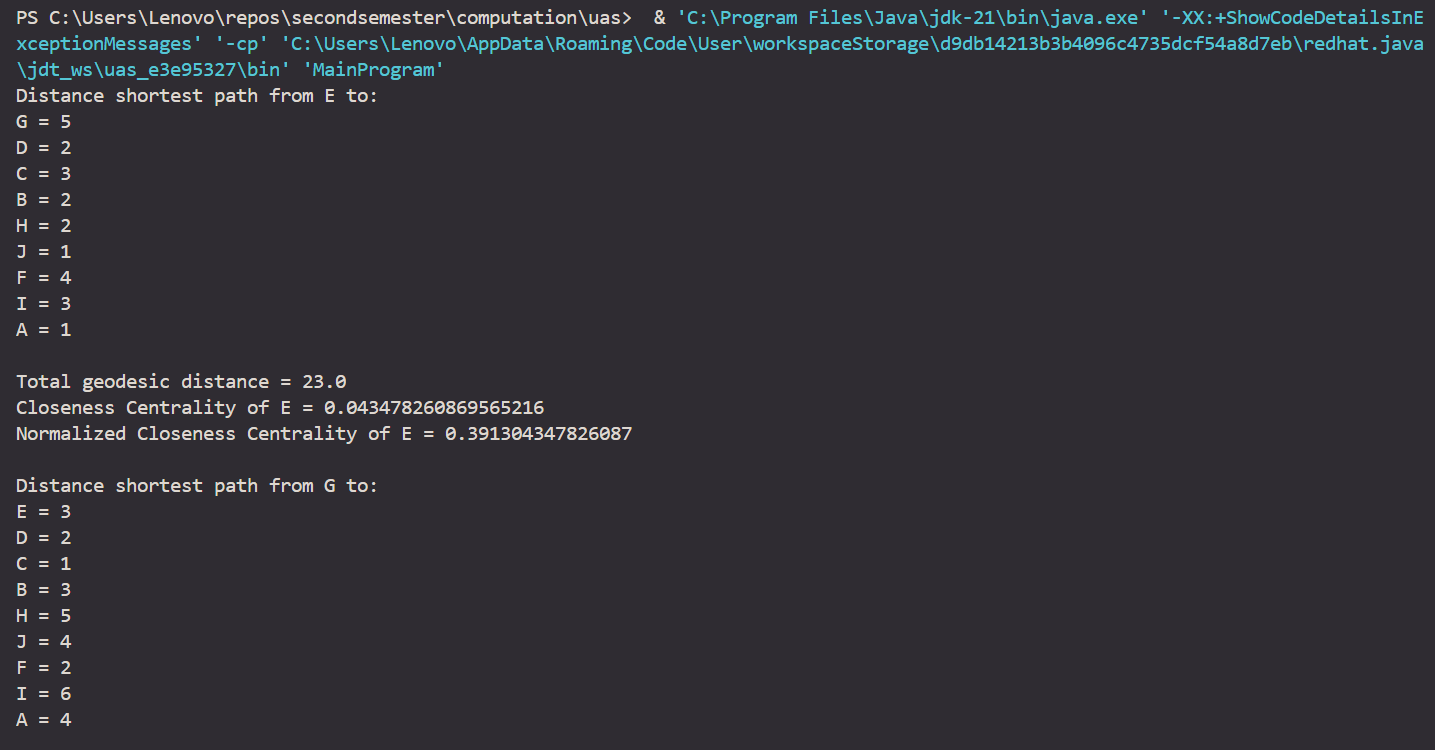
# **LAMPIRAN A LISTING PROGRAM**

Gambar 1 - Variabel yang digunakan dalam fungsi ClosenessCentrality

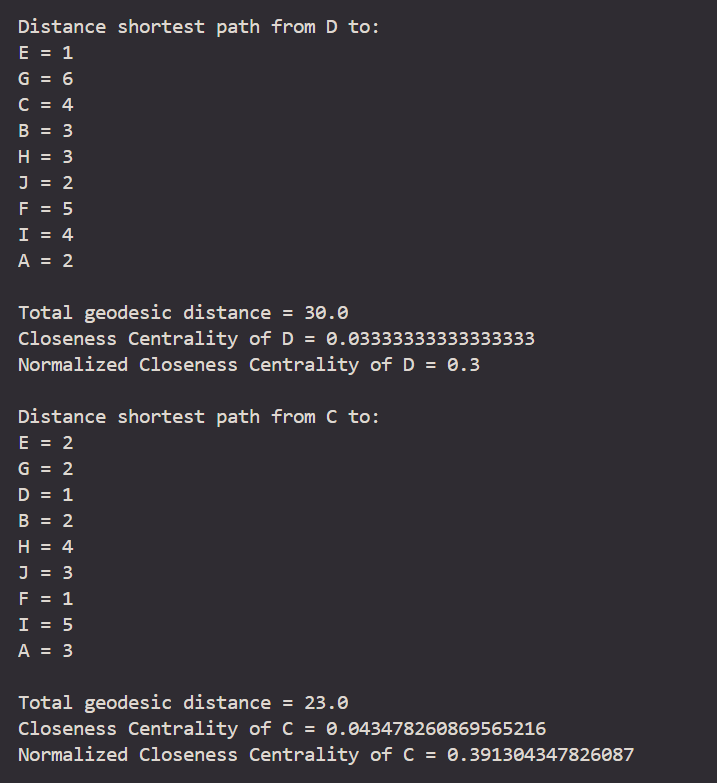
Gambar 2 - Proses iterasi dalam mencari jarak terpendek menggunakan algoritma Dijkstra menggunakan fungsi ClosenessCentrality

Gambar 3 - Perhitungan total jumlah geodesic distance dari satu individu ke salah satu individu lain dan ke semua individu

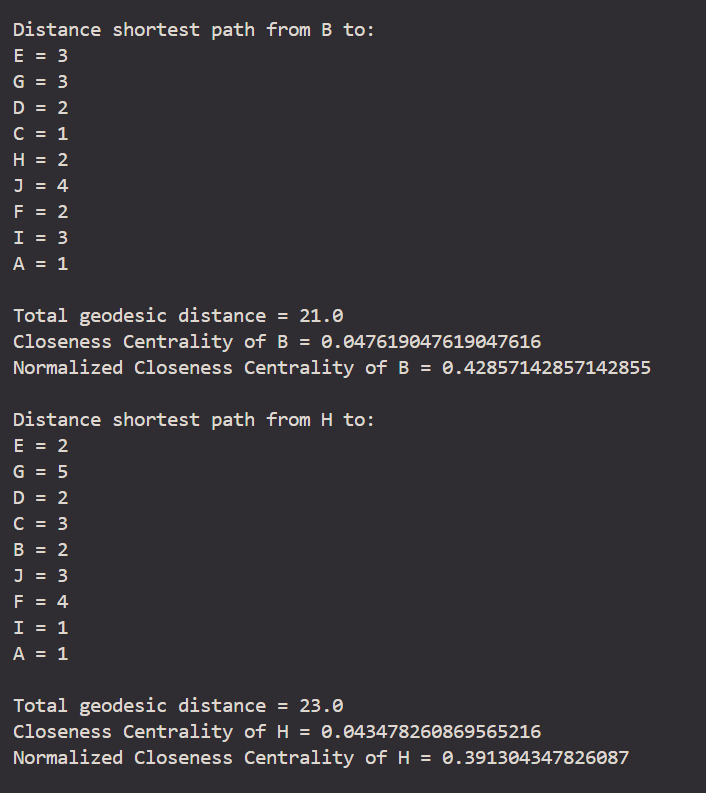
Gambar 4 - Perhitungan closeness centrality dan normalized closeness centrality sesuai rumus dan penampilan hasilnya



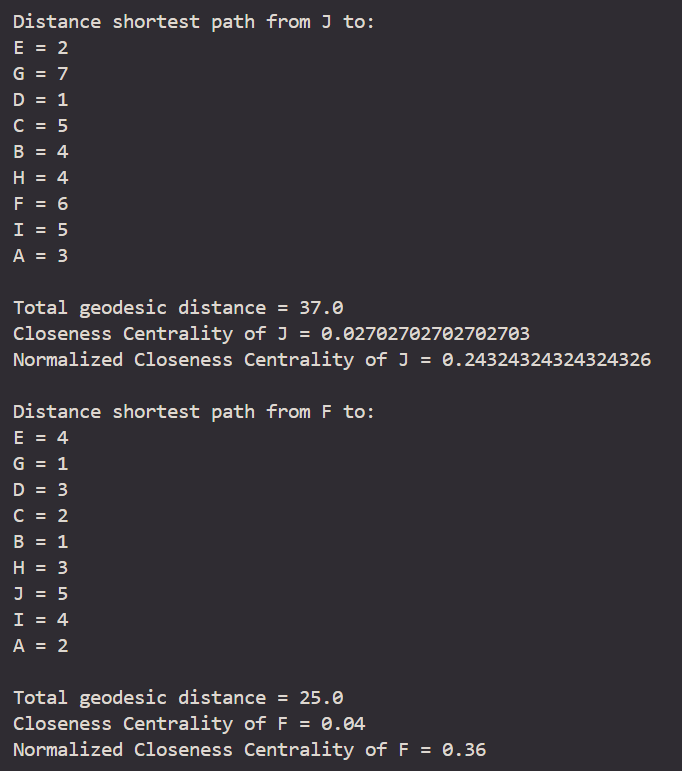
Gambar 5 - Output penggunaan fungsi ClosenessCentrality bagian E dan G



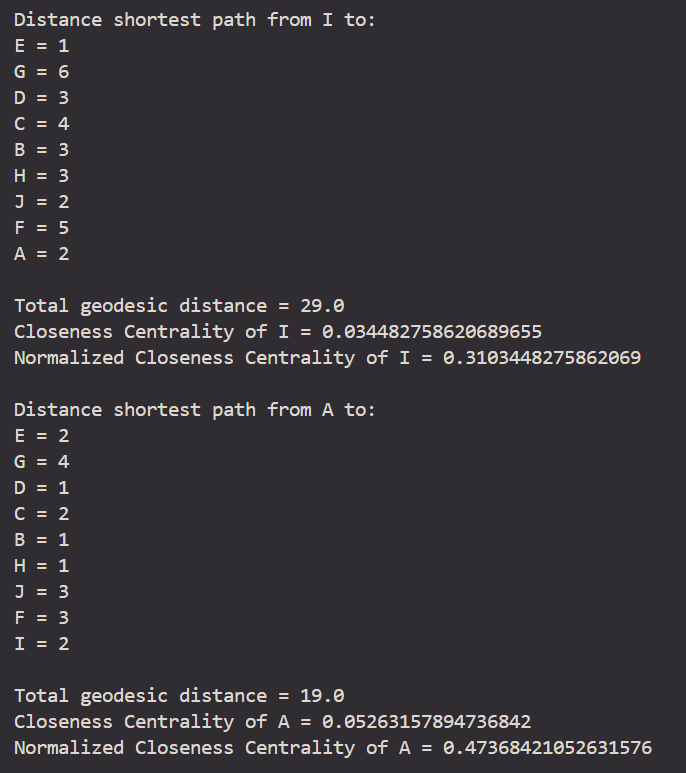
Gambar 6 - Output penggunaan fungsi ClosenessCentrality bagian D dan C



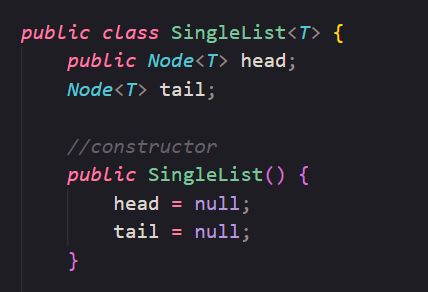
Gambar 7 - Output penggunaan fungsi ClosenessCentrality bagian B dan H



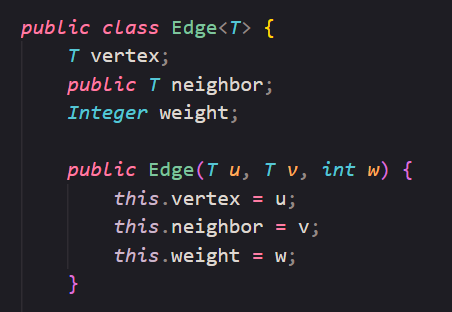
Gambar 8 - Output penggunaan fungsi ClosenessCentrality bagian J dan F



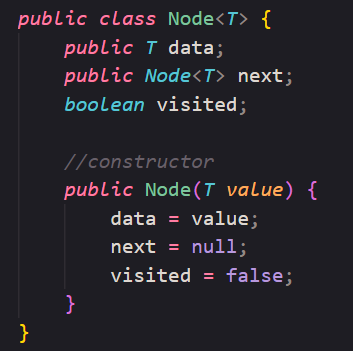
Gambar 9 - Output penggunaan fungsi ClosenessCentrality bagian I dan A



Gambar 10 - Class dan constructor SingleList



Gambar 11 - Class dan constructor Edge



Gambar 12 - Class dan constructor Node