

Aplikasi Algoritma Breadth First Search dalam Social Network Analysis dan Friend Recommendation System

Marcel Ryan Antony - 13521127

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung 40132, Indonesia

13521127@std.stei.itb.ac.id

Abstract—*Social Network Analysis* adalah sebuah analisis yang berfokus pada relasi dan interaksi antar individual, organisasi, atau entitas sosial lainnya. *Social Network Analysis* biasanya diterapkan dengan menggunakan teori graf dan statistic. Selain itu *Social Network Analysis* sendiri merupakan salah satu fondasi yang dapat digunakan untuk *Friend Recommendation System*. Dengan menggunakan algoritma *Breadth First Search*, akan diterapkan *Social Network System* dan *Friend Recommendation System*

Keywords—*Breadth First Search; Social Network Analysis; Friend Recommendation System*

I. PENDAHULUAN

Social Network Analysis merupakan salah satu hal yang dibutuhkan di zaman sekarang dimana platform-platform media sosial sudah semakin digunakan oleh orang-orang dari kalangan umur mana pun. Mengapa *Social Network Analysis* dibutuhkan? *Social Network Analysis* sendiri adalah sebuah metode untuk menganalisis hubungan dan interaksi antar individu, organisasi, atau entitas sosial lainnya pada sosial media. Hal ini tentu penting, apalagi untuk aplikasi-aplikasi sosial media seperti Facebook, Instagram, Twitter, dan lain-lain dimana aplikasi-aplikasi diatas memiliki jutaan pengguna tiap harinya sehingga dibutuhkan sebuah metode untuk memberikan info berupa relasi-relasi dari pengguna.

Beberapa manfaat dari *Social Network Analysis* adalah dapat mengetahui relasi sosial antar entitas sosial, mengidentifikasi individu-individu yang memiliki pengaruh pada media sosial, memprediksi bagaimana sebuah informasi dapat menyebar, bagaimana opini dibentuk, dan masih banyak lagi. Namun, kegunaan utama dari *Social Network Analysis* adalah memberikan *insight-insight* penting tentang struktur sosial yang ada.

Pada makalah ini, *Social Network Analysis* utamanya akan digunakan untuk menganalisis hubungan antar individu dan bagaimana keterkaitan antar individu yang ada pada sosial media. Hal ini dapat dilakukan dengan menggunakan algoritma *Breadth First Search* karena *Social Network Analysis* sendiri utamanya diterapkan dengan teori graf untuk menunjukkan struktur dari *social network*.

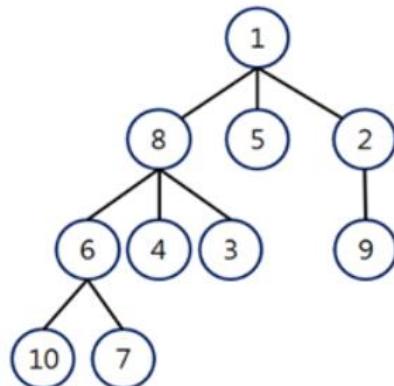
Selain itu, pada makalah ini juga akan dibahas *Friend Recommendation System* dimana sistem ini adalah sistem yang memberikan saran teman baru atau koneksi baru pada sosial media. *Friend Recommendation System* sendiri berkaitan erat dengan *Social Network Analysis* karena diperlukannya informasi tentang struktur sosial media dalam implementasinya.

Melalui mata kuliah IF2211 Strategi Algoritma, penulis merasa bahwa masalah ini dapat diselesaikan dengan algoritma *Breadth First Search*. Dengan melakukan iterasi pembacaan graf dari level ke level, informasi antar individu dengan individu dapat dengan lebih mudah diketahui. Oleh karena itu diharapkan algoritma *Breadth First Search* dapat membantu aplikasi-aplikasi sosial media untuk menganalisis suatu struktur sosial dan memberikan rekomendasi koneksi baru atau pertemuan baru

II. TEORI DASAR

A. Algoritma Breadth First Search

Breadth First Search adalah algoritma pencarian pada graf yang memulai pencarian dari simpul akar dan kemudian mengunjungi semua simpul yang berada pada tingkat kedalaman (*depth*) yang sama sebelum melanjutkan pencarian pada simpul dengan tingkat kedalaman selanjutnya.

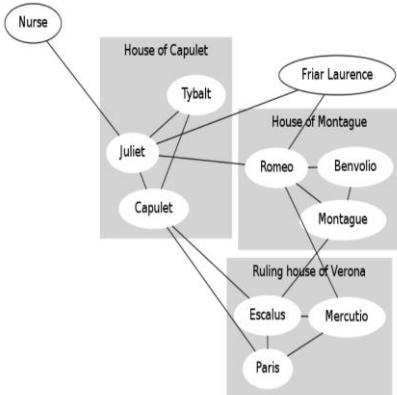


Gambar 1. Graf Contoh

Berdasarkan pada gambar graf diatas, pencarian akan dimulai dari simpul 1 dan kemudian dilanjutkan ke simpul 8, 5, 2 terlebih dahulu sebelum mengunjungi simpul anak dari simpul 8 dan 2. Implementasi algoritma BFS umumnya berkaitan dengan struktur data *queue* yang memiliki properti FIFO (*First In First Out*) untuk menentukan urutan simpul yang dikunjungi.

B. Social Network Analysis

Social Network Analysis adalah sebuah metode untuk menganalisis struktur sosial dengan tujuan mendapatkan informasi penting berupa relasi antar individu, interaksi antar individu, dan lain-lain. Dalam implementasinya, akan digunakan struktur graf dimana sebuah simpul (*node*) akan mewakilkan entitas sosial (dapat berupa individu atau organisasi) dan sisi (*edge*) akan mewakilkan hubungan antar dua simpul. Sebagai contoh dapat dilihat pada graf dibawah ini



Gambar 2. Graf Social Network

Berdasarkan gambar di atas, sebuah simpul mewakilkan satu orang dan sisi antar dua simpul berarti orang yang diwakilkan dua simpul tersebut berteman. Sebagai contoh simpul Nurse dan Juliet memiliki sisi yang menghubungkan keduanya sehingga Juliet merupakan teman dari Nurse dan sebaliknya. Pada gambar diatas, abaikan *grouping* dengan latar belakang abu-abu karena yang difokuskan disini hanyalah simpul dan sisinya saja. Dengan menggunakan *Social Network Analysis* dan membentuk graf *Friendship Network* dapat dilihat, banyak informasi yang didapatkan sebagai contoh dari graf diatas kita dapat mengetahui Benvolio merupakan teman dari Romeo sehingga Benvolio temen level kedua dari Juliet.

Mengapa informasi diatas penting? Hal ini penting untuk *Friend Recommendation System* yang akan diterapkan nantinya. Dengan mengetahui garis besar hubungan antar individu, maka akan dengan mudah didapatkan juga rekomendasi terbaik untuk teman baru pada suatu individu.

Social Network Analysis yang diimplementasikan pada makalah ini adalah menganalisis pada hubungan antara dua individu, dan juga dapat digunakan untuk mengecek teman *n-level* yang dimiliki oleh pengguna.

C. Friend Recommendation System

Friend Recommendation System merupakan salah satu sistem penting yang dimiliki oleh aplikasi sosial media mana

pun sekarang contohnya Facebook. Sistem ini digunakan untuk merekomendasikan teman baru pada suatu pengguna yang kemungkinan memiliki relasi dengan pengguna tersebut juga pada sosial media. Hal ini dapat dilakukan dengan menggunakan *Social Network Analysis* seperti yang sudah dijelaskan sebelumnya. Pada makalah ini, hanya akan diterapkan *Friend Recommendation System* yang sederhana dimana teman yang direkomendasikan hanya teman dari teman pengguna saja. Meskipun pada dasarnya, biasa rekomendasi teman akan diambil sampai *distance* nya 4 atau 5 user, penulis tidak akan mengambil teman sampai *distance* nya 4 atau 5 user melainkan hanya 2 user saja. Hal ini dikarenakan penulis merasa apabila teman yang direkomendasikan *distance* nya lebih dari 2 user, rekomendasi teman pengguna biasanya sudah tidak relevan lagi dengan pengguna (biasanya sudah merupakan *total stranger* atau orang yang tidak dikenal sama sekali oleh pengguna). Hal ini didasarkan oleh pengalaman penulis dalam menggunakan sosial media dimana biasa rekomendasi teman yang diberikan sosial media kebanyakan sudah merupakan *total stranger* bagi pengguna.

Oleh karena itu pada makalah ini, *Friend Recommendation System* yang diimplementasikan hanya akan sebatas merekomendasikan teman dari teman pengguna. Sebagai contoh berdasarkan Gambar 2, apabila ingin mencari rekomendasi teman untuk Nurse, maka rekomendasi teman yang akan didapat hanya Tybalt, Capulet, Friar Laurence, dan Romeo.

III. IMPLEMENTASI

Implementasi dilakukan menggunakan bahasa pemrograman *Python*. Implementasi dibuat dengan membuat algoritma *Friend Recommender System* dan *Social Network Analysis* menggunakan *Breadth First Search*.

A. Implementasi Data Network

Pada program, data antar individu disimpan menggunakan database *MySQL* yang nantinya akan diakses untuk mencari hubungan antar individu. Berikut adalah individu-individu yang ada pada database dan hubungan antar individu tersebut :

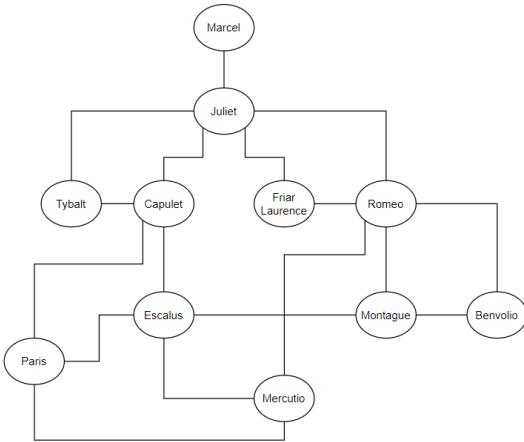
name
Benvolio
Capulet
Escalus
Friar Laurence
Juliet
Marcel
Mercutio
Montague
Paris
Romeo
Tybalt

Gambar 3. Data Individu yang digunakan

name	friend_name
Montague	Benvolio
Romeo	Benvolio
Escalus	Capulet
Juliet	Capulet
Paris	Capulet
Tybalt	Capulet
Capulet	Escalus
Mercutio	Escalus
Montague	Escalus
Paris	Escalus
Juliet	Friar Laurence
Romeo	Friar Laurence
Capulet	Juliet
Friar Laurence	Juliet
Marcel	Juliet
Romeo	Juliet
Tybalt	Juliet
Juliet	Marcel
Escalus	Mercutio
Paris	Mercutio
Romeo	Mercutio
Benvolio	Montague
Escalus	Montague
Romeo	Montague
Capulet	Paris
Escalus	Paris
Mercutio	Paris
Benvolio	Romeo
Friar Laurence	Romeo
Juliet	Romeo
Mercutio	Romeo
Montague	Romeo
Capulet	Tybalt
Juliet	Tybalt

Gambar 4. Hubungan antar individu

Agar lebih jelas, berikut adalah representasi hubungan antar individu dalam graf :



Gambar 5. Hubungan antar individu dalam graf

Simpul merepresentasikan individu dan sisi antar simpul berarti kedua individu berteman.

B. Implementasi Social Network Analysis

Berikut adalah kode program untuk *Social Network Analysis* :

```
from classtype.People import People
from classtype.Graph import Graph

def socialNetworkAnalysis(graph : Graph, srcPerson : People, destPerson : People):
    # Set adjacency matrix
    checked = [0 for _ in range(len(graph.getPersons()))]

    # Set checked true to the src person
    searchedIndex = graph.getPersonIndexByName(srcPerson.getName())
    checked[searchedIndex] = True
```

```
# Initialize depth level
depth = 1

# Initialize name list to be traversed
nameList = []

# Add srcPerson friend to the list
for friendName in srcPerson.getFriends():
    nameList.append(friendName)
    index = graph.getPersonIndexByName(friendName)
    checked[index] = True

# Set boolean condition
check = True

# Traversing through the graph to see relation between srcPerson and destPerson
while check:
    newNameList = []

    for personName in nameList:
        if (personName == destPerson.getName()):
            return depth

        person = graph.getPersonByName(personName)

        # Get person friends
        for friendName in person.getFriends():
            # Get person friend index
            friendIndex = graph.getPersonIndexByName(friendName)
            if (not checked[friendIndex]):
                checked[friendIndex] = True
                newNameList.append(friendName)

    nameList = newNameList

    depth += 1

    check = (len(nameList) != 0)

return 0

def getNLevelFriend(graph: Graph, person: People, level):
    # Set adjacency matrix
    checked = [0 for _ in range(len(graph.getPersons()))]

    # Set checked true to the src person
    searchedIndex = graph.getPersonIndexByName(person.getName())
    checked[searchedIndex] = True

    # Initialize depth level
    depth = 1

    # Initialize name list to be traversed
    nameList = []
```

```

# Add person's friend
for friendName in person.getFriends():

    nameList.append(friendName)

    index = graph.getPersonIndexByName(friendName)

    checked[index] = True

# Set boolean condition
check = True

while check:

    if (depth == level):
        return nameList

    newNameList = []

    for personName in nameList:
        person = graph.getPersonByName(personName)

        # Get person friends
        for friendName in person.getFriends():

            # Get person friend index
            friendIndex = graph.getPersonIndexByName(friendName)

            if (not checked[friendIndex]):
                checked[friendIndex] = True
                newNameList.append(friendName)

    nameList = newNameList

    depth += 1

    check = (len(nameList) != 0)

return []

```

```

1 from classtype.Graph import Graph
2 from classtype.People import People
3 from algorithm.sna import *
4 import heapq
5
6 def recommendFriends(graph: Graph, person: People):
7     return getNLevelFriend(graph, person, 2)

```

Pada kode diatas, dapat dilihat dari import terdapat kelas Graph dan kelas Person. Kedua kelas tersebut adalah kelas buatan untuk kelas Person akan mengandung attribut berupa nama dan teman dari orang tersebut sedangkan kelas Graph akan mengandung attribut berupa seluruh orang/individu yang ada pada database/*social network*. Selain itu dapat dilihat juga terdapat import dari module sna yang berisi fungsi-fungsi untuk *Social Network Analysis*.

Fungsi recommendFriends menerima 2 parameter berupa graf yang berisi struktur *social network* dan individu yang ingin diberikan rekomendasi. Implementasi ini hanya memanfaatkan fungsi dari *Social Network Analysis* yaitu *getNLevelFriend* yang digunakan untuk mengambil *n-level* teman pengguna. Pada *Friend Recommendation System* ini hanya diambil teman level kedua pengguna sebagai rekomendasi teman untuk pengguna karena alasan yang sudah dijelaskan sebelumnya.

D. Implementasi Kode Utama

Berikut adalah kode utama untuk *Social Network Analysis* dan *Friend Recommendation System*

```

import mysql.connector

from classtype.People import People
from classtype.Graph import Graph
from algorithm.recommender import recommendFriends
from algorithm.sna import socialNetworkAnalysis, getNLevelFriend

try:
    conn = mysql.connector.connect(
        host = 'localhost',
        user = 'root',
        port = 3307,
        password = 'password',
        database = 'makalah_stima'
    )

    cursor = conn.cursor()

    # Get all person in the database

    cursor.execute("SELECT * FROM Person")

    records = cursor.fetchall()

    # List all person in the database
    listOfPerson = []

```

Berdasarkan kode diatas, terdapat 2 fungsi utama yaitu *socialNetworkAnalysis* yang menerima parameter graf dan dua person. Fungsi ini digunakan untuk mengetahui apa hubungan antar dua orang, untuk lebih spesifiknya pada relasi *n-level* ke berapa kah mereka berteman. Fungsi ini menggunakan pendekatan *Breadth First Search* dimana iterasi akan dilakukan dengan mengecek individu-individu pada tingkat kedalaman 1 terlebih dahulu dan kemudian melanjutkan pencarian ke tingkat kedalaman selanjutnya. Fungsi ini akan mengembalikan tingkat kedalaman / *n-level* ketika *destPerson* ditemukan pada graf struktur *social network*

Kemudian terdapat fungsi *getNLevelFriend* yang menerima tiga parameter berupa graf, individu yang ingin dicari temannya, dan tingkat kedalamannya. Fungsi ini juga menggunakan pendekatan *Breadth First Search* dimana tiap simpul/individu akan diiterasi berdasarkan tingkat kedalaman terendah terlebih dahulu. Apabila graf telah sampai pada tingkat kedalaman yang ingin di cek, maka individu-individu akan disimpan ke dalam sebuah *list* dan dijadikan *return*

C. Implementasi Friend Recommendation System

Berikut adalah kode program untuk *Friend Recommendation System* :

```

for row in records:
    person = People(row[0])
    listOfPerson.append(person)

# Get all person friend in database
for person in listOfPerson:
    cursor.execute("SELECT friend_name FROM friends WHERE name = %s",
    (person.getName(),))

    # Fetch the data
    records = cursor.fetchall()

    # Assign the data to a list
    listOfFriends = []

    for row in records:
        listOfFriends.append(row[0])

    # Assign the list to the person
    person.setFriends(listOfFriends)

# Set the graph
graph = Graph(listOfPerson)

print("Welcome to Social Network Analysis and Friend Recommendation Program!!!")
print("Please choose what do you want to do with this program :")
print("1. Social Network Analysis Program")
print("2. Friend Recommendation Program")

choice = int(input("Input :"))

# Friend Recommendation Program
if (choice == 2):
    # Print all names to select for the recommender system
    graph.printNames()

    inputIndex = int(input("Select a person to be searched by number from the
above list to search :"))

    person = graph.getPersonByIndex(inputIndex-1)

    # Get recommended Friends
    recommendedFriends = recommendFriends(graph, person)

    # Print all the recommended friend names
    print("The recommended friend for you are :")
    for i in range(len(recommendedFriends)):
        print(f"{i+1}. {recommendedFriends[i]}")

else:

    print("Select what do you want to analysis :")
    print("1. Relation between person A or person B")
    print("2. Get all the person that is n-level friend of person A")

choice = int(input("Input:"))

# Print all names to select for the recommender system

```

```

graph.printNames()

# Social network analysis to analyze relation between 2 person
if (choice == 1):
    firstPersonIndex = int(input("Select the first person by the number from
the above list :"))

    secondPersonIndex = int(input("Select the first person by the number from
the above list :"))

    firstPerson = graph.getPersonByIndex(firstPersonIndex-1)
    secondPerson = graph.getPersonByIndex(secondPersonIndex-1)

    depth = socialNetworkAnalysis(graph, firstPerson, secondPerson)

    if depth == 1:
        print(f"{secondPerson.getName()} is the first level friend of
{firstPerson.getName()}")
    elif depth == 2:
        print(f"{secondPerson.getName()} is the second level friend of
{firstPerson.getName()}")
    elif depth == 3:
        print(f"{secondPerson.getName()} is the third level friend of
{firstPerson.getName()}")
    elif depth == 4:
        print(f"{secondPerson.getName()} is the fourth level friend of
{firstPerson.getName()}")

    # Social network analysis to get n-level friend of a person
else:
    personIndex = int(input("Select the person you want to analysis by the
number from above list :"))

    level = int(input("Select level :"))

    person = graph.getPersonByIndex(personIndex-1)

    nLevelFriend = getNLevelFriend(graph, person, level)

    print(f"All of your {level}-level friend are :")

    for i in range(len(nLevelFriend)):
        print(f"{i+1}. {nLevelFriend[i]}")

except mysql.connector.Error as e:
    print("Error reading data from MySQL table")
finally:
    if conn.is_connected():
        conn.close()
        cursor.close()

```

Kode utama diatas menggunakan *localhost* untuk menghubungkan *MySQL* dengan algoritma-algoritma yang ada.

IV. ANALISIS

A. Social Network Analysis

- Mencari Relasi antar dua individu

Mencari Relasi antara Marcel dan Benvolio

```
Welcome to Social Network Analysis and Friend Recommendation Program!!!
Please choose what do you want to do with this program :
1. Social Network Analysis Program
2. Friend Recommendation Program
Input : 1
Select what do you want to analysis :
1. Relation between person A or person B
2. Get all the person that is n-level friend of person A
Input: 1
1. Benvolio
2. Capulet
3. Escalus
4. Friar Laurence
5. Juliet
6. Marcel
7. Mercutio
8. Montague
9. Paris
10. Romeo
11. Tybalt
Select the first person by the number from the above list : 6
Select the first person by the number from the above list : 1
Benvolio is the third level friend of Marcel
```

Dapat dilihat berdasarkan *Gambar 5*, Benvolio memang merupakan teman level ketiga untuk Marcel karena Marcel berteman dengan Juliet dan Juliet berteman dengan Romeo dan Romeo berteman dengan Benvolio. User path yang dihasilkan Marcel-Juliet-Romeo-Benvolio

Mencari relasi antara Romeo dan Juliet

```
Welcome to Social Network Analysis and Friend Recommendation Program!!!
Please choose what do you want to do with this program :
1. Social Network Analysis Program
2. Friend Recommendation Program
Input : 1
Select what do you want to analysis :
1. Relation between person A or person B
2. Get all the person that is n-level friend of person A
Input: 1
List of person in the database:
1. Benvolio
2. Capulet
3. Escalus
4. Friar Laurence
5. Juliet
6. Marcel
7. Mercutio
8. Montague
9. Paris
10. Romeo
11. Tybalt
Select the first person by the number from the above list : 5
Select the first person by the number from the above list : 10
Romeo is the first level friend of Juliet
```

Dapat dilihat berdasarkan *Gambar 5* Romeo merupakan teman langsung dari Juliet sehingga Romeo adalah *first level friend* dari Juliet dan sebaliknya

- Mencari *n-level* teman dari seorang individu

Mencari teman level ketiga dari Marcel

```
Welcome to Social Network Analysis and Friend Recommendation Program!!!
Please choose what do you want to do with this program :
1. Social Network Analysis Program
2. Friend Recommendation Program
Input : 1
Select what do you want to analysis :
1. Relation between person A or person B
2. Get all the person that is n-level friend of person A
Input: 2
List of person in the database:
1. Benvolio
2. Capulet
3. Escalus
4. Friar Laurence
5. Juliet
6. Marcel
7. Mercutio
8. Montague
9. Paris
10. Romeo
11. Tybalt
Select the person you want to analysis by the number from above list : 6
Select level : 3
All of your 3-level friend are :
1. Escalus
2. Paris
3. Benvolio
4. Mercutio
5. Montague
```

Dapat dilihat dari *Gambar 5*, teman level ketiga dari Marcel sudah sesuai dengan yang dihasilkan program

Mencari teman level pertama dari Marcel

```
Welcome to Social Network Analysis and Friend Recommendation Program!!!
Please choose what do you want to do with this program :
1. Social Network Analysis Program
2. Friend Recommendation Program
Input : 1
Select what do you want to analysis :
1. Relation between person A or person B
2. Get all the person that is n-level friend of person A
Input: 2
List of person in the database:
1. Benvolio
2. Capulet
3. Escalus
4. Friar Laurence
5. Juliet
6. Marcel
7. Mercutio
8. Montague
9. Paris
10. Romeo
11. Tybalt
Select the person you want to analysis by the number from above list : 6
Select level : 1
All of your 1-level friend are :
1. Juliet
```

Karena teman Marcel hanya Juliet maka teman level pertama dari Marcel hanyalah Juliet (dapat dilihat dari *Gambar 5*)

B. Friend Recommendation System

Mencari rekomendasi teman untuk Marcel

```
Welcome to Social Network Analysis and Friend Recommendation Program!!!
Please choose what do you want to do with this program :
1. Social Network Analysis Program
2. Friend Recommendation Program
Input : 2
List of person in the database:
1. Benvolio
2. Capulet
3. Escalus
4. Friar Laurence
5. Juliet
6. Marcel
7. Mercutio
8. Montague
9. Paris
10. Romeo
11. Tybalt
Select a person to be searched by number from the above list to search : 6
The recommended friend for you are :
1. Capulet
2. Friar Laurence
3. Romeo
4. Tybalt
```

Rekomendasi teman untuk Marcel hanya merupakan teman dari Juliet karena Marcel hanya berteman dengan Juliet (dapat dilihat pada Gambar 5)

untuk hal ini. Namun, untuk *Friend Recommendation System* meskipun algoritma *Breadth First Search* juga dapat digunakan, terdapat beberapa algoritma yang lebih baik daripada *BFS* karena terdapat beberapa faktor yang harus dipertimbangkan selain *user distance* saja.

UCAPAN TERIMA KASIH

Puji syukur penulis ucapan kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya, maka makalah “Aplikasi Algoritma Breadth First Search dalam Social Network Analysis dan Friend Recommendation System” ini dapat diselesaikan dengan tepat waktu.

Penulis juga mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T. selaku dosen pengajar Mata Kuliah IF2211 Strategi Algoritma Kelas 01 Tahun Ajaran 2022/2023 atas bimbingan dan pengajaran yang telah diberikan kepada penulis selama ini.

Terakhir, penulis juga mengucapkan terima kasih kepada semua orang yang telah membaca makalah ini dan juga kepada semua orang yang telah memberi penulis dukungan selama pengerjaan makalah ini.

SAMPEL KODE

Sampel kode dapat diakses di pranala berikut :
<https://github.com/MarcelRyan/sna-and-friend-recommendation-bfs>

REFERENSI

- [1] Bharati, Ashank. (2019). *Social Networking with BFS and Neo4j*. <https://medium.com/nuclei-technology-blog/social-networking-with-bfs-and-neo4j-ea52dc4ce198>. Diakses pada 21 Mei 2023, pukul 13.34 WIB
- [2] Munir, Rinaldi. (2021). *Breadth First Search (BFS) dan Depth First Search (DFS) (Bagian 1)*. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/BFS-DFS-2021-Bag1.pdf>. Diakses pada 21 Mei 2023, pukul 14.52 WIB
- [3] Munir, Rinaldi. (2021). *Breadth First Search (BFS) dan Depth First Search (DFS) (Bagian 2)*. <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/BFS-DFS-2021-Bag1.pdf>. Diakses pada 21 Mei 2023, pukul 14.52 WIB

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 22 Mei 2023

Marcel Ryan Antony - 13521127

V. KESIMPULAN

Berdasarkan penelitian ini, dapat disimpulkan bahwa algoritma *Breadth First Search* dapat digunakan untuk *Social Network Analysis* dan *Friend Recommendation System*. Karena pada dasarnya *Social Network Analysis* direpresentasikan dengan menggunakan graf sehingga penggunaan algoritma *Breadth First Search* sangat cocok