

# Penerapan Algoritma BFS dalam Memberikan *Insight* dari Graf *Social Networking*

Florenzia Wijaya 13518020  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
florenziawijaya11@gmail.com

**Abstrak**—Beberapa tahun ini, *social networking* telah menjadi suatu hal yang umum dalam penggunaan internet. Adanya *social networking* saat ini sangat membantu orang-orang dalam berkenalan, berkomunikasi, dan memperluas relasi pertemanan maupun rekan bisnis. Algoritma *Breadth First Search (BFS)* dapat diaplikasikan pada *social networking* untuk memberikan rekomendasi teman berdasarkan level pertemanan tertentu. Rekomendasi ini dapat diperoleh dari jaringan pertemanan pada suatu *platform*. Jaringan pertemanan ini dapat dipetakan ke dalam bentuk graf.

**Kata Kunci**—Graf, *Breadth First Search*, *Social Networking*, Strategi Algoritma

## I. PENDAHULUAN

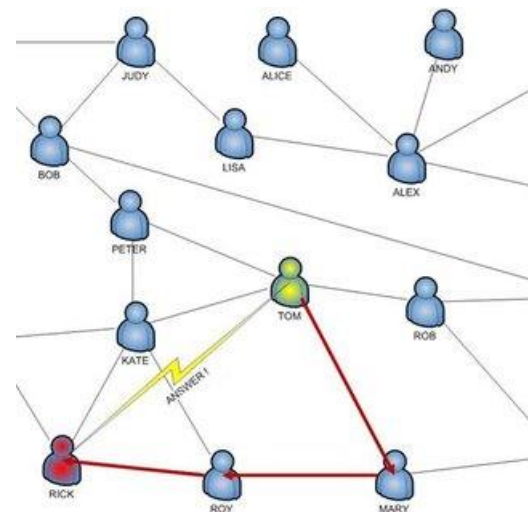
Dewasa ini, internet bukanlah hal yang biasa lagi. Hampir semua orang mengetahui apa itu internet dan menggunakannya. Internet telah berhasil membantu manusia dalam berbagai aspek, seperti pendidikan, perbankan, hiburan, dan lain-lain. Selain itu, internet juga memberikan kemudahan dalam penyampaian informasi antarmanusia. Dengan adanya sosial media, orang-orang dapat berkomunikasi, bahkan berkenalan dengan orang baru dan memperluas jaringan pertemanannya tanpa mempedulikan jarak yang terbentang.

Jika seseorang menggunakan suatu media sosial, maka ia termasuk dalam *social networking* (jaringan sosial) media sosial tersebut. Pada sebuah situs *social networking*, orang-orang saling terhubung. Hubungan antarorang ini biasanya terbentuk berdasarkan hubungan pertemanan. Pada tahun 2011, Facebook melaporkan bahwa jarak terpendek antarpengguna hanyalah sekitar 4.74, dari 721 juta pengguna di seluruh dunia dengan 69 miliar hubungan pertemanan. Hal ini berarti, antara dua pengguna di dunia, rata-rata, mereka hanya dipisahkan oleh lima orang. Pernyataan ini didukung oleh teori *Six Degree of Separation* yang mendemonstrasikan bagaimana dua orang terhubung oleh rata-rata lima orang terlebih dahulu. Ini juga memberi arti bahwa pengguna bisa saja menemukan orang yang dicarinya atau mungkin diketahuinya tetapi belum berteman melalui fitur rekomendasi teman yang disediakan.

Hampir seluruh *platform* media sosial menyediakan fitur ini. Bahkan, LinkedIn memiliki sebuah fitur bernama koneksi. Semua pengguna yang tergabung dalam sebuah jaringan (*network*) seorang pengguna disebut *connections*. LinkedIn juga mempunyai fitur *degree of connections* yang menunjukkan derajat koneksi antara seorang pengguna dengan pengguna lainnya. *Degree of connections* ini sama dengan

level pertemanan. Jika seseorang berteman atau memiliki koneksi secara langsung dengan seorang pengguna lainnya, maka pengguna tersebut memiliki gelar *1<sup>st</sup> connection* atau teman level satu dan begitu pula seterusnya untuk level-level selanjutnya.

Jaringan pertemanan pada berbagai macam jaringan sosial ini dapat direpresentasikan dalam bentuk graf. Setiap pengguna akan digambarkan dalam simpul dan hubungan pertemanan antarpengguna digambarkan menjadi sisi.



**Gambar 1.** Contoh Representasi Graf pada *Social Networking*

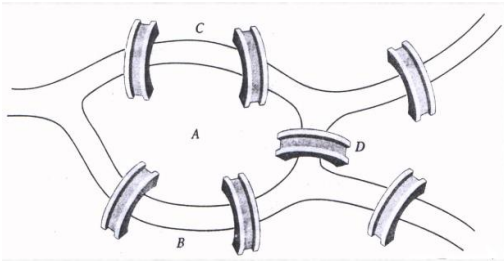
(Sumber : [https://www.researchgate.net/figure/A-sample-social-network-graph\\_fig1\\_262331004](https://www.researchgate.net/figure/A-sample-social-network-graph_fig1_262331004))

Dalam lingkup media sosial, algoritma BFS lebih sering digunakan karena hubungan yang secara langsung (level 1 yang berarti berteman atau memiliki koneksi secara langsung) lebih penting untuk dikunjungi terlebih dahulu. Secara teori, menggunakan algoritma BFS untuk mencari rekomendasi teman pada level pertemanan pun cukup efektif karena algoritma BFS akan mengunjungi seluruh simpul pada level tersebut hingga semuanya telah dikunjungi sehingga bisa mencatat semua simpul yang ada pada level tersebut.

## II. DASAR TEORI

### A. Graf

Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Graf pertama kali diperkenalkan pada 1736 pada Masalah Jembatan Königsberg.



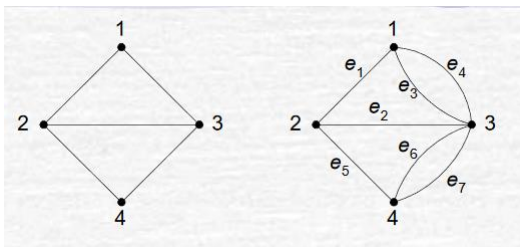
**Gambar 2.** Masalah Jembatan Königsberg

(Sumber :

[http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf))

Sebuah graf dapat direpresentasikan dengan notasi  $G = (V, E)$ . Simbol  $V$  melambangkan himpunan tidak kosong dari simpul-simpul dan simbol  $E$  melambangkan himpunan sisi yang menghubungkan sepasang simpul.

Ada beberapa jenis graf. Berdasarkan ada-tidaknya gelang atau sisi ganda pada graf, graf dibagi menjadi dua, yaitu graf sederhana dan graf tak sederhana. Graf sederhana adalah graf yang tidak memiliki gelang ataupun sisi ganda, sedangkan graf tak sederhana adalah graf yang mengandung sisi ganda atau gelang. Sisi ganda adalah sisi yang menghubungkan dua buah simpul yang sama.



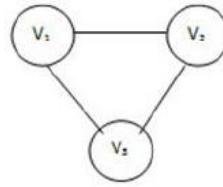
**Gambar 3.** Graf Sederhana dan Graf Tak Sederhana

(Sumber :

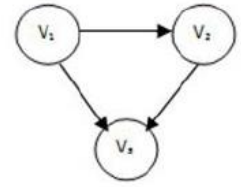
[http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf))

Selain itu, graf juga dapat digolongkan berdasarkan orientasi arah pada sisi. Penggolongannya adalah graf tak berarah dan graf berarah. Graf tak berarah (*undirected graph*) adalah graf yang sisinya tidak mempunyai orientasi arah, sedangkan graf berarah (*directed graph*) adalah graf yang setiap sisinya diberikan orientasi arah.

### Undirected Graph



### Directed Graph

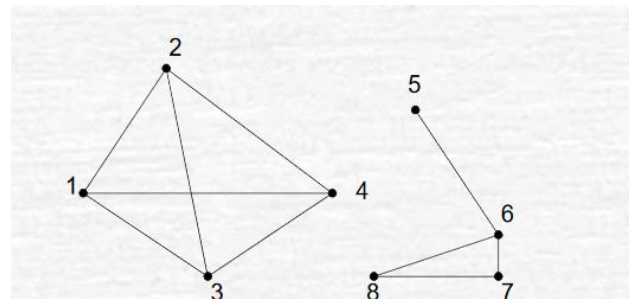


**Gambar 4.** Graf Berarah dan Graf Tidak Berarah

(Sumber : [https://www.researchgate.net/figure/Directed-and-Undirected-graph-Ref-7\\_fig2\\_282653028](https://www.researchgate.net/figure/Directed-and-Undirected-graph-Ref-7_fig2_282653028))

Ada beberapa terminologi pada graf. Salah satunya adalah ketetanggaan (*adjacent*). Ketetanggaan pada graf didefinisikan sebagai keadaan saat ada dua simpul yang memiliki hubungan langsung. Dengan kata lain, jika ada sebuah sisi yang menghubungkan dua buah simpul, maka kedua simpul tersebut dikatakan bertetangga. Pada graf, ada juga terminologi bersisian. Untuk sembarang sisi  $e = (v_j, v_k)$ , pernyataan ini memiliki arti bahwa terdapat sisi  $e$  yang menghubungkan simpul  $v_j$  dan  $v_k$ .

Selain itu, ada pula terminologi terhubung. Dua buah simpul dikatakan terhubung jika terdapat suatu lintasan (sisi yang menghubungkan) antara dua simpul tersebut. Suatu graf disebut terhubung jika terdapat lintasan antara semua pasangan simpul yang ada pada graf tersebut.



**Gambar 5.** Graf Terhubung

(Sumber :

[http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf))

Setiap simpul mempunyai tingkatan atau level. Simpul akar mempunyai level 0 dan terus bertambah seiring dengan turunnya tingkatan pencarian hingga level  $n$  yang merupakan level paling terakhir atau daun paling bawah.

Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul tersebut. Pada graf berarah, terdapat dua jenis derajat, yaitu derajat masuk dan derajat keluar.

### B. Algoritma Breadth First Search (BFS)

Algoritma traversal graf adalah algoritma yang mengunjungi simpul dengan cara sistematis. Algoritma ini digunakan sebagai bentuk pencarian solusi dari persoalan yang direpresentasikan dalam bentuk graf. Algoritma pencarian lebar atau *Breadth First Search* (BFS) adalah salah satu dari

algoritma ini. Algoritma BFS termasuk algoritma tanpa informasi. Dalam proses pencarian solusi ini, ada dua jenis pendekatan graf yang digunakan, yaitu graf statis dan graf dinamis. Graf statis adalah graf yang sudah terbentuk sebelum proses pencarian dimulai, sedangkan graf dinamis adalah graf yang mulai dibentuk saat proses pencarian dilakukan.

Algoritma BFS memiliki pendekatan graf statis yang berarti graf sudah tersedia sebelum proses pencarian dimulai. Ciri khas dari algoritma ini adalah pencarian yang dimulai dari akar dan dilanjutkan dengan pencarian bertahap level demi level untuk memeriksa seluruh simpul pada kedalaman tertentu hingga ditemukan tujuannya. Pencarian dilakukan dari kiri ke kanan. Algoritma dari BFS sendiri adalah sebagai berikut :

1. Mengunjungi suatu simpul yang dijadikan simpul awal.
2. Mengunjungi semua simpul yang bertetangga dengan simpul awal tersebut terlebih dahulu.
3. Mengunjungi simpul yang belum dikunjungi dan simpul tersebut bertetangga dengan simpul-simpul yang telah dikunjungi, dan seterusnya hingga proses pencarian selesai.

Untuk mengaplikasikan algoritma BFS, dibutuhkan beberapa struktur data. Pertama, matriks ketetanggaan yang berukuran  $n \times n$  dengan elemen yang akan bernilai 1 jika simpul  $i$  dan simpul  $j$  bertetangga dan elemen yang akan bernilai 0 jika simpul  $i$  dan simpul  $j$  tidak bertetangga. Kedua, *queue* untuk menyimpan kumpulan simpul yang akan dikunjungi. Ketiga, tabel *boolean* yang masing-masing elemen akan bernilai *true* jika simpul  $i$  sudah dikunjungi dan bernilai *false* jika simpul  $i$  belum dikunjungi. Pada setiap langkah, simpul yang sedang dikunjungi digunakan untuk mengidentifikasi simpul-simpul selanjutnya yang akan dikunjungi dengan memasukkannya ke antrian.

Berikut adalah *pseudocode* dari algoritma BFS.

```

procedure BFS(input v:integer)
{ Traversal graf dengan algoritma pencarian BFS.
Masukan: v adalah simpul awal kunjungan
Keluaran: semua simpul yang dikunjungi dicetak ke layar
}
Deklarasi
w : integer
q : antrian;

procedure BuatAntrian(input/output q : antrian)
{ membuat antrian kosong, kepala(q) diisi 0 }

procedure MasukAntrian(input/output q:antrian, input v:integer)
{ memasukkan v ke dalam antrian q pada posisi belakang }

procedure HapusAntrian(input/output q:antrian,output v:integer)
{ menghapus v dari kepala antrian q }

function AntrianKosong(input q:antrian) → boolean
{ true jika antrian q kosong, false jika sebaliknya }

Algoritma:
BuatAntrian(q)      { buat antrian kosong }
write(v)           { cetak simpul awal yang dikunjungi }
dikunjungi[v]←true { simpul v telah dikunjungi, tandai dengan true }
MasukAntrian(q,v)  { masukkan simpul awal kunjungan ke dalam antrian }

{ kunjungi semua simpul graf selama antrian belum kosong }
while not AntrianKosong(q) do
  HapusAntrian(q,v) { simpul v telah dikunjungi, hapus dari antrian }
  for tiap simpul w yang bertetangga dengan simpul v do
    if not dikunjungi[w] then
      write(w)      {cetak simpul yang dikunjungi}
      MasukAntrian(q,w)
      dikunjungi[w]←true
    endif
  endfor
endwhile
{ AntrianKosong(q) }
NUM-RN-MLK/IF2211/2013

```

**Gambar 6.** Pseudocode Algoritma BFS

(Sumber :

[http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/BFS-dan-DFS-\(2020\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/BFS-dan-DFS-(2020).pdf))

Suatu algoritma memiliki beberapa aspek yang dapat dinilai. Keuntungan dari algoritma ini adalah tidak akan menemukan jalan buntu pada saat proses pencarian. Selama ada satu solusi, maka algoritma BFS pasti akan menemukan solusi tersebut. Oleh karena itu, algoritma BFS memiliki properti *completeness* yang berarti solusi terjamin dapat ditemukan jika solusi tersebut memang ada pada graf dan nilai *branching factor* terbatas. *Branching factor* adalah maksimum percabangan yang mungkin dari suatu simpul.

Algoritma BFS juga memiliki properti *optimality* yang berarti solusi yang didapatkan terjamin optimal. Kompleksitas waktu yang dimiliki algoritma ini adalah  $O(b^d)$ , sama dengan kompleksitas ruangnya dengan  $b$  adalah *branching factor* dan  $d$  adalah *depth* yang berarti kedalaman dari solusi terbaik. Algoritma ini kurang baik dari segi kompleksitas ruang karena perlu menyimpan seluruh simpul dalam suatu graf.

### C. Jaringan Sosial (*Social Networking*)

*Social networking* adalah penggunaan media sosial yang berbasis internet untuk berkomunikasi dengan teman, keluarga, rekan kerja, klien, pelanggan (Will Kenton, 2020). Kolaborasi, berbagi ide, mengetahui kabar dari orang-orang yang dikenal, dan mencari teman baru adalah beberapa hal yang dapat dilakukan. Selain itu, *social networking* juga menjadi salah satu tempat yang paling signifikan untuk digunakan sebagai sarana pemasaran berbagai macam produk. Kegiatan-kegiatan ini biasa dilakukan pada suatu *platform* bertipe media sosial yang penggunaannya bisa mencapai jutaan orang berbeda.

*Social networking* ini dapat digunakan untuk keperluan di berbagai bidang, mulai dari keperluan sosial, keperluan bisnis, ataupun keduanya. Ada banyak situs yang menyediakan sarana untuk memenuhi keperluan ini, mulai dari Facebook, Twitter, LinkedIn, dan Instagram. Situs media sosial pertama kali dibuat pada 1996 oleh Mount dan Dan Pelson dengan nama Bolt.com, tetapi media sosial ini tidak benar-benar disebut media sosial. Media sosial pertama yang benar-benar disebut media sosial adalah SixDegrees.com yang dibuat pada 1997 oleh Andrew Weinreich.

*Social networking* dapat dimanfaatkan untuk mencari teman ataupun menambah relasi profesional (*networking*). Dalam mencari teman, media sosial memegang peranan yang cukup krusial dalam menghubungkan seseorang dengan orang lainnya, membuat mereka saling mengobrol dan mengenal satu sama lain lebih dalam lagi. Dalam *networking*, setelah berkenalan dan mengobrol dalam waktu yang cukup, orang yang baru dikenal tersebut dapat dijadikan "*networking buddy*". Koneksi *networking* ini telah diterapkan pada aplikasi LinkedIn. Pada LinkedIn, seseorang dapat mulai mencari koneksi ini dengan terkoneksi dengan seseorang yang telah dikenal dan dipercayai pada status profesional. Selanjutnya, orang tersebut akan menjadi *1<sup>st</sup> connection* dan mempunyai jaringan koneksi yang lebih luas lagi.

Meskipun tidak terlalu terlihat perbedaannya, keduanya memiliki suatu kesamaan. Baik koneksi teman maupun sebatas koneksi dari *networking*, koneksi tersebut dapat mengenalkan kita dengan orang lainnya yang ia kenal. Dari sana, tentunya koneksi kita pun dapat bertambah dan dari koneksi pertemanan tersebut, mungkin kita dapat menemukan orang yang sedang kita cari. Selain itu, pengguna yang dapat memberi pengaruh yang besar pun dapat ditemukan dengan asumsi bahwa semakin banyak *followers*, maka semakin besar pengaruhnya.

Analisis dari graf *social networking* menjadi suatu hal yang cukup penting dalam berbagai bidang. Graf ini mempunyai distribusi dan bentuk yang cukup unik sebagai karakteristiknya. Secara daring, ada fitur yang bernama *Friends Recommendation* (rekomendasi teman) atau *People You May Know* (orang yang mungkin Anda kenal). Layanan *social networking* sekarang dapat memberikan rekomendasi teman ke pengguna berdasarkan graf sosial atau pertemanan. Contohnya, Facebook menggunakan analisis hubungan sosial (*social link analysis*) antara pengguna-pengguna yang terhubung dan memiliki teman yang sama (*mutual*), lalu merekomendasikan pengguna lain sebagai kemungkinan pengguna yang diketahui.

### III. PENERAPAN ALGORITMA BFS PADA *SOCIAL NETWORKING*

Sebelumnya, berikut adalah batasan dari *insight* yang akan ditunjukkan :

1. Rekomendasi teman
2. Rute atau sekuens untuk mencapai seorang pengguna
3. Pengguna yang merupakan teman dari banyak pengguna lain

Jaringan pertemanan pada suatu *platform* dari *social networking* dapat dipetakan dalam bentuk graf. Pada graf ini, seorang pengguna direpresentasikan oleh sebuah simpul. Jika ada koneksi atau hubungan pertemanan antara dua pengguna ini, maka akan ada sebuah sisi yang menghubungkan kedua simpul yang melambangkan dua pengguna tersebut. Dengan kata lain, sebuah sisi merepresentasikan sebuah hubungan pertemanan antara dua orang.

Graf yang digunakan untuk representasi bisa berupa graf berarah maupun graf tidak berarah. Dalam kasus graf tidak berarah, jika ada sisi yang menghubungkan dua pengguna, maka kedua pengguna tersebut dianggap saling mengenal atau saling memiliki kontak masing-masing pada *list* pertemanannya. Dalam kasus graf berarah, jika ada sisi berarah yang menghubungkan dua pengguna, maka hubungan pertemanan mengikuti arah anak panah. Jika terdapat dua sisi berlawanan arah yang menghubungkan dua pengguna, maka barulah kedua pengguna tersebut dianggap saling mengenal. Untuk pengaplikasian pada program, graf direpresentasikan dalam bentuk *adjacency list*.

1. Rekomendasi Teman

Apabila diketahui sebuah graf pertemanan dari sebuah situs *social networking*, maka mungkin saja fitur rekomendasi teman ini diaplikasikan dengan algoritma

BFS. Sesuai dengan *pseudocode*-nya, pertama-tama, algoritma BFS akan melakukan traversal ke seluruh simpul yang bertetangga dengan simpul pengguna yang menjadi acuan pertama. Langkah awal dari traversal ini adalah dengan memasukkan simpul pertama itu sendiri ke dalam antrian sebagai elemen pertama. Selanjutnya, traversal ini dilakukan dengan cara memasukkan masing-masing simpul yang bertetangga dengannya ke dalam sebuah antrian.

Lalu, selama antrian tersebut tidak kosong, maka akan terjadi *loop* untuk melakukan proses hingga antrian tidak kosong dan proses pencarian telah mencapai simpul terakhir. Pada proses *loop* yang ini, pada awalnya akan mengambil elemen pertama dari antrian untuk dijadikan simpul acuan. Simpul acuan yang baru ini dapat bertetangga dengan simpul-simpul lainnya. Maka dari itu, akan dilakukan *loop* kedua untuk mengetahui simpul-simpul mana saja yang bertetangga dengan simpul ini.

Jika simpul yang bertetangga dengan simpul acuan tersebut belum dikunjungi sebelumnya, maka tabel *Boolean* untuk simpul tersebut akan diisi nilai *True* yang menandakan bahwa sekarang simpul tersebut telah dikunjungi dan tidak perlu dikunjungi lagi nantinya. Jika simpul tersebut telah dikunjungi sebelumnya, maka tidak perlu dikunjungi lagi. Simpul tersebut kemudian akan dimasukkan ke dalam antrian di elemen terakhir karena antrian yang digunakan pada algoritma BFS bersifat FIFO (*First In First Out*).

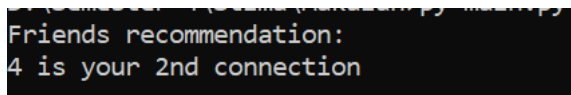
Setelah menambahkan simpul ke antrian, akan dicek level dari simpul tersebut dan disimpan ke dalam sebuah *array*. Untuk menampilkan rekomendasi pertemanan, relatif terhadap simpul acuan pertama, yang perlu diberikan hanyalah simpul-simpul (pengguna) yang memiliki level di atas level satu. Hal ini karena jika simpul tersebut memiliki level satu, maka itu berarti simpul tersebut telah terkoneksi secara langsung dengan simpul awal tersebut sehingga tidak perlu ditampilkan lagi. Namun, jika simpul tersebut memiliki level di atas level satu, hal tersebut berarti simpul tersebut tidak bertetangga (tidak terhubung secara langsung).

Algoritma BFS cocok digunakan untuk masalah ini karena perlu dilakukan pencarian secara menyeluruh dalam graf dan dimulai dari simpul-simpul yang bertetangga dengan simpul awal yang diacu. Setelah semua simpul tetangga dikunjungi dan didapatkan simpul-simpul lain yang bertetangga dengan simpul tetangga dari simpul awal yang diacu tadi, barulah akan didapatkan rekomendasi teman.

Data uji yang digunakan :

- 1 2
- 1 3
- 2 3
- 2 4
- 3 4

Hasil uji coba program dengan simpul awal adalah simpul 1 :



**Gambar 7.** Hasil Uji Coba untuk Rekomendasi Teman  
(Sumber : Penulis)

Pada kasus di atas, berikut adalah ilustrasi dari proses yang terjadi.

Iterasi	Simpul ekspan	Antrian	Level simpul ekspan
Inisialisasi	1	{1}	0
1	1	{2,3}	0
2	2	{3,4}	1
3	3	{4}	1
4	4	{}	2

Karena di sini mencari rekomendasi teman untuk simpul 1, simpul yang akan diambil adalah simpul dengan level di atas 1, relatif terhadap simpul 1. Pada kasus ini, hanya simpul 4 yang mempunyai level di atas level 1, yang berarti rekomendasi teman untuk pengguna 1 adalah pengguna 4.

### 2. Rute untuk Mencari Pengguna

Dalam mencari rute atau *path* antara dua pengguna, diperlukan struktur data tambahan, yaitu *dictionary*. Fungsi *dictionary* di sini adalah sebagai pencatat *parent* dari setiap simpul. *Parent* yang dimaksud di sini adalah simpul yang bertetangga dengan simpul yang sedang diiterasi dan dikunjungi sebelum simpul yang diiterasi itu.

Untuk proses *loop* hampir sama dengan *loop* untuk mencari rekomendasi teman, hanya terdapat sedikit tambahan. *Loop* pertama tetap akan digunakan untuk terus melakukan proses hingga antrian tidak kosong dan belum ditemukannya pengguna yang dicari. Saat memasuki *loop* ini, elemen pertama akan diambil dari antrian dan dicek apakah elemen yang diambil ini sama dengan simpul tujuan. Jika sama, maka program akan keluar dari *loop* dan menampilkan rute. Jika tidak sama, maka program akan masuk ke *loop* kedua. *Loop* kedua ini memiliki mekanisme yang sama dengan *loop* untuk rekomendasi teman.

Akan dilakukan *loop* untuk melakukan traversal ke seluruh simpul yang bertetangga dengan simpul acuan saat itu. Jika simpul tersebut belum dikunjungi, maka akan dicatat *parent* dari simpul tersebut. Lalu, tabel *Boolean* untuk simpul tersebut akan diisi *True* dan simpul tersebut akhirnya akan dimasukkan ke dalam antrian.

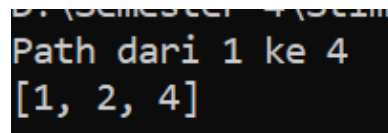
Data uji yang digunakan :

- 1 2
- 1 3
- 2 3

2 4

3 4

Hasil uji coba program :



**Gambar 8.** Hasil Uji Coba untuk Rute Pencarian  
(Sumber : Penulis)

Pada kasus di atas, berikut adalah ilustrasi dari proses yang terjadi.

Iterasi	Simpul ekspan	Antrian	Parent simpul ekspan
Inisialisasi	1	{1}	-
1	1	{2,3}	-
2	2	{3,4}	1
3	3	{4}	1
4 (Solusi)	4	{}	2

### 3. Pengguna yang merupakan teman dari banyak pengguna lain

Algoritma BFS juga dapat digunakan untuk mencari pengguna yang merupakan teman dari banyak pengguna lain. Cara mencari pengguna dengan kriteria ini adalah dengan mencari derajat keluar dari seluruh pengguna karena yang digunakan saat ini adalah graf berarah.

*Loop* pertama masih sama dengan sebelumnya, yaitu untuk memastikan proses pencarian akan terus dilakukan hingga antrian kosong atau telah mencapai simpul terakhir. Setelah mengambil elemen pertama dari antrian, akan dibuat suatu *list* untuk menyimpan seluruh simpul yang bertetangga dengan simpul yang saat itu sedang diacu.

Sesudah mendapatkan *list* tersebut, kembali dilakukan *loop* untuk menghitung derajat masuk dari simpul tersebut. Cara menghitungnya adalah dengan menyimpan seluruh *value* dari derajat masuk masing-masing simpul ke dalam *array*. Nilai tersebut tidak disimpan dalam indeks untuk simpul yang saat itu diacu, melainkan disimpan dalam indeks simpul yang bertetangga dengan simpul yang sedang diacu.

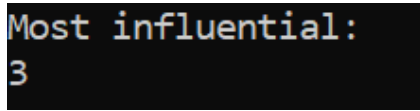
Setelah proses tersebut selesai, dilakukan lagi proses BFS, yaitu memasukkan simpul-simpul yang bertetangga dengan simpul yang sedang diacu dan belum dikunjungi ke dalam antrian.

Data uji yang digunakan :

- 1 2
- 1 3

2 3  
2 4  
2 1  
3 4  
4 3  
4 1

Hasil uji coba program :



**Gambar 9.** Hasil Uji Coba untuk Mencari Pengguna yang merupakan Teman dari Banyak Pengguna Lain

(Sumber : Penulis)

Pada kasus di atas, berikut adalah ilustrasi dari proses yang terjadi :

Iterasi	Simpul ekspan	Antrian	Derajat masuk (simpul jumlah derajat masuk) ->
Inialisasi	1	{1}	-
1	1	{2,3}	2 -> 1, 3 -> 1
2	2	{3,4}	3 -> 2, 4 -> 1
3	3	{4}	4 -> 2, 1 -> 1
4	4	{}	1 -> 2, 3 -> 3

Dapat dilihat bahwa meskipun antrian pada saat simpul 4 diekspan sudah kosong karena simpul 1 dan simpul 3 telah diekspan sebelumnya, derajat masuk dari simpul yang bertetangga dengan simpul 4 tetap harus dihitung. Proses ini tetap dilakukan karena proses untuk menghitung derajat masuk suatu simpul dilakukan terlebih dahulu dari pengecekan untuk memasukkan simpul-simpul yang bertetangga dengan simpul acuan dan belum dikunjungi ke antrian, seperti yang telah dijelaskan di atas.

## VI. KESIMPULAN

Graf dapat digunakan untuk merepresentasikan jaringan pertemanan yang ada di *social networking*. Data yang direpresentasikan dalam bentuk graf *social networking* dapat memberikan banyak *insight*, seperti memberikan rekomendasi pertemanan, mencari seorang pengguna melalui rute terpendek, dan menemukan pengguna yang merupakan teman dari banyak pengguna lainnya. Semua *insight* ini dapat

diperoleh dengan menggunakan algoritma *Breadth First Search* (BFS).

## V. VIDEO LINK DI YOUTUBE

Berikut adalah *link* ke video penjelasan yang ada di Youtube : <https://youtu.be/3BbQ6kblP8w>.

## VI. UCAPAN TERIMA KASIH

Penulis mengucapkan syukur kepada Tuhan Yang Maha Esa karena atas berkat dan rahmatnya, Penulis dapat menyelesaikan karya tulis berjudul “Penerapan Algoritma BFS dalam Memberikan Rekomendasi Teman”. Penulis juga berterima kasih kepada Ibu Nur Ulfa Maulidevi, dosen pengampu mata kuliah Strategi Algoritma Kelas 2, atas segala ilmu dan pedoman yang telah diberikan. Tak lupa, Penulis juga berterima kasih kepada keluarga dan teman-teman Penulis yang telah membantu dan memberikan semangat.

## VII. REFERENSI

- [1] Bharati, Ashank. (2019). *Social Networking with BFS and Neo4j*. <https://medium.com/gonuclei/social-networking-with-bfs-and-neo4j-ea52dc4ce198>. (diakses 25 April 2020 pukul 22.00 WIB).
- [2] Bródka, Piotr. (2012). *Web-based Knowledge Exchange through Social Links in The Workplace*. [https://www.researchgate.net/figure/A-sample-social-network-graph\\_fig1\\_262331004](https://www.researchgate.net/figure/A-sample-social-network-graph_fig1_262331004). (diakses 26 April 2020 pukul 1.06 WIB).
- [3] Computer Hope. (2019). *Social network*. <https://www.computerhope.com/jargon/s/socinew.htm>. (diakses 30 April 2020 pukul 22.04 WIB).
- [4] Hartono, Teguh. (2018). *BAB II LANDASAN TEORI*. <https://docplayer.info/72845492-Bab-ii-landasan-teori.html>. (diakses 30 April 2020 pukul 21.00 WIB).
- [5] Kenton, Will. (2020). *Social Networking*. <https://www.investopedia.com/terms/s/social-networking.asp>. (diakses 30 April 2020 pukul 21.47 WIB).
- [6] Munir, Rinaldi. (2019). *Graf*. [http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20\(2015\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf). (diakses 30 April 2020 pukul 20.28 WIB).
- [7] Munir, Rinaldi. (2020). *Breadth/Depth First Search (BFS/DFS)*. [http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/BFS-dan-DFS-\(2020\).pdf](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/BFS-dan-DFS-(2020).pdf). (diakses 30 April 2020 pukul 20.57 WIB).
- [8] Padmanab, Karthik Anantha. (2014). *Optimizing Bradth-First Search for Social Networks*. <https://www.sumologic.com/blog/optimizing-breadth-first-search-for-social-networks/>. (diakses 26 April 2020 pukul 23.14 WIB).
- [9] Remis, Luis Carlos Maria. (2016). *Breadth First Search for Social Network Graphs on Heterogenous Platforms*. <https://pdfs.semanticscholar.org/5c6b/9b1e4dd764bcd4576d4d722b3a6615cbf3d4.pdf>. (diakses 26 April 2020 pukul 21.20 WIB).
- [10] Zafarani, Reza, dkk. (2014). *Social Media Mining: An Introduction*. <http://dmml.asu.edu/smm/chapters/SMM-ch2.pdf>. (diakses 26 April 2020 pukul 23.53 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, 3 Mei 2020



Florencia Wijaya 13518020