

Tugas Besar II IF2211 Strategi Algoritma  
Semester II Tahun 2020/2021  
**Pengaplikasian Algoritma BFS dan DFS dalam Fitur *People You May Know*  
Jejaring Sosial Facebook**

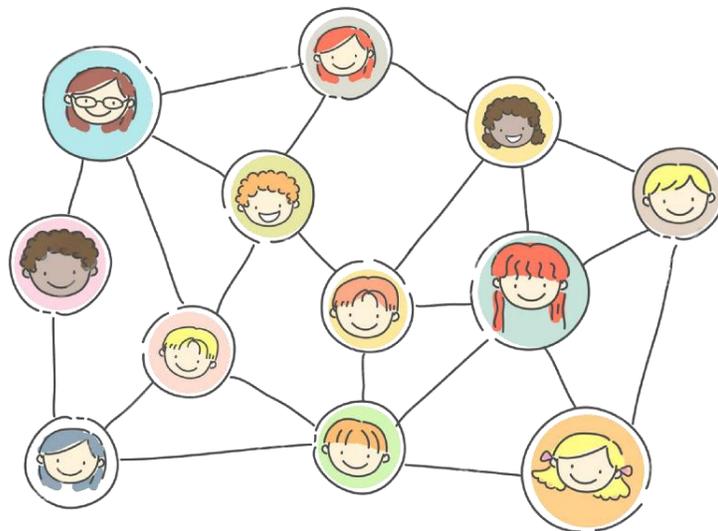
**Batas pengumpulan :** Jumat, 26 Maret 2021 sebelum pukul 23.59 WIB

**Arsip pengumpulan :**

- *Source* program yang bisa dijalankan disertai *readme.txt*
- Laporan (*soft copy*)

**Latar belakang :**

Bermain media sosial merupakan aktivitas yang sangat menyenangkan. Media sosial adalah sebuah media daring, dengan para penggunanya bisa dengan mudah berpartisipasi, berbagi, dan menciptakan isi blog, jejaring sosial, wiki, forum dan dunia virtual. Dari berbagai jenis media sosial, jejaring sosial merupakan jenis yang paling diminati oleh orang banyak. Popularitas jejaring sosial seperti facebook, twitter, Instagram, dan lainnya semakin meningkat dalam beberapa tahun terakhir. Pengguna media sosial pun berasal dari berbagai kalangan, mulai dari anak SD hingga orang tua berusia lanjut. Salah satu manfaat dari jejaring sosial adalah pengguna dapat berkomunikasi dengan orang-orang dari berbagai belahan dunia, baik orang yang sudah dikenal, maupun orang yang belum pernah dikenal sebelumnya.



Gambar 1. Ilustrasi graf pertemanan pada social network Facebook

(Sumber: <https://www.freecodecamp.org/news/deep-dive-into-graph-traversals-227a90c6a261/>)

Facebook sebagai salah satu pelopor jejaring sosial sejak tahun 2004 kini telah menembus angka 2,6 miliar pengguna. Fitur friend recommendation menjadi sangat penting karena banyaknya jumlah pengguna Facebook. Fitur bernama "People You May Know" ini dapat memberikan pengguna rekomendasi teman yang sebaiknya di-add, misalnya teman sekolah, teman kuliah, mantan pacar, atau orang yang kita kenal lewat suatu kegiatan tertentu.

Faktor utama saran pertemanan melalui fitur tersebut adalah berdasarkan *mutual friend* yang dimiliki oleh kedua akun pengguna. Misalnya pengguna A dan C belum berteman di facebook, tetapi keduanya berteman dengan pengguna B, berarti A dan C memiliki *mutual friend* yang sama, yaitu B. Semakin banyak *mutual friend* yang dimiliki antar kedua akun, maka semakin tinggi rekomendasi akun tersebut untuk di-add.

Selain itu, di setiap social media, termasuk Facebook, pengguna dapat mengeksplorasi akun-akun pengguna lainnya yang tidak memiliki mutual friends sama sekali. Akan tetapi, dengan menelusuri graf pertemanan antar akun sehingga kita dapat mengetahui 'jarak' antar akun agar bisa saling terhubung dan berteman.

### Deskripsi tugas:

Dalam tugas besar ini, Anda akan diminta untuk membangun sebuah aplikasi GUI sederhana yang dapat memodelkan beberapa fitur dari *People You May Know* dalam jejaring sosial media (*Social Network*). Dengan memanfaatkan algoritma *Breadth First Search* (BFS) dan *Depth First Search* (DFS), Anda dapat menelusuri *social network* pada akun facebook untuk mendapatkan rekomendasi teman seperti pada fitur *People You May Know*. Selain untuk mendapatkan rekomendasi teman, Anda juga diminta untuk mengembangkan fitur lain agar dua akun yang belum berteman dan tidak memiliki mutual friends sama sekali bisa berkenalan melalui jalur tertentu.

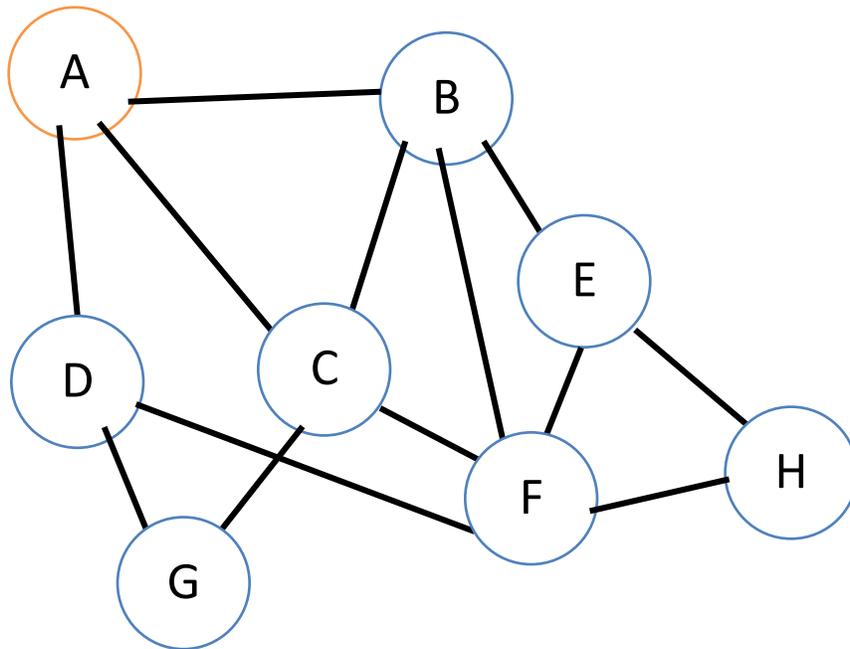
### Contoh Input dan Output Program

Contoh berkas file eksternal:

```
13
A B
A C
A D
B C
B E
B F
C F
C G
D G
D F
E H
E F
F H
```

Gambar 2. Contoh input berkas file eksternal

Visualisasi graf pertemanan yang dihasilkan dari file eksternal:



Gambar 3. Contoh visualisasi graf pertemanan dari file eksternal

Untuk **fitur friend recommendation**, misalnya pengguna ingin mengetahui daftar rekomendasi teman untuk akun A. Maka output yang diharapkan sebagai berikut

Daftar rekomendasi teman untuk akun A:

Nama akun: F

3 mutual friends:

B

C

D

Nama akun: G

2 mutual friends:

C

D

Nama akun: E

1 mutual friend:

B

Gambar 4. Hasil output yang diharapkan untuk rekomendasi akun A

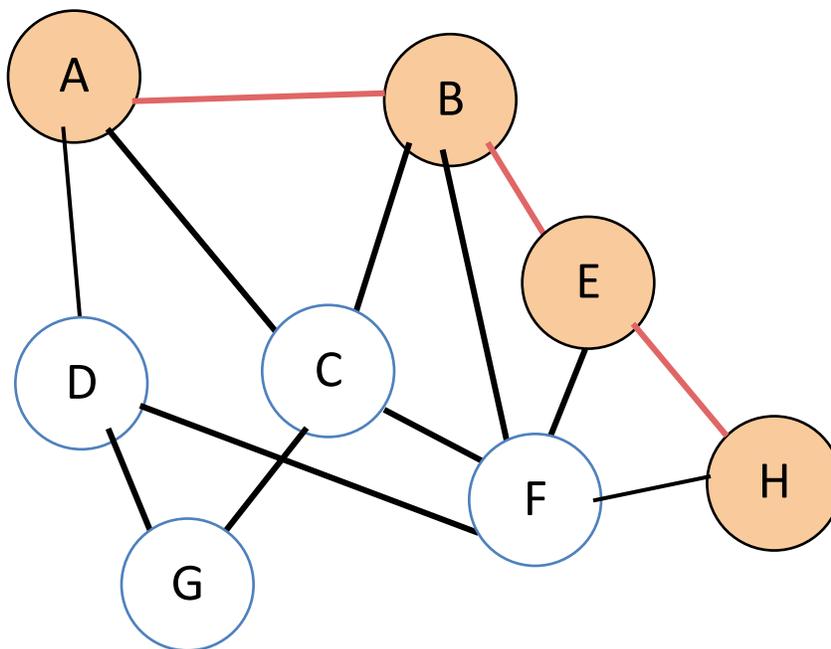
Untuk fitur **explore friends**, misalnya pengguna ingin mengetahui seberapa jauh jarak antara akun A dan H serta bagaimana jalur agar kedua akun bisa terhubung. Berikut output graf dengan penelusuran BFS yang dihasilkan.

Berikut output yang diharapkan untuk penelusuran menggunakan BFS.

```
Nama akun: A dan H
2nd-degree connection
A → B → E → H
```

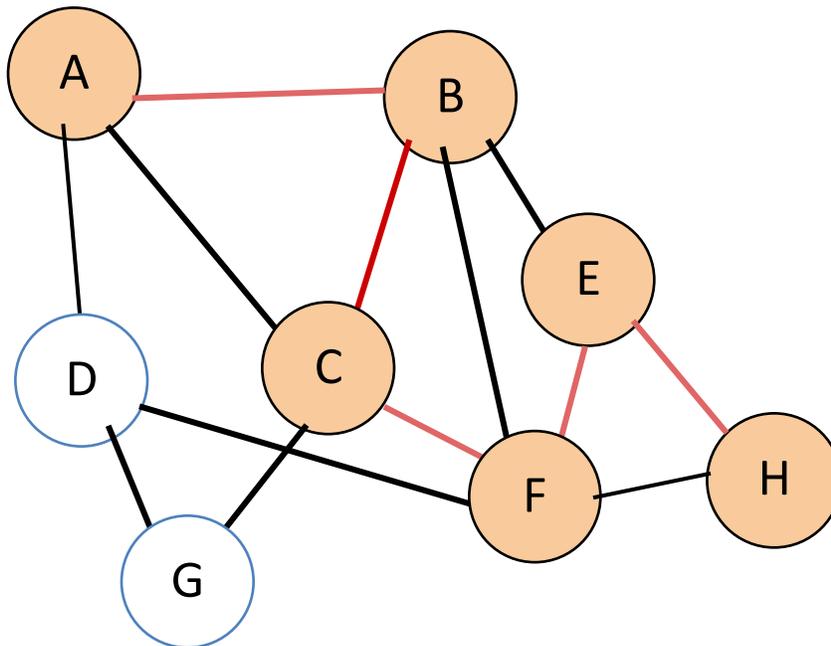
Gambar 5. Hasil output akun Nth degree connection akun A dan H menggunakan BFS

Perhatikan busur antara akun A dan H, terbentuk **salah satu jalur koneksi** sebagai berikut: A-B-E-H (ada beberapa jalur lainnya, seperti A-D-F-H, dll, urutan simpul untuk ekspan **diprioritaskan berdasarkan abjad**). Akun A dan H tidak memiliki *mutual friend*, tetapi kedua akun merupakan 2nd-degree connection karena di antara A dan H ada akun B dan E yang saling berteman. Sehingga akun H dapat terhubung sebagai teman dengan jalur melalui akun B dan akun E. Jalur koneksi dari A ke H menggunakan BFS digambarkan dalam bentuk graf sebagai berikut.



Gambar 6. Hasil visualisasi jalur koneksi menggunakan BFS

Sedangkan untuk penggunaan algoritma DFS, diperoleh jalur lainnya, yaitu A-B-C-F-E-H yang digambarkan dalam bentuk graf sebagai berikut



Gambar 7. Hasil visualisasi jalur koneksi menggunakan DFS

Pada fitur explore friends, apabila terdapat dua buah akun yang tidak bisa saling terhubung (tidak ada jalur koneksi), maka akan ditampilkan bahwa akun tersebut tidak bisa terhubung melalui jalur koneksi yang sudah dimilikinya sekarang sehingga orang tersebut memang benar-benar harus memulai koneksi baru dengan orang tersebut.

Misalnya terdapat dua orang baru, yaitu J dan I yang hanya terhubung antara J-I. Maka jalur koneksi yang dibentuk dari A ke J adalah.

Nama akun: A dan J Tidak ada jalur koneksi yang tersedia Anda harus memulai koneksi baru itu sendiri.
---

Gambar 8. Hasil output tidak ada jalur koneksi antara A dan J

### Spesifikasi Program:

Aplikasi yang akan dibangun dibuat berbasis GUI. Berikut ini adalah contoh tampilan dari aplikasi GUI yang akan dibangun.

**People You May Know**

Graph File :  Graph1.txt

Algorithm :  DFS  BFS

<Visualisasi Graph>

Choose Account :

Explore friends with :

**Friends Recommendations for A:**

- B (A -> C -> B, 1st Degree)  
2 Mutual Friends : C, D
- C  
3 Mutual Friends : E, F, G
- 
- 
- 

Gambar 9. Tampilan layout dari aplikasi desktop yang dibangun

Spesifikasi GUI:

1. Program dapat menerima input berkas file eksternal dan menampilkan visualisasi graph.
2. Program dapat memilih algoritma yang digunakan.
3. Program dapat memilih akun pertama dan menampilkan *friends recommendation* untuk akun tersebut.
4. Program dapat memilih akun kedua dan menampilkan jalur koneksi kedua akun dalam bentuk visualisasi graf dan teks bertuliskan jalur koneksi kedua akun.
5. GUI dapat dibuat **sekreatif** mungkin asalkan memuat 4 spesifikasi di atas.

Program yang dibuat harus memenuhi **spesifikasi wajib** sebagai berikut:

- 1) Buatlah program dalam bahasa **C#** untuk melakukan penelusuran *social network* facebook sehingga diperoleh daftar rekomendasi teman yang sebaiknya di-add. Penelusuran harus memanfaatkan algoritma **BFS dan DFS**.
- 2) Awalnya program menerima sebuah berkas file eksternal yang berisi informasi pertemanan di facebook. Baris pertama merupakan sebuah integer N yang adalah banyaknya pertemanan antar akun di facebook. Sebanyak N baris berikutnya berisi dua buah string

(A, B) yang menunjukkan akun A dan B sudah berteman (lebih jelasnya akan diberikan contoh pada bagian 3).

- 3) Program kemudian dapat menampilkan **visualisasi graf pertemanan** berdasarkan informasi dari file eksternal tersebut. Graf pertemanan ini merupakan **graf tidak berarah** dan **tidak berbobot**. Setiap akun facebook direpresentasikan sebagai sebuah node atau simpul pada graf. Jika dua akun berteman, maka kedua simpul pada graf akan dihubungkan dengan sebuah busur.

Proses visualisasi ini boleh memanfaatkan pustaka atau kaskas yang tersedia. Sebagai referensi, salah satu kaskas yang tersedia untuk melakukan visualisasi adalah **MSAGL** (<https://github.com/microsoft/automatic-graph-layout>) Berikut ini adalah panduan singkat terkait penggunaan MSAGL oleh tim asisten yang dapat diakses pada: <https://docs.google.com/document/d/1XhFSpHU028Gaf7YxkmdbluLkQgVI3MY6gt1t-PL30LA/edit?usp=sharing>

- 4) Terdapat dua fitur utama, yaitu:
  - a. Fitur friend recommendation
    - i. Program menerima sebuah pilihan akun dari user yang hendak dicari rekomendasi temannya. Pemilihan nama akun akan diterima melalui **GUI**. Cara pemilihan dibebaskan, bisa input dari keyboard atau meng-klik langsung sebuah node dari graf.
    - ii. Program akan menampilkan daftar rekomendasi teman seperti pada fitur *People You May Know* facebook berupa nama akun tersebut secara **terurut** mulai dari **mutual friend terbanyak** antar kedua akun beserta daftar nama akun *mutual friend*.
  - b. Fitur explore friends
    - i. Dua akun yang tidak memiliki *mutual friend*, masih memiliki peluang untuk berteman jika kedua akun mempunyai **common Nth degree connection**, yaitu **jalur** yang menghubungkan kedua akun yang terpisah sejauh N akun (node pada graf).
    - ii. Program menerima pilihan dua akun yang belum berteman.
    - iii. Program akan menampilkan nilai N-th degree connection antar kedua akun dan memberikan jalur melalui akun mana saja sampai kedua aku bisa terhubung.
    - iv. Dari graph yang sudah dibentuk, aplikasi harus dapat menyusun jalur koneksi hasil *explore friends* antara akun satu dengan akun yang ingin dituju Aplikasi juga harus dapat menunjukkan langkah-langkah pencarian, baik dengan algoritma BFS maupun DFS.
    - v. Jika tidak ditemukan jalur koneksi sama sekali antar kedua akun karena graf not fully connected, maka tampilkan informasi bahwa kedua akun tidak dapat terhubung.
- 5) Mahasiswa **tidak diperkenankan** untuk melihat atau menyalin library lain yang mungkin tersedia bebas terkait dengan pemanfaatan BFS dan DFS.

### Lain-lain:

1. Anda dapat menambahkan fitur-fitur lain yang menunjang program yang anda buat (unsur kreativitas).
2. Tugas dikerjakan berkelompok, minimal 2 orang dan maksimal 3 orang, boleh lintas kelas namun **tidak boleh sekelompok** dengan **orang yang sama dengan tubes stima sebelumnya**.
3. Semua kelompok harap mengisi data kelompok mereka pada link <http://bit.ly/PendataanStima2>
4. Program dibuat dengan bahasa C# dengan kakas Visual Studio .NET, pelajirlah C# desktop development, disarankan untuk menggunakan Visual Studio untuk mempermudah pengerjaan.
5. Program harus modular dan mengandung komentar yang jelas.
6. Beri nama aplikasi anda tersebut dengan nama-nama yang menarik dan mudah diingat.
7. Dilarang menggunakan kode program yang diunduh dari Internet. Mahasiswa harus membuat program sendiri, tetapi belajar dari program yang sudah ada tidak dilarang.
8. Batas akhir pengumpulan tugas adalah **Jumat, 26 Maret 2021 pukul 23.59 WIB**. Keterlambatan dalam mengumpulkan akan diberi penalti pengurangan skor yang cukup signifikan.
9. Semua pertanyaan menyangkut tugas ini dapat dikomunikasikan lewat QnA yang bisa diakses pada <http://bit.ly/QnATubes2>
10. **Bonus** (nilai maksimal 10): Setiap kelompok membuat video aplikasi yang mereka buat kemudian mengunggahnya ke Youtube. Video yang dibuat harus memiliki audio dan menampilkan wajah dari setiap anggota kelompok. Pada waktu demo aplikasi di depan asisten, mahasiswa mengakses video Youtube tersebut dan memutarinya di depan asisten. Beberapa contoh video tubes tahun-tahun sebelumnya dapat dilihat di YouTube dengan menggunakan kata kunci "Tubes Stima", "Tugas besar stima", "strategi algoritma", dll.
11. Demo akan dilakukan, tunggu informasi lanjut setelah waktu pengerjaan tugas berakhir.
12. Setiap anggota kelompok harus memahami seluruh program, termasuk bagian yang bukan bagian mereka.
13. Program disimpan dalam folder **Tubes2\_NIM** dengan NIM merupakan NIM anggota terkecil. Berikut merupakan struktur dari isi folder tersebut.
  - a. Folder **src** berisi **source code**
  - b. Folder **bin** berisi **executable** code / hasil build dari program C#
  - c. Folder **doc** berisi **laporan tugas besar** dengan format **nama\_kelompok.pdf**
  - d. README untuk tata cara penggunaan yang minimal berisi:
    - i. Penjelasan singkat algoritma yang diimplementasikan
    - ii. Requirement program dan instalasi tertentu bila ada
    - iii. Cara menggunakan program
    - iv. Author / identitas pembuat

Folder tersebut **di-zip** dengan format yang sama dengan nama folder. Link pengumpulan akan diberitahukan lebih lanjut oleh asisten.

### Isi laporan:

- **Cover:** Cover laporan ada foto anggota kelompok (foto bertiga). Foto ini menggantikan logo “gajah” ganesha.
- **Bab 1:** Deskripsi tugas (dapat menyalin spesifikasi tugas ini).
- **Bab 2:** Landasan Teori.
  - Dasar teori (graph traversal, BFS, DFS) secara umum
  - Penjelasan singkat mengenai C# desktop application development
- **Bab 3:** Analisis Pemecahan Masalah.
  - Langkah-langkah pemecahan masalah
  - Proses mapping persoalan menjadi elemen-elemen algoritma BFS dan DFS.
  - Contoh ilustrasi kasus lain yang berbeda dengan contoh pada spesifikasi tugas
- **Bab 4:** Implementasi dan pengujian.
  - Implementasi program (*pseudocode* program utama, **bukan kode program**).
  - Penjelasan struktur data yang digunakan dalam program dan spesifikasi program
  - Penjelasan tata cara penggunaan program (interface program, fitur-fitur yang disediakan program, dan sebagainya)
  - Hasil pengujian (*screenshot* antarmuka program dan beberapa data uji beserta skenario pengujian)
  - Analisis dari desain solusi algoritma BFS dan DFS yang diimplementasikan pada setiap pengujian yang dilakukan. Misalnya adalah apakah strategi DFS lebih baik dari BFS pada kasus kasus tertentu, dan analisis kalian mengenai mengapa hal itu bisa terjadi.
- **Bab 5:** Kesimpulan, Saran, Refleksi dan Komentar terhadap Tugas ini
- Daftar Pustaka.

### Keterangan laporan:

1. Laporan ditulis dalam bahasa Indonesia yang baik dan benar.
2. Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).

### Penilaian:

1. **Bagian 1** : Laporan (30%)
  - a. Mapping persoalan ke dalam elemen algoritma BFS dan DFS (5%)
  - b. Hasil pengujian dan analisis algoritma (10%)
  - c. Komponen-komponen lain dalam laporan (10%)
  - d. Kesesuaian laporan dengan program (5%)
2. **Bagian 2** : Implementasi Program (70%)
  - a. Kebenaran program (30%)
  - b. Pemahaman terhadap cara kerja program (25%)
  - c. Kreativitas pembuatan GUI (15%)
3. **Bagian 3** : Bonus (10%)
  - a. Membuat video demonstrasi program (10%)

--- Selamat Mengerjakan, it's not worth it if you're not having fun ---