

# Aplikasi BFS dan DFS pada Fitur Friend Suggestion di Jejaring Sosial

Filbert Reinaldha - 13510040  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13510040@std.stei.itb.ac.id

**Abstract**—Jejaring sosial adalah salah satu aplikasi yang sangat sering diakses di internet pada saat ini. Banyak fitur yang bisa menarik orang untuk menggunakan jejaring sosial. Salah satu fitur dari jejaring sosial yang biasanya selalu ada adalah fitur friend suggestion. Fitur friend suggestion merupakan fitur yang sangat menarik dari sebuah jejaring sosial. Fitur ini pun memiliki beberapa cara untuk mendapatkan list “teman-yang-mungkin-diketahui”. Salah satu caranya adalah dengan melihat 2<sup>nd</sup> grade friend.

Algoritma BFS dan DFS dapat dipakai untuk menelusuri temannya teman dari pengguna yang mungkin bisa menjadi teman lainnya bagi pengguna. Setiap teman tingkat 2 yang belum menjadi teman pengguna akan dimasukkan ke dalam sebuah struktur data list untuk kemudian ditunjukkan kepada pengguna.

**Index Terms**—Jejaring sosial, Friend-you-may-know, BFS, DFS

## I. PENDAHULUAN

Di zaman modern ini, interaksi antar manusia sudah mencapai tingkat yang lebih tinggi lagi. Teknologi yang masih berkembang secara pesat membawa internet ke kehidupan kita sehari-hari. Dengan berkembangnya internet, interaksi tidak harus selalu dilakukan secara langsung tatap muka, namun bisa melalui internet dengan berbagai fitur yang ada. Terpisahkan jarak yang sangat jauh pun sudah bukan lagi menjadi alasan seseorang untuk tidak berkomunikasi.

Ada banyak cara seseorang dapat menggunakan layanan internet untuk melakukan interaksi, salah satu contohnya adalah media sosial. Media sosial adalah interaksi antar manusia dimana mereka membuat, berbagi, bertukar dan mengomentari hal-hal seputar kehidupan mereka di sebuah komunitas virtual dan jaringan<sup>1</sup>. Yang termasuk dari banyak tipe dari media sosial adalah proyek kolaboratif seperti Wikipedia, blog seperti Blogger / Blogspot, jejaring sosial seperti Facebook, komunitas konten seperti Youtube, dan dunia virtual seperti Second Life<sup>2</sup>.

Sebuah jejaring sosial tentu memiliki kelebihan dan fitur-fitur yang dapat menarik pengguna. Friend suggestion adalah fitur yang cukup menarik yang biasanya

ada pada sebuah jejaring sosial sekarang ini. Friend suggestion merupakan cara sistem memberikan daftar pengguna lain yang belum menjadi teman pengguna namun “mungkin” adalah teman pengguna di dunia nyata. Banyak cara bagi sistem untuk menawarkan “teman” pada pengguna, misal pada jejaring sosial Facebook, ada pilihan dengan melihat kesamaan tempat tinggal, kesamaan sekolah, maupun dari temannya teman.

The image shows a vertical list of filter categories for friend suggestions on Facebook. Each category has a list of selected items and a text input field for additional options.

- Hometown**: Jakarta, Indonesia; Enter another city
- Current City**: Bandung, Indonesia; Enter another city
- High School**: SMA Labschool Jakarta; Enter another high school
- Mutual Friend**: Fadhillah Reinaldhi, Fandi Pradhana, Marchy Panggabean; Enter another name
- College or University**: Bandung Institute of Technology, Institut Teknologi Bandung; Enter another college or unive...

Gambar 1: Fitur friend-you-may-know pada jejaring sosial Facebook



Gambar 2: Fitur who to follow pada jejaring sosial Twitter

Di makalah ini, penulis akan lebih fokus pada friend suggestion dengan melihat temannya teman / 2nd grade friend yang belum menjadi teman kita. Algoritma Breadth-First Search dapat digunakan menjadi solusi dari masalah ini.

## II. DASAR TEORI

### A. Algoritma Breadth-First Search

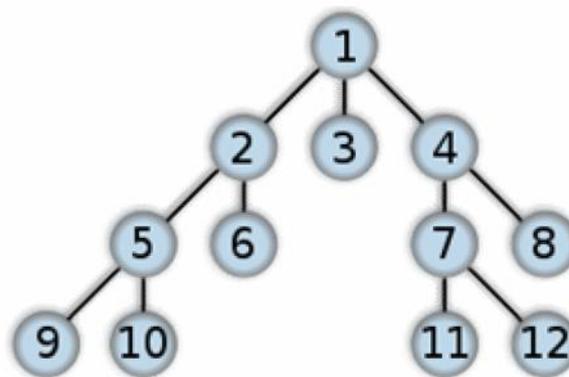
Algoritma breadth-first search merupakan metode untuk memecahkan masalah friend suggestion dengan tipe "friend's friend". Arti dari breadth-first search sendiri adalah pencarian melebar. Pada teori graf, BFS adalah strategi pencarian pada graf dimana pencarian dibatasi oleh 2 operasi, yaitu a.) kunjungi dan lihat sebuah node, dan b.) pindah ke node yang bertetangga dengan node yang sedang dikunjungi sekarang, atau dalam kata lain dalam 1 tingkat/level.

Algoritma BFS menggunakan struktur data queue untuk menyimpan hasil yang didapat dari penelusuran graf. Sistem kerjanya adalah sebagai berikut :

1. Kunjungi simpul akar, masukkan ke queue
2. Keluarkan elemen pertama pada queue, dan cek :
  - a. Jika elemen yang dicari ada pada simpul ini, hentikan pencarian dan kembalikan hasilnya
  - b. Selain itu masukkan suksesor / anak dari simpul ini yang belum dikunjungi ke queue
3. Kalau queue kosong, berarti semua simpul pada graf telah dicek. Hentikan pencarian dan kembalikan pesan tidak ketemu solusi
4. Kalau queue tidak kosong, ulangi nomor 2.

Jika graf berbentuk pohon berakar, maka semua simpul pada level  $d$  dikunjungi lebih dahulu sebelum mengunjungi simpul-simpul pada level  $d + 1$ .

Berikut adalah contoh persoalan graf yang diselesaikan dengan algoritma BFS,



Gambar 3: Contoh penelusuran graf menggunakan algoritma BFS.

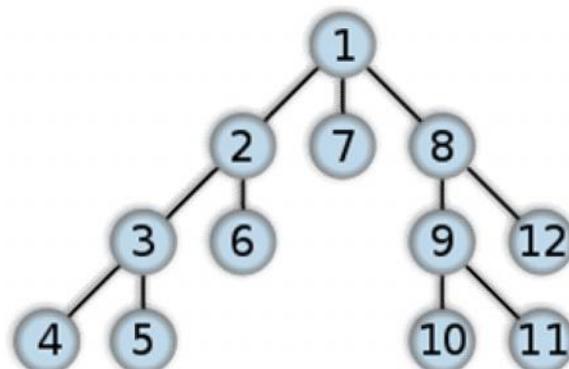
Sumber : [http://en.wikipedia.org/wiki/Breadth-first\\_search](http://en.wikipedia.org/wiki/Breadth-first_search)

### B. Algoritma Depth-First Search

Sama seperti algoritma BFS, algoritma DFS merupakan salah satu algoritma penelusuran graf. Cara kerja algoritma DFS adalah sebagai berikut :

1. Kunjungi simpul  $v$
2. Catat anak-anak dari simpul  $v$
3. Masuk ke salah satu anak simpul  $v$ , sebut  $w$ , catat lagi anak dari simpul  $w$ . Ulangi terus DFS dari simpul  $w$
4. Ketika mencapai simpul  $u$  sedemikian rupa hingga semua simpul yang bertetangga dengannya telah dikunjungi, pencarian diruntut balik (*backtrack*) ke simpul terakhir yang dikunjungi sebelumnya dan mempunyai simpul  $w$  yang belum dikunjungi
5. Pencarian berakhir bila tidak ada lagi simpul yang belum dikunjungi yang dapat dicapai dari simpul yang telah dikunjungi.

Berikut adalah contoh persoalan graf yang diselesaikan dengan algoritma DFS,



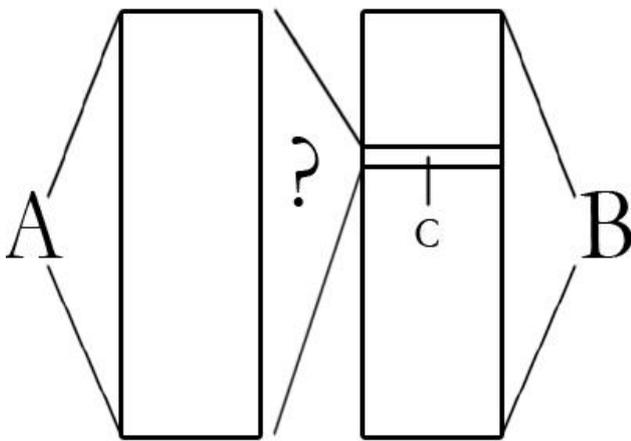
Gambar 4: Contoh penelusuran graf menggunakan algoritma DFS

Sumber : [http://en.wikipedia.org/wiki/Depth-first\\_search](http://en.wikipedia.org/wiki/Depth-first_search)

### III. METODE PENERAPAN ALGORITMA

#### A. Algoritma Brute Force

Penggunaan algoritma brute force adalah dengan membandingkan satu-satu temannya teman pengguna dengan daftar teman pengguna. Kita juga membutuhkan struktur data list untuk menyimpan data teman yang belum dimiliki pengguna, misal dengan nama "people-to-add". Misal pengguna dinotasikan dengan A dan teman pengguna dinotasikan dengan B. Pertama-tama kita list semua teman dari B. Kemudian, kita lakukan pencarian teman B, misal kita notasikan dengan C, pada list teman A. Jika tidak ada pada list teman A, maka kita tambahkan C pada list people-to-add, kemudian lanjutkan pencarian dengan elemen list teman B berikutnya. Berikut ilustrasi pencariannya,



Gambar 5: Ilustrasi pencarian dalam penggunaan algoritma Brute Force

#### B. Algoritma BFS

Pencarian menggunakan algoritma BFS memerlukan representasi dalam struktur graf dari teman pengguna. Diasumsikan graf yang terbentuk sudah memenuhi beberapa kriteria. Bila pengguna dinotasikan lagi dengan A dan teman pengguna dinotasikan dengan B, maka graf yang terbentuk harus memenuhi kriteria berikut :

1. A kita set sebagai simpul teratas pada graf / simpul akar.
2. B sebagai teman dari pengguna adalah simpul-simpul dibawah akar / simpul level 2
3. Cek semua teman B,
  - a. Jika sudah terdaftar menjadi salah satu simpul level 2, lewat
  - b. Jika belum terdaftar, tambah menjadi simpul level 3 dibawah simpul B yang sedang diperiksa

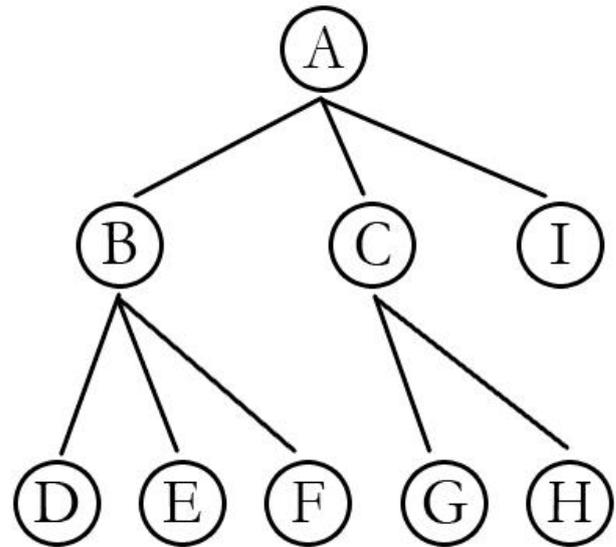
Kita ambil contoh sebagai berikut :

1. A berteman dengan B, C, dan I

2. B berteman dengan A, C, D, E, F

3. C berteman dengan A, B, G, H

Maka graf yang terbentuk dapat digambarkan sebagai berikut,



Gambar 6: Ilustrasi Graf untuk penyelesaian algoritma BFS

Dengan graf yang sudah merepresentasikan kebutuhan awal dari algoritma BFS, selanjutnya untuk mencari daftar teman yang mungkin kita tahu adalah dengan menelusuri simpul-simpul level 3. Hasil telusuran dapat disimpan dalam sebuah struktur data seperti list. Sehingga pada contoh diatas, pengguna-pengguna yang mungkin diketahui tapi belum berteman dengan pengguna A adalah D, E, F, G, dan H.

#### C. Algoritma DFS

Pencarian menggunakan algoritma DFS juga memerlukan representasi dalam struktur graf dari teman pengguna. Berbeda dengan BFS yang sebelum digunakan, graf yang dipakai harus sudah memenuhi kriteria, graf untuk algoritma DFS menuliskan semua teman yang ada. Sehingga simpul yang digambarkan untuk graf pertemanan akan sangat banyak. Sistem pengecekannya sebenarnya mirip dengan Brute Force, namun hanya menggunakan 2 list yaitu list teman langsung dari pengguna dan list 2<sup>nd</sup> grade friend.

Untuk setiap simpul pada level 2, masukkan ke list teman langsung dari pengguna. Setiap algoritma DFS sampai di simpul yang tidak punya anak (simpul daun), pengecekan dilakukan antara simpul tersebut dengan elemen-elemen pada list teman langsung dari pengguna. Jika tidak ada, maka masukkan ke dalam list 2<sup>nd</sup> grade friend.

#### IV. ANALISIS PENERAPAN ALGORITMA

##### A. Algoritma Brute Force

Pencarian menggunakan algoritma Brute Force membutuhkan waktu yang cukup lama karena setiap elemen pada list teman B harus dibandingkan 1:1 dengan setiap elemen pada list teman A. Jika A memiliki 50 list teman, dan B memiliki 100 list teman, maka perbandingan yang dilakukan adalah  $50 \times 100 = 5000$ .

##### B. Algoritma BFS

Pencarian dengan algoritma BFS membutuhkan waktu yang relatif lebih cepat daripada algoritma Brute Force karena penyusunan graf BFS sudah terurut, sehingga perbandingan yang dilakukan untuk mencari apakah teman B ada di simpul-simpul level 2 dapat dihentikan jika telah melewati indeks seharusnya teman B itu berada.

##### C. Algoritma DFS

Untuk pencarian menggunakan algoritma DFS, waktu yang dibutuhkan akan lebih lama dari BFS, kira-kira hampir sama dengan Brute Force, karena sistem kerjanya mirip dengan Brute Force, namun penggunaan memorinya lebih hemat karena tidak menggunakan banyak list.

#### V. KESIMPULAN

1. Penggunaan algoritma Brute Force untuk pencarian "friend-you-may-know" bertipe 2<sup>nd</sup> grade friend akan memakan waktu yang lama karena harus membandingkan 1:1 setiap kemungkinan
2. Algoritma BFS dilain sisi akan jauh lebih cepat karena representasi graf sudah terurut, sehingga pencarian bisa dihentikan jika elemen tidak ditemukan pada posisi yang seharusnya.
3. Algoritma DFS lebih lambat dari BFS, namun bisa menjadi alternatif algoritma Brute Force untuk meningkatkan efisiensi penggunaan memori

#### REFERENSI

- [1] Ahlqvist, Toni; Bäck, A., Halonen, M., Heinonen, S (2008). "Social media roadmaps exploring the futures triggered by social media". *VTT Tiedotteita - Valtion Teknillinen Tutkimuskeskus* (2454): 13.
- [2] Kaplan Andreas M., Haenlein Michael, (2010), Users of the world, unite! The challenges and opportunities of social media, *Business Horizons*, Vol. 53, Issue 1 (page 62-64)
- [3] <http://www.cse.ohio-state.edu/~gurari/course/cis680/cis680Ch14.html#QQ1-46-92>
- [4] Munir, Rinaldi, "Diktat Kuliah IF3051 Strategi Algoritma", Program Studi Teknik Informatika, 2009.

#### 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, 21 Desember 2012



Filbert Reinaldha - 13510040