

Aplikasi Algoritma Bidirectional Breadth First Search pada Fitur Degree pada Connections dalam aplikasi Social Networking LinkedIn

Fabianus Harry Setiawan - 13518144
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13518144@std.stei.itb.ac.id

Abstrak — Sosial Media merupakan suatu jaring yang sangat luas, dan tentunya banyak sekali potensi koneksi yang bisa kita dapatkan dalam bermain Social Media. LinkedIn merupakan salah satu social media yang memungkinkan kita untuk menjalin koneksi dengan orang lain secara profesional. Dalam LinkedIn, terdapat fitur Connections, dimana kita bisa menjalin koneksi dan “berteman” dengan orang lain. Dan pada fitur Connections tersebut terdapat Degree, yang implementasinya dibuat menggunakan Bidirectional Breadth First Search

Keywords— Social Media, LinkedIn, Networking, Connection, Path, Degree

I. PENDAHULUAN

Breadth First Search merupakan salah satu algoritma yang paling terkenal yang seringkali digunakan untuk melakukan pencarian solusi optimal. Bidirectional Breadth First Search merupakan pengembangan dari algoritma BFS, yang bisa memangkas waktu pemrosesan dengan cara melakukan 2 BFS secara simultan. Algoritma ini seringkali digunakan untuk memecahkan persoalan optimasi yang ada pada kehidupan sehari – hari. Bahkan perusahaan – perusahaan besar yang ada di dunia ini masih sering menggunakan algoritma ini untuk memecahkan persoalan yang ada di perusahaan mereka. Salah satu perusahaan tersebut adalah LinkedIn.

LinkedIn merupakan salah satu aplikasi Social Networking yang paling banyak digunakan di dunia. Aplikasi ini sudah digunakan oleh lebih dari 600 juta orang di dunia. Fungsi dari Aplikasi ini adalah membangun koneksi profesional agar orang – orang bisa lebih produktif dan sukses. Dalam aplikasi ini, juga terdapat banyak sekali *job vacancy* yang membuat linkedin juga menjadi salah satu platform untuk mencari pekerjaan.

Dalam LinkedIn, seseorang bisa menjalin koneksi dengan orang lain melalui fitur Connections. Fitur Connections memungkinkan seseorang melihat rekomendasi orang yang mungkin dikenal yang sudah disusun oleh algoritma linkedin dan menjalin koneksi dengan orang tersebut.

Pada Fitur Connections juga, terdapat fitur tambahan degree, yang dapat melihat seberapa jauh koneksi seseorang dengan orang lain. Fitur degree ini memanfaatkan Algoritma Bidirectional Breadth First Search dalam Implementasinya, dimana Node menandakan user yang menggunakan linkedin, dan hubungan pada graph menandakan bahwa user tersebut

terkoneksi dengan user pada node tujuan. Dengan menggunakan Algoritma Bidirectional Breadth First Search, LinkedIn dapat melihat path terpendek antara satu orang dengan orang lainnya, dan memanfaatkan Panjang path itu untuk kepentingan fitur Degree yang ada pada platform mereka.

II. TEORI DASAR

A. Algoritma Breadth First Search (BFS)

Breadth First Search merupakan salah satu algoritma pencarian traversal yang memanfaatkan penggunaan struktur graf untuk mencari solusi optimal dari suatu persoalan. Algoritma ini melakukan pencarian dan penghidupan simpul secara bertingkat. Algoritma ini tergolong merupakan algoritma pencarian tanpa Informasi tambahan. Algoritma ini juga menerapkan konsep FIFO (*First In First Out*) untuk simpul – simpulnya.

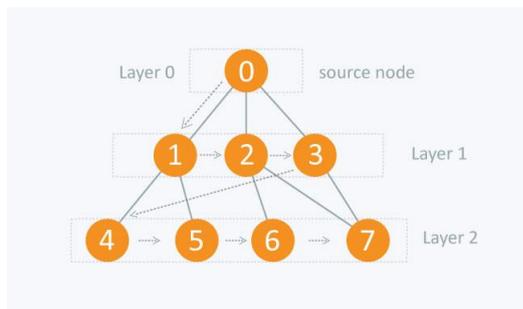
Terdapat beberapa komponen yang ada di dalam algoritma Breadth First Search, yaitu :

1. *Node*
Menandakan state – state yang dikunjungi dan diproses
2. *Queue*
Merupakan tools untuk menentukan urutan state yang akan diproses
3. *Goal State*
Menandakan State akhir yang harus dicapai (state tujuan)
4. *Visited Flag*
Tidak menutup kemungkinan bahwa ada lebih dari satu simpul yang memiliki hubungan dengan simpul yang sama. Oleh karena itu, seringkali dalam algoritma ini, dimanfaatkan suatu flag visited di setiap simpul, dimana ketika simpul tersebut sudah dikunjungi, maka flag visited diubah menjadi True untuk mencegah simpul yang sudah dikunjungi, kembali dikunjungi dikemudian waktu.

Berikut Langkah – Langkah yang harus dilakukan dalam Algoritma BFS :

1. Buatlah Queue dan masukkan node akar ke dalam Queue
2. Pop simpul dari Queue,

3. Apabila simpul yang di pop merupakan simpul solusi, maka stop pencarian
4. Apabila simpul bukan merupakan simpul solusi, maka masukkan semua tetangga ke dalam queue sesuai dengan urutan.
5. Kembali ke Langkah 2



Gambar 2.1. Breadth First Search

Sumber :

<https://www.hackerearth.com/de/practice/algorithms/graphs/breadth-first-search/tutorial/>

B. Algoritma Bidirectional Breadth First Search

Bidirectional Breadth First Search merupakan pengembangan dari algoritma Breadth First Search, dimana algoritma ini dilakukan dengan cara melakukan 2 Proses BFS sekaligus, satu dilakukan dari simpul awal, dan satu lagi dilakukan dari simpul tujuan. Algoritma ini akan berhenti apabila terdapat simpul yang sama yang dihidupkan dari kedua proses BFS tersebut.

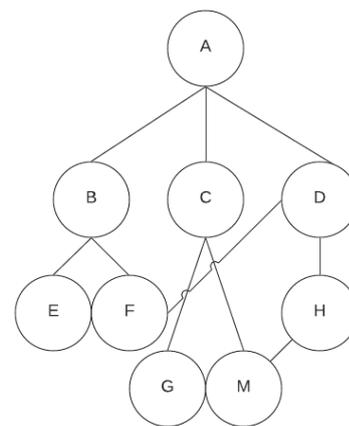
Langkah – Langkah Untuk Algoritma Bi-BFS, yaitu :

1. Buatlah 2 penghitungan BFS, dengan akar graph yang pertama dari start state dan akar graph yang kedua dari goal state
2. Masukkan masing – masing akar ke masing – masing queue.
3. Pop masing – masing simpul dari masing – masing queue. Kemudian masukkan semua simpul yang berhubungan dengan simpul yang dihidupkan ke dalam queue masing – masing.
4. Cek apakah ada anggota queue dari queue 1 dan queue 2 yang sama, apabila ada yang sama, gabungkan path dari akar ke simpul yang ditemukan pada queue 1 dengan path dari simpul yang ditemukan pada queue 2 ke goal.
5. Apabila tidak ada yang sama, Kembali ke Langkah 3

C. Perbandingan Breadth First Search dan Bidirectional Breadth First Search

1. Simulasi Graph

Misalkan diberikan graph seperti gambar di bawah ini, dengan start state dari simpul A dan goal state simpul H

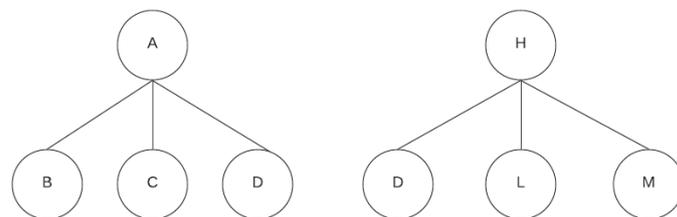


Gambar 2.2. Graph Simulasi

Algoritma BFS akan melakukan pencarian dengan urutan berikut ini :

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \rightarrow M \rightarrow H$

Sedangkan, Untuk algoritma Bi-BFS, akan terbentuk graph sebagai berikut



Gambar 2.3. Graph Simulasi Bi-BFS

Dengan urutan pencarian sebagai berikut :

- A (start state) \rightarrow membangkitkan BCD, H (goal state) \rightarrow membangkitkan DLM
- Cek apakah ada yang sama ketika membangkitkan Kedua Simpul
- Karena D sama, konstruksi path dari start state ke D dan dari Goal State ke D, lalu lakukan merge.
- $A \rightarrow D, H \rightarrow D$, merge keduanya menjadi $A \rightarrow D \rightarrow H$

2. Kompleksitas Algoritma

Untuk Algoritma Breadth First Search, Kompleksitas algoritmanya adalah $O(k^q)$, dan untuk Algoritma Bidirectional Breadth First Search, Kompleksitas Algoritmanya adalah $O(k^{q/2} + k^{q/2})$, yang dapat disederhanakan menjadi $O(k^{q/2})$. Dengan k adalah jumlah simpul tetangga (diasumsikan sama) dan q adalah Panjang path dari akar ke goal.

Dari kompleksitas Algoritmanya, Algoritma Bi-BFS akan jauh lebih cepat jika dibandingkan dengan algoritma BFS. Sehingga, algoritma ini akan jauh lebih efektif digunakan dalam skala yang besar.

3. Keuntungan dan Kerugian

• Breadth First Search

Keuntungan :

- Proses Validasi Goal State yang cepat, karena hanya memerlukan validasi berupa menyamakan goal state dengan simpul yang dihidupkan

Kerugian :

- Dibutuhkan lebih banyak pencarian dan penghidupan simpul, akan sangat berdampak untuk pencarian dengan skala yang besar

• Bidirectional Breadth First Search

Keuntungan :

- Pencarian yang dibutuhkan akan lebih sedikit, mempercepat *processing time* apabila dilakukan pada skala yang besar

Kerugian :

- Proses Validasi Goal State sulit, karena harus validasi melalui pencarian suatu elemen pada queue, lalu kemudian dilakukan merge path.

D. Aplikasi Social Networking LinkedIn



Gambar 2.4. Logo Aplikasi Social Networking LinkedIn

Sumber : play.google.com

LinkedIn merupakan salah satu Professional Social Networking Application yang paling banyak digunakan di dunia. Penggunaannya terhitung saat tanggal makalah ini ditulis sudah mencapai 645 juta orang. LinkedIn mulai dirancang pada tahun 2002 dan dirilis resmi pada tanggal 5 Mei 2003. Perusahaan ini saat ini dipegang oleh **Jeff Weiner** yang menjabat sebagai CEO.

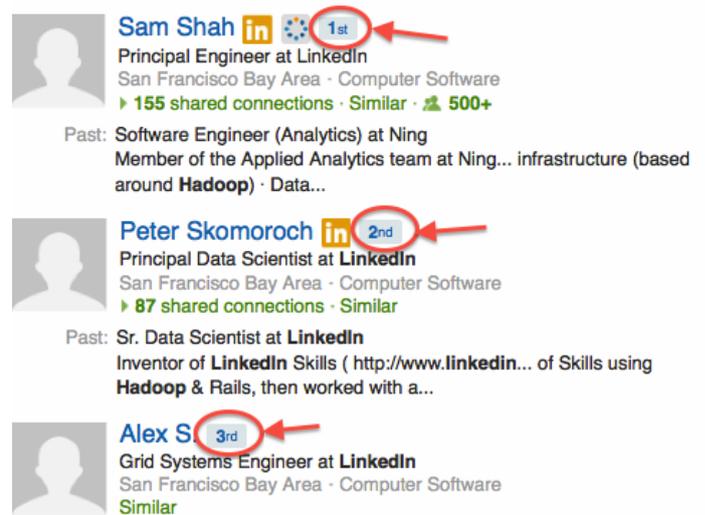
Dalam LinkedIn, kita bisa bertemu dan menjalin koneksi dengan orang – orang di luar berdasarkan minat dan bidang profesional. Kita juga dapat mencari lowongan kerja di dalam platform ini, karena banyaknya perusahaan yang menawarkan lowongan pekerjaan. Kita juga bisa mendapatkan banyak sekali ilmu baru yang berkaitan dengan dunia profesional, seperti bagaimana membuat CV yang benar, atau pertanyaan apa saja yang sering ditanyakan saat wawancara kerja. Di aplikasi ini juga, orang – orang dapat berbagi pengalaman yang menarik yang ia alami mengenai hidupnya, dan orang lain pun dapat melihat dan belajar dari pengalaman tersebut.

E. Fitur LinkedIn Connections

Pada LinkedIn, Connections adalah sebuah fitur dimana seseorang bisa menjalin koneksi dengan orang lainnya, sehingga bisa berkomunikasi dan melihat aktivitas dari orang tersebut

beserta dengan hal – hal yang diunggah oleh orang tersebut ke dalam LinkedIn.

F. LinkedIn Connection's Degree



Gambar 2.5. Connection's Degree LinkedIn

Sumber : <https://engineering.linkedin.com/real-time-distributed-graph/using-set-cover-algorithm-optimize-query-latency-large-scale-distributed>

Dan hal inilah yang menjadi fokus utama dari makalah ini, yaitu persoalan Degree dalam LinkedIn, dimana untuk fitur degree ini digunakan algoritma Bidirectional Breadth First Search. Namun, sebenarnya Apa itu Degree?

Degree adalah suatu fitur yang menyatakan keeratan hubungan seseorang dengan orang lain. Degree dibagi berdasarkan hubungan antara akun LinkedIn User dengan akun LinkedIn orang lain. Pada Gambar 2.5. (diambil dari google), tertulis degree dari user linkedin lain, yang ditandai dengan lingkaran berwarna merah. Pada LinkedIn, Degree dibagi menjadi 3 kategori, yaitu 1st, 2nd, dan 3rd.

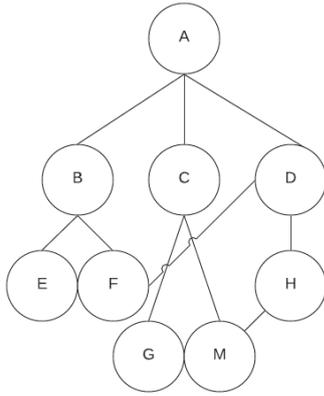
• 1st Degree Connection

1st Degree Connection adalah orang yang terhubung langsung dengan kita. Hal tersebut menandakan bahwa orang ini adalah orang yang kita kenal dan mengenal kita juga, sehingga dapat dipercaya. Apabila kita memutuskan untuk connect ke seseorang dan seseorang tersebut menerimanya, atau apabila seseorang mengirimkan permintaan untuk connect dan kita terima, maka secara otomatis orang tersebut akan langsung menjadi 1st Degree Connection kita.

• 2nd Degree Connection

2nd Degree Connection adalah orang yang terhubung dengan koneksi kita dan tidak terhubung langsung dengan kita. Karena linkedin memiliki graf koneksi untuk setiap user di linkedin, melakukan pemrosesan untuk mencari 2nd Degree Connection tentunya dapat dilakukan dengan mudah.

Mari lihat contoh simulasi berikut ini :



Gambar 2.6. Simulasi 2nd Degree Connection

Pada graf ini, user disimbolkan sebagai simpul A. Contoh 1st Degree Connection dari A adalah B,C, dan D. Contoh 2nd Degree Connection disini adalah E,F,G,M,dan H. Perlu diperhatikan bahwa walaupun D merupakan koneksi dari F yang merupakan 2nd Degree Connection dari A, bukan berarti D merupakan 3rd Degree Connection dari A, dikarenakan ada path yang lebih pendek dari A ke D, yaitu mereka terhubung langsung. Jadi, linkedin akan menentukan Degree Connection secara otomatis melalui jarak terpendek dari pemrosesan Bi-BFS yang dilakukan.

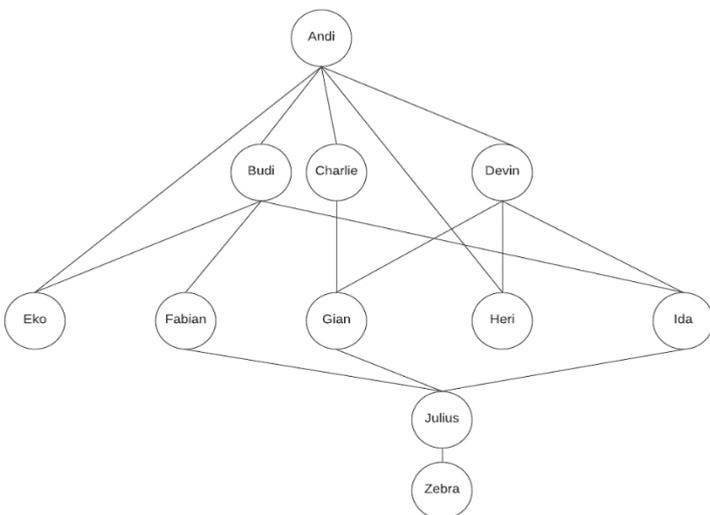
- 3rd Degree Connection

3rd Degree Connection adalah orang yang terhubung dengan 2nd Degree Connection kita, namun tidak terhubung dengan satu pun 1st Degree Connection kita. Dengan kata lain, jarak atau path terpendek dari kita ke orang ini adalah melalui 2 orang lainnya di tengah – tengah kita.

III. APLIKASI ALGORITMA BIDIRECTIONAL BREADTH FIRST SEARCH PADA FITUR CONNECTIONS DALAM APLIKASI SOCIAL NETWORKING LINKEDIN

1. Simulasi Sederhana

Untuk melakukan simulasi, saya membuat suatu struktur graf sebagai berikut :



Gambar 3.1. Simulasi Graph Koneksi LinkedIn

Pada graf ini, *node* menyatakan pengguna LinkedIn dan *edge* menyatakan koneksi antara 2 user. Diimplementasikan algoritma Bidirectional Breadth First Search dengan pseudocode sebagai berikut :

```
Function findPathBiBFS(HashMap<Integer,
Person> people, int source, int
destination)
{
    BFSData sourceData = new BFSData
    (people.get(source));
    BFSData destData = new BFSData
    (people.get(destination));

    while (!sourceData.isFinished() &&
    !destData.isFinished())
    {
        /* Search from source. */
        Person collision = search(people,
        sourceData, destData);
        if (collision != null) then
            → merge(sourceData, destData,
            collision.getID());

        /* Search from destination. */
        collision = search(people, destData,
        sourceData);
        if (collision != null) then
            → merge(sourceData, destData,
            collision.getID());
    }

    → null;
}
```

Dan dari algoritma tersebut, saya mencoba beberapa test case, dengan start state yang sama yaitu dari sudut pandang Andi sebagai user, didapatkan hasil sebagai berikut :

1. Andi – Budi

Yang pertama, adalah hubungan antara Andi dan Budi, dimana hasilnya seharusnya menjadi 1st Degree Connection karena Andi terhubung langsung dengan Budi, dan berikut hasil Test Casenya

```
Source : Andi
Destination : Budi
Path dari Start ketika Collision :
Andi --> Budi
Path Ke Goal ketika collision :
Budi
Path keseluruhan :
Andi --> Budi
Budi adalah 1st degree Connection dari Andi
```

Gambar 3.2. Hasil Test Case 1, Andi – Budi

Program akan memproses 2 path, yaitu path dari Start dan path dari Goal (Bidirectional), lalu akan berhenti ketika path tersebut mengalami collision dan menemukan iterasi yang sama. Kemudian, program akan melakukan merge path seperti dilihat pada gambar 3.2.

2. Andi – Heri

Test case yang kedua adalah hubungan antara Andi dan Heri. Walaupun Heri merupakan koneksi dari devin, yang merupakan koneksi dari Andi, namun, karena Heri terkoneksi langsung dengan Andi, seharusnya Heri tetap menjadi 1st Degree Connection dan bukannya 2nd Degree Connection. Berikut hasil test casenya

```
Source : Andi
Destination : Heri
Path dari Start ketika Collision :
Andi --> Heri
Path Ke Goal ketika collision :
Heri
Path keseluruhan :
Andi --> Heri
Heri adalah 1st degree Connection dari Andi
```

Gambar 3.3. Test Case 2, Andi – Heri

3. Andi – Fabian

Yang ketiga adalah hubungan antara Andi dan Fabian, dimana Fabian merupakan koneksi dari Budi yang merupakan koneksi dari Andi, sehingga, seharusnya Fabian adalah 2nd Degree Connection dari Andi. Berikut hasil test casenya

```
Source : Andi
Destination : Fabian
Path dari Start ketika Collision :
Andi --> Budi
Path Ke Goal ketika collision :
Budi --> Fabian
Path keseluruhan :
Andi --> Budi --> Fabian
Fabian adalah 2nd degree Connection dari Andi
```

Gambar 3.4. Test Case 3, Andi – Fabian

4. Andi – Julius

Test Case yang keempat adalah hubungan antara Andi dan Julius, dimana Path minimal yang harus ditempuh dari Andi ke Julius berjarak 3, yang membuat Julius menjadi 3rd Degree Connection dari Andi. Berikut hasil dari test casenya

```
Source : Andi
Destination : Julius
Path dari Start ketika Collision :
Andi --> Charlie --> Gian
Path Ke Goal ketika collision :
Gian --> Julius
Path keseluruhan :
Andi --> Charlie --> Gian --> Julius
Julius adalah 3rd degree Connection dari Andi
```

Gambar 3.5. Test Case 4, Andi – Julius

5. Andi – Zebra

Test Case berikutnya adalah hubungan antara Andi dengan Zebra, dimana Path minimal yang harus ditempuh dari Andi ke Zebra berjarak 4. Berikut hasil test casenya

```
Source : Andi
Destination : Zebra
Path dari Start ketika Collision :
Andi --> Budi --> Fabian
Path Ke Goal ketika collision :
Fabian --> Julius --> Zebra
Path keseluruhan :
Andi --> Budi --> Fabian --> Julius --> Zebra
Koneksi dengan Zebra terlalu jauh
```

Gambar 3.6. Test Case 5, Andi – Zebra

LinkedIn membatasi Degree hanya sampai pada 3rd Degree Connection, sehingga untuk hubungan antara 2 orang yang lebih jauh dari itu, tidak akan diproses oleh LinkedIn, sehingga, karena pada simulasi ini ada 3 orang di antara andi dan zebra yang menghubungkan mereka, yang apabila ada seharusnya menjadi 4th Degree Connection, Koneksi mereka dianggap terlalu jauh dan diabaikan oleh LinkedIn.

2. Penerapan Algoritma Skala LinkedIn

Bidirectional Breadth First Search hanya merupakan algoritma untuk menentukan degree dari suatu user ke user lain. Namun, tentunya masih banyak sekali algoritma optimasi yang digunakan oleh LinkedIn dalam mendukung degree ini. Untuk memperlihatkan rekomendasi orang yang muncul, tentunya ada algoritmanya sendiri, seperti filter berdasarkan kebangsaan, gender, dan yang lainnya.

Sebagai tambahan juga, Walaupun Breadth First Search biasanya menggunakan suatu flag untuk menentukan apakah node tertentu sudah dikunjungi, tentunya hal tersebut tidak bisa dilakukan pada skala besar, karena pencarian yang terjadi Bersama – sama sangatlah memungkinkan, maka kita dapat mengganti fungsi dari flag tersebut dengan cara menggunakan hash map, sehingga pencarian bisa dijalankan Bersama – sama tanpa harus terganggu oleh flag yang terus berganti – ganti.

IV. KESIMPULAN

Algoritma Bidirectional Breadth First Search merupakan algoritma yang digunakan untuk menyelesaikan persoalan optimasi. Algoritma Bidirectional Breadth First Search bisa menemukan solusi dengan waktu yang lebih cepat dibandingkan dengan Algoritma Breadth First Search biasa, dikarenakan kompleksitas algoritmanya yang lebih ringkas. Algoritma ini akan selalu menghasilkan solusi yang optimal, dalam kasus ini adalah path terpendek antara 1 orang dengan orang lainnya.

Algoritma Bi-BFS itulah yang digunakan oleh LinkedIn dalam fitur Degree pada Connectionsnya untuk mencari tingkatan hubungan antara 1 orang dengan orang lainnya. Tentunya, algoritma ini hanya merupakan salah satu optimasi dari masih banyak optimasi lainnya yang dilakukan.

V. UCAPAN TERIMA KASIH

Pertama – tama, penulis memanjatkan Puji dan Syukur kepada Tuhan yang Maha Esa, karena atas limpahan rahmatnya, saya dapat menyelesaikan Makalah Strategi Algoritma yang berjudul “Aplikasi Algoritma Bidirectional Breadth First Search pada Fitur degree pada Connections dalam aplikasi Social Networking LinkedIn” ini dan seluruh rangkaian kegiatan pembelajaran Mata Kuliah IF2211 sepanjang semester ini.

Tentunya tanpa rahmatnya, saya tidak akan bisa mencapai apa yang berhasil saya capai sekarang.

Penulis juga menyampaikan terima kasih kepada Ir. Rinaldi Munir, selaku dosen IF2211 untuk K3, karena telah menyampaikan banyak sekali pengetahuan baru yang kami dapatkan pada kelas ini dengan sangat jelas dan mudah dipahami. Cara mengajar Pak Rinaldi adalah salah satu cara mengajar favorit saya dikarenakan pemilihan kata yang cenderung mudah dan penjelasan yang sederhana namun langsung pada intinya.

Penulis juga menyampaikan terima kasih kepada semua teman dan orang – orang yang berperan dalam terselesaikannya makalah ini, dan tidak lupa juga terima kasih untuk penulis yang ada di referensi karena sudah bersedia membagikan ilmunya kepada penulis sehingga makalah ini dapat terselesaikan.

LINK VIDEO YOUTUBE

Untuk video penjelasan mengenai makalah ini, dapat diakses pada link <https://youtu.be/Pm4lRbK1ao4>

REFERENSI

[1] J. W. Kimball. *Kimball's Biology Pages – Blood*. 1999.

- [2] <https://www.geeksforgeeks.org/design-data-structures-for-a-very-large-social-network-like-facebook-or-linkedin/> Diakses pada 1 Mei 2020
- [3] <https://stackoverflow.com/questions/10995699/how-do-you-use-a-bidirectional-bfs-to-find-the-shortest-path> Diakses pada 1 Mei 2020
- [4] Rinaldi Munir, 'Breadth First Search dan Depth First Search. [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 pada 1 Mei 2020
- [5] <https://about.linkedin.com/id-id> Diakses pada 2 Mei 2020

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.

Bogor, 1 Mei 2020



Fabianus Harry Setiawan
13518144