

# Modifikasi Arsitektur *Graph Search* dan Penggunaan Algoritma Prim sebagai Fitur Rekomendasi Pertemanan dalam Media Sosial

Kevin Austin Stefano 13518104<sup>1</sup>  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
<sup>1</sup>13518104@std.stei.itb.ac.id

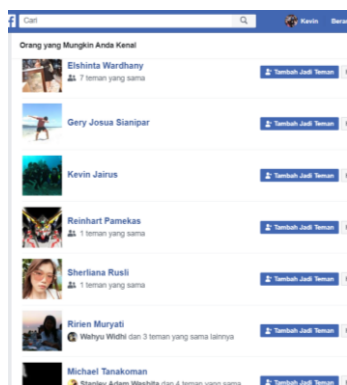
**Abstract**—Di era globalisasi ini, media sosial berkembang dengan sangat cepat. Fitur-fitur baru mulai bermunculan yang menandakan perkembangan inovasi dan algoritma tidak pernah berhenti pada suatu titik. Namun, seringkali kita tidak mengetahui bagaimana cara media sosial mengelola fitur-fitur mereka. Salah satu fitur yang populer dalam media sosial adalah rekomendasi pertemanan atau *suggestion friend*. Penulis memodelkan fitur rekomendasi pertemanan dalam bentuk arsitektur *Graph Search* dengan penggunaan Algoritma Prim sebagai bentuk implementasi graf dan pohon merentang minimum.

**Keywords**—algoritma prim, graf, media sosial, pertemanan

## I. PENDAHULUAN

Di era digitalisasi ini, sosial media menjadi sesuatu yang penting bagi setiap orang. Dengan sosial media, maka kita bisa berhubungan secara tidak langsung dengan orang-orang yang berjauhan dari kita. Namun terlebih lagi, sosial media memungkinkan kaum milenial untuk mengenal orang-orang baru yang belum pernah dikenal sebelumnya.

Salah satu dari banyak fitur istimewa yang dimiliki oleh setiap media sosial adalah fitur *suggestion friend*. Fitur ini dibuat oleh sosial media agar orang-orang yang mungkin kita kenal di dunia nyata, namun kita belum connect atau saling mengikuti di media sosial, bisa berhubungan satu sama lain. Fitur ini akan menyarankan teman-teman kita sehingga kita bisa saling berkomunikasi, berhubungan lebih erat di dalam sosial media.



Gambar 1. *Suggestion friend list* di Facebook  
(<https://www.facebook.com/friends/requests/?fcfref=nr>)

Banyak jenis algoritma yang dipakai oleh perusahaan sosial media berskala besar untuk mengolah *suggestion friend list* atau rekomendasi pertemanan. Dalam makalah ini, kita akan mencoba membuat fitur *suggestion friend* menggunakan modifikasi arsitektur *Graph Search* yang ditemukan oleh Facebook pada tahun 2013. Kita juga akan mencoba membuatnya dengan algoritma Prim yang sudah dipelajari sehingga kita bisa melihat penerapan Algoritma Prim dan penggunaannya dalam pembuatan fitur media sosial.

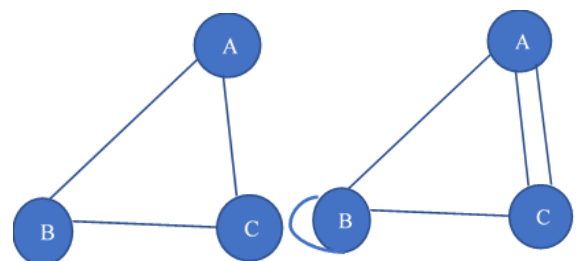
## II. LANDASAN TEORI

### 2.1. Definisi Graf

Graf adalah himpunan objek-objek yang saling terhubung satu sama lainnya. Dalam graf, kita biasanya menyebut objek-objek tersebut dengan sebutan simpul atau *vertices*. Sedangkan penghubung antara simpul yang satu dengan simpul lainnya kita sebut dengan sisi atau *edges*. Suatu sisi dapat dikatakan sisi tunggal jika hanya satu sisi yang menghubungkan sepasang simpul. Namun, sisi dikatakan sisi ganda jika ada lebih dari satu sisi yang menghubungkan sepasang simpul yang sama. Jika sepasang simpul yang terhubung oleh sisi keduanya adalah simpul yang sama, maka sisi yang menghubungkannya disebut gelang atau kalang.

### 2.2. Jenis Graf

- A. Berdasarkan keberadaan sisi ganda dan gelang  
Graf memiliki jenis yang beraneka ragam. Jenis paling umum dari graf adalah graf yang dibedakan berdasarkan keberadaan sisi ganda dan gelang dalam suatu graf.
  - a. Graf Sederhana



Gambar 2. Graf sederhana di sebelah kiri dan graf tak sederhana di sebelah kanan.

Graf sederhana atau *simple graph* adalah graf yang tidak memiliki sisi ganda dan tidak memiliki gelang.

b. Graf Tidak Sederhana

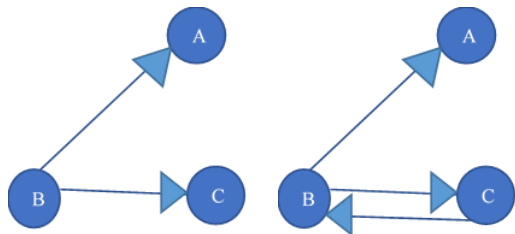
Graf tidak sederhana atau *unsimple graph* adalah kebalikan dari graf sederhana yaitu graf yang memiliki sisi ganda atau gelang. Graf tidak sederhana memiliki banyak sekali macam-macamnya, karena pada umumnya, graf tidak sederhana sering diaplikasikan dan memiliki banyak implementasinya dalam dunia nyata.

B. Berdasarkan arah dari sisinya

Graf juga dapat dibedakan berdasarkan ada tidaknya arah dari setiap sisi dari simpul-simpulnya. Adapun berdasarkan arah sisinya, graf dibedakan menjadi dua.

a. Graf Berarah

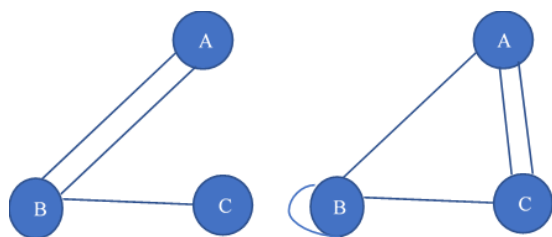
Suatu graf dikatakan berarah jika setiap sisinya memiliki orientasi arah tujuan. Graf berarah sendiri dibagi lagi menjadi dua, yaitu graf berarah biasa yaitu setiap sisinya memiliki orientasi arah, namun sisi ganda tidak diperbolehkan. Jenis kedua adalah graf ganda berarah yang mana sisinya berarah namun sisi ganda diperbolehkan.



Gambar 3. Graf berarah biasa di sebelah kiri dan graf ganda berarah di sebelah kanan

b. Graf Tidak Berarah

Suatu graf dikatakan tidak berarah jika sisinya tidak memiliki orientasi arah. Graf sederhana adalah graf yang tidak berarah. Graf ganda adalah graf tidak berarah namun sisi gelang tidak diperbolehkan. Sedangkan graf berarah yang memperbolehkan sisi ganda dan sisi gelang disebut dengan graf semu.



Gambar 4. Graf ganda di sebelah kiri dan graf semu di sebelah kanan

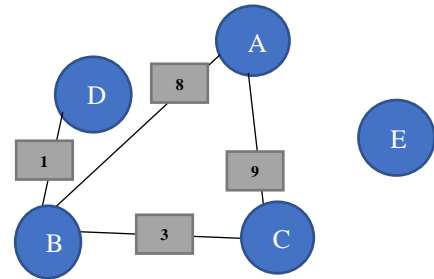
C. Berdasarkan bobot sisinya

a. Graf Tak Berbobot

Graf yang sisinya tidak memiliki bobot. Gambar-gambar sebelumnya menunjukkan graf tak berbobot.

b. Graf Berbobot

Graf berbobot atau *wighted graph* adalah graf yang di semua sisinya memiliki bobot, nilai, atau harga. Nilai atau bobot yang terkandung di setiap sisi merepresentasikan hal yang beragam di setiap grafnya. Ada graf yang bobotnya menampilkan informasi seperti jarak, ukuran, waktu, jumlah derajat, dan sebagainya.



Gambar 5. Graf berbobot

2.2 Terminologi Graf

Adapun beberapa terminologi graf adalah sebagai berikut.

A. Bertetangga

Simpul dalam suatu graf G dikatakan bertetangga jika ada sisi yang menghubungkan kedua simpul tersebut. Pada gambar 5, simpul D dan simpul E bertetangga karena dihubungkan oleh sisi (D,E).

B. Bersisian

Suatu sisi bersisian dengan suatu simpul jika sisi tersebut terhubung langsung atau menghubungkan simpul bersangkutan. Gambar 5 menunjukkan simpul B bersisian dengan sisi (B,A) dan (B,D).

C. Derajat

Derajat adalah banyaknya sisi yang terhubung atau bersisian dengan sisi tersebut. Gambar 5 menunjukkan derajat simpul B adalah 2 dan derajat simpul E adalah 0 atau tidak ada.

D. Simpul terpencil

Simpul terpencil adalah simpul dengan derajat nol. Jika kita lihat dari gambar 5, maka simpul E adalah simpul terpencil.

E. Lintasan

Lintasan atau *Path* adalah barisan berselang seling antara simpul dengan sisi-sisi dari graf. Tinjau graf dari gambar 5 maka lintasan D, B, A adalah lintasan dengan barisan sisi (D,B) dan (B,A) yang mana panjang lintasannya adalah 2.

F. Siklus

Siklus atau sirkuit adalah lintasan yang dimulai dari simpul yang sama dengan simpul yang mengakhirinya. Contohnya adalah lintasan B,A,C,B berdasarkan gambar 5.

### G. Terhubung

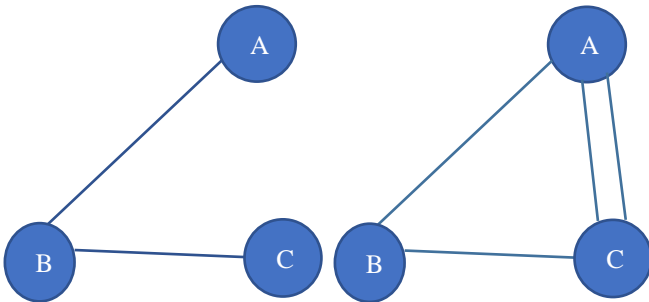
Sepasang simpul bisa kita katakan terhubung atau *connected* jika ada lintasan yang bisa dijalani dari simpul yang satu ke simpul satunya. Jika setiap simpul terhubung dalam suatu graf, maka graf tersebut disebut graf terhubung. Sebaliknya, jika ada satu saja simpul yang tidak terhubung dalam suatu graf maka graf tersebut disebut dengan graf tak terhubung. Gambar 5 bisa kita katakan sebagai contoh dari graf tak terhubung.

### H. Graf Kosong

Graf kosong adalah graf yang tiap simpulnya tidak memiliki sisi yang bersisian dengannya. Atau dengan kata lain himpunan sisinya adalah *null* atau kosong.

### I. Upagraf

Suatu graf ( $G_1$ ) disebut sebagai upagraf dari suatu graf ( $G$ ) jika setiap simpul dan sisi yang ada di graf  $G_1$  merupakan himpunan bagian dari simpul dan sisi yang terdapat di graf  $G$ .



Gambar 6. Graf di sebelah kiri adalah upagraf dari graf di sebelah kanannya

### J. Upagraf Rentang

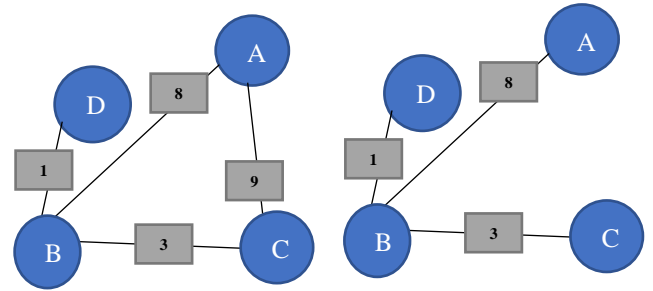
Suatu graf ( $G_1$ ) disebut sebagai upagraf rentang dari suatu graf ( $G$ ) jika setiap simpul dalam graf  $G_1$  terdapat semuanya dalam graf  $G$ , namun himpunan sisi dari graf  $G_1$  merupakan himpunan bagian dari graf  $G$ . Graf pada gambar 6 juga menunjukkan bahwa graf di sebelah kiri merupakan upagraf rentang dari graf di sebelah kanannya.

## 2.3 Pohon

Pohon adalah graf yang memenuhi tiga syarat berikut. Syarat pertama adalah graf tidak berarah atau tidak memiliki orientasi tujuan pada setiap sisinya. Syarat kedua adalah pohon merupakan graf terhubung atau *connected*. Dan yang terakhir adalah graf yang tidak mengandung sirkuit. Atau dengan kata lain tidak ada simpul yang memiliki lintasan yang kembali kepada dirinya. Sekumpulan pohon yang saling lepas atau tidak terhubung bisa kita sebut dengan hutan.

Dalam pohon kita mengenal istilah pohon merentang atau *spanning tree*. Pohon merentang adalah upagraf merentang dari suatu pohon. Misalkan kita memiliki suatu graf berbobot, terhubung, dan tak berarah. Maka pohon merentang yang bisa dibentuk dari graf tersebut dengan bobot atau nilai yang paling

minimum bisa kita katakan sebagai pohon merentang minimum atau *minimum spanning tree*.



Gambar 7. Graf di sebelah kanan adalah pohon merentang minimum dari pohon di sebelah kirinya

## 2.4 Algoritma Prim

Algoritma Prim adalah algoritma yang digunakan untuk mencari pohon merentang minimum dari suatu graf yang memiliki bobot. Jika kita memiliki graf dengan simpul yang memiliki banyak sisi dan terhubung dengan berbagai macam bobotnya, maka algoritma Prim akan meminimalisikannya dengan memilih bobot dari sisi yang paling minimum, kemudian membentuknya menjadi pohon merentang minimum.

Adapun langkah-langkah untuk membentuk algoritma Prim adalah sebagai berikut. Yang pertama kita ambil sisi graf dengan bobot yang paling minimum, kemudian kita masukkan ke dalam  $T$  sebagai inisialisasi awal. Selanjutnya pilih sisi yang berbobot minimum dan bersisian dengan simpul  $T$  tapi tidak membentuk sirkuit di  $T$ . Jika validasi berhasil maka masukkan sisi tersebut ke dalam  $T$ . Ulangi langkah kedua atau langkah setelah inisialisasi sebanyak  $n-2$  kali hingga kita mendapatkan pohon merentang minimum dari graf tersebut.

## 2.5 Arsitektur Graph Search

Arsitektur *Graph Search* pada awalnya diperkenalkan oleh Facebook pada bulan Maret tahun 2013. Arsitektur ini dirancang sebagai solusi untuk menjawab pertanyaan dari user Facebook dalam kolom pencarian. Mark Zuckerberg mengumumkan bahwa algoritma *Graph Search* ini akan mempertemukan antara informasi yang ingin diperoleh oleh *user* dalam graf jaringan pertemanan pengguna. Pengembangan awal algoritma ini dimulai oleh Lars Rasmussen dan Tom Stocky. Layanan Facebook *Graph Search* yang menggunakan algoritma ini pada awalnya diluncurkan secara terbatas kepada beberapa penggunanya.

Layanan yang menggunakan arsitektur *Graph Search* bertumbuh sangat pesat dan multifungsi karena bisa melakukan banyak hal. Jika kita ingin mencari suatu keterhubungan dengan suatu individu, kita bisa melakukannya dengan arsitektur ini. Misalnya kita ingin mengetahui “teman dari orang A yang tinggal di tempat B” dalam Facebook Search. Maka Facebook akan menggunakan *lexer* untuk memecahkan *query* atau permintaan dari *user* menjadi *keypoint* atau kata-kata kunci. Maka kata kunci yang didapat adalah  $A \rightarrow$  teman A dan teman  $A \rightarrow$  tinggal di tempat B. Setelah itu akan diolah dengan metode *parsing* dan dilakukan pencarian dalam arsitektur *Graph Search*.

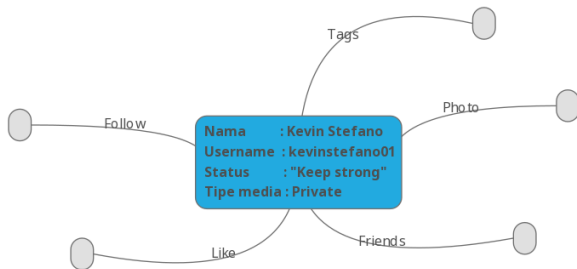
Karena sifatnya yang multifungsi dan dapat memperoleh berbagai informasi keterhubungan individu dengan skala besar, maka Facebook menutup layanan Facebook *Graph Search* dengan alasan keamanan pribadi pengguna sehingga Facebook tidak menggunakan lagi layanan Facebook *Graph Search* berbasis pencarian dalam arsitektur graf. Namun, sistem arsitektur ini sangatlah bagus untuk dipelajari sebagai dasar pengembangan skema fitur *suggestion* dan *search* dalam sosial media dengan penambahan filter untuk menyaring data pribadi pengguna agar tidak semua dapat ter-*expose* kepada publik.

### III. METODOLOGI

#### 3.1. Implementasi Arsitektur *Graph Search*

Arsitektur *Graph Search* dibangun dengan arsitektur berbentuk graf yang menggambarkan keterhubungan entitas dalam suatu sosial media. Ciri-Ciri dasar suatu graf adalah graf memiliki simpul dan sisi yang menghubungkan simpul-simpul yang ada. Pada arsitektur *Graph Search*, setiap simpul menggambarkan entitas yang terdiri dari tipe bentuk. Entitas disini menggambarkan suatu individu dalam suatu sosial media dan informasi mengenai dirinya.

Misalkan suatu individu A dalam suatu sosial media, memiliki informasi diri mengenai namanya, username atau ID dalam sosial media, status dalam sosial media, tipe laman di sosial media, tanggal ulang tahun, dan informasi-informasi pribadi lain yang menggambarkan simpul dalam arsitektur graf ini. Sedangkan sisi dari simpul menggambarkan keterhubungan individu dengan hal-hal yang dilakukannya dalam sosial media. Sisi dari simpul suatu individu bisa menggambarkan kesukaan individu, foto-foto suatu individu, teman-teman-teman yang dimiliki, dan masih banyak lagi. Sehingga kita bisa menggambarkan suatu graf dengan simpul besar berisikan data pribadi kita dan simpul-simpul kecil yang terhubung berisikan hal-hal yang menggambarkan diri kita di media sosial.



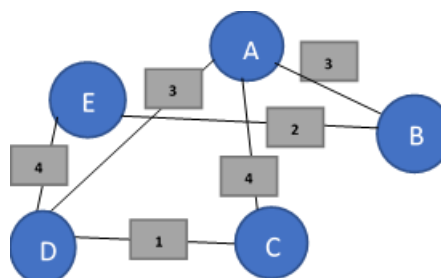
Gambar 8. Graf dengan model arsitektur Facebook *Graph Search*

#### 3.2. Pembuatan Graf dengan Modifikasi Arsitektur *Graph Search* berbasis Algoritma Prim

Arsitektur *Graph Search* memiliki cakupan yang sangat luas dalam menjelaskan mengenai keterhubungan. Kita akan memodifikasi arsitektur *Graph Search* menjadi graf umum yang dapat merepresentasikan keterhubungan antara entitas dalam suatu sosial media.

Langkah pertama yang dilakukan adalah dengan membuat graf dengan simpul yang merepresentasikan individu-individu dalam sosial media. Setiap simpul merepresentasikan setiap

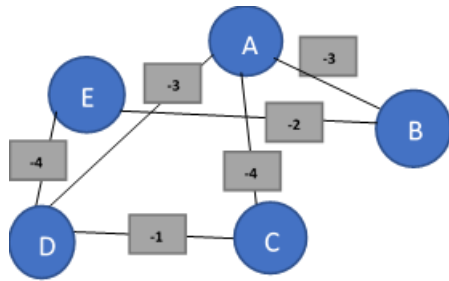
individu-individu yang ada dalam media sosial. Selanjutnya kita membuat setiap sisinya dengan keterhubungan seperti yang dijelaskan oleh arsitektur *Graph Search*. Namun kita menghitung keterhubungan tersebut menjadi bobot yang dituliskan dalam sisi. Misalkan, kita memiliki parameter keterhubungan sebanyak 4 buah. Parameter pertama adalah antarindividu memiliki kesukaan yang sama akan suatu hal. Parameter kedua menggambarkan hobi yang sama antarindividu. Parameter ketiga adalah foto yang disukai antarindividu. Parameter yang terakhir adalah apakah kedua individu saling berteman di sosial media. Jika suatu individu dan individu lainnya terhubung satu sama lain dengan 4 parameter di atas, maka kita menuliskan bobot sisi antarsimpul yang bersangkutan dengan angka 4. Namun, jika hanya 3 atau 2 parameter yang terpenuhi oleh dua buah individu, maka kita menuliskan bobot antarsimpul menjadi 3 atau 2.



Gambar 9. Graf model langkah pertama

Untuk menentukan hubungan kedekatan antarindividu maka kita akan melihat dari yang bobotnya terbesar. Dengan kata lain kita bisa mengatakan bahwa simpul atau individu yang memiliki bobot besar atau keterhubungan terbesar menandakan tingginya kesamaan yang dimiliki antarindividu dalam sosial media. Oleh karena itu, kita akan mencari *the longest path* atau lintasan dengan bobot tertinggi namun dengan lintasan terpendek dari graf yang sudah kita bentuk yang menunjukkan simpul-simpul yang terhubung dengan sangat kuat dalam sosial media.

Namun, algoritma-algoritma yang ada pada dasarnya ditujukan untuk mencari *the shortest path* atau lintasan dengan bobot terkecil. Sedangkan, kita memerlukan algoritma untuk mendapatkan lintasan dengan bobot tertinggi namun diraih dengan lintasan terpendek. Oleh karena itu, kita akan melakukan manipulasi kepada graf agar graf yang kita miliki menunjukkan bobot minimum namun esensi dari bobot tersebut tetap menunjukkan nilai maksimum. Langkah kedua yang harus kita lakukan terhadap graf kita adalah dengan dengan mentransformasi bobot sisinya (E) menjadi bilangan negatifnya (-E). Hal ini menyebabkan kita mencari *the longest path* dari graf yang ada dengan metode *the shortest path* karena bobot-bobot dari setiap sisi sudah dinegasikan. Namun hal ini tidak mempengaruhi esensi dari bobot yang ada. Jika pada awalnya bobot yang tertinggi menunjukkan bahwa keterhubungan yang kuat antarsimpul. Maka sekarang, bobot yang paling rendahlah yang menunjukkan simpul dengan keterhubungan yang paling kuat karena nilai bobotnya sudah kita negasikan.

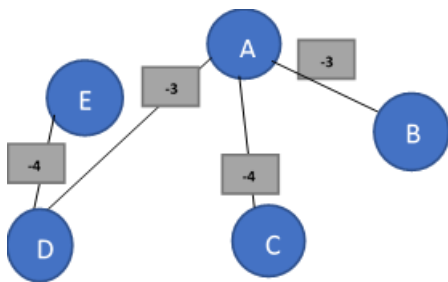


Gambar 10. Graf model langkah kedua

Jika sudah dibentuk menjadi bilangan negatifnya, maka kita bisa langsung mencari *the shortest path* menggunakan algoritma-algoritma yang ada. Karena mencari *the longest simple path* pada dasarnya bisa dilakukan dengan mencari *the shortest path* dengan nilai bobot sisi yang sudah dinegasikan.

Langkah ketiga adalah kita mencari *the shortest path* dengan pohon merentang minimum. Banyak algoritma yang bisa digunakan untuk mendapatkan pohon merentang minimum, seperti algoritma Prim, algoritma Kruskal, dan masih banyak. Namun disini, kita akan mencoba mengimplementasikan penerapan algoritma Prim untuk mencari pohon merentang dengan bobot minimum berdasarkan graf yang kita miliki.

Adapun langkah-langkah untuk membentuk pohon merentang minimum dengan algoritma Prim pertama kita mengambil sisi graf dengan bobot yang paling minimum, kemudian kita masukkan ke dalam G sebagai inisialisasi awal. Selanjutnya pilih sisi yang berbobot minimum dan bersisian dengan simpul G tapi tidak membentuk sirkuit di G. Jika validasi berhasil maka masukkan sisi tersebut ke dalam graf G. Ulangi langkah kedua atau langkah setelah inisialisasi sebanyak  $n-2$  kali hingga kita mendapatkan pohon merentang minimum dari graf tersebut.



Gambar 11. Pohon merentang minimum hasil model langkah ketiga

Setelah kita sudah berhasil mendapatkan pohon merentang minimumnya, maka kita bisa menarik kesimpulan akhir. Berdasarkan gambar 11, kesimpulan yang bisa didapat bahwa individu E memiliki kedekatan yang sangat kuat di media sosial dengan individu D, begitu pula antara individu A dan individu C, individu D dan individu A juga memiliki kedekatan di media sosial, begitu pula antara individu A dan individu B. Karena kedekatan yang kuat inilah maka perusahaan media sosial, umumnya, menyarankan pertemanan antara individu-individu yang terhubung kuat namun belum berteman atau

*connected* dalam sosial media. Perusahaan media sosial biasanya menyarankan “teman yang mungkin anda kenal” atau *suggestion friend list* berdasarkan parameter yang dimiliki masing-masing perusahaan.

#### IV. STUDI KASUS

Misalkan kita membuat indikator kedekatan satu individu dengan individu lainnya dalam sosial media dengan bentuk sebagai berikut.

Tabel 1. Tabel indikator keterhubungan antarindividu dalam sosial media

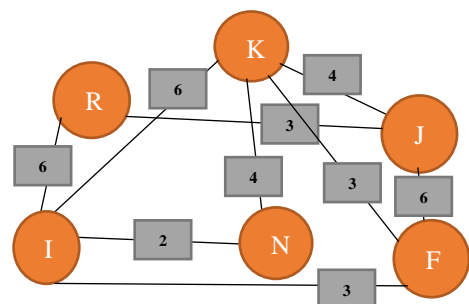
No	Indikator Keterhubungan
1.	Saling mengikuti ( <i>follow</i> ) orang yang sama di sosial media
2.	Saling menyukai foto satu sama lain
3.	Saling memberikan komentar di <i>post</i> satu sama lain
4.	Saling membagikan ( <i>share</i> ) foto, <i>post</i> , satu dengan lainnya
5.	Berkuliah di tempat yang sama
6.	Memiliki hobi / kegemaran yang sama

Kemudian kita mengambil sampel aktivitas individu dalam media sosial, misalnya Instagram. Aktivitas-aktivitas yang dilakukan oleh individu dalam sosial media dan masuk dalam parameter indikator keterhubungan kita letakkan dalam nilai aspek keterhubungan yang terpenuhi.

Tabel 2. Tabel contoh aktivitas individu dalam media sosial

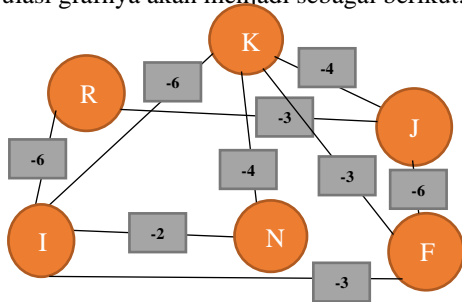
No.	Nama	Aspek Keterhubungan yang Terpenuhi
1.	Kevin, Indra	1,2,3,4,5,6
2.	Indra, Ryan	1,2,3,4,5,6
3.	Kevin, Ryan	1,2,5,6
4.	Ryan, Jun	1,5,6
5.	Nadya, Indra	1,2
6.	Kevin, Nadya	1,2,3,4
7.	Jun, Fang	1,2,3,4,5,6
8.	Kevin, Fang	1,5,6
9.	Indra, Fang	1,5,6

Berdasarkan data-data di atas, maka kita membentuknya menjadi suatu graf berbobot. Pertama kita membuat simpul dari nama-nama individu bersangkutan. Misalkan kita menggunakan inisial dari nama, seperti K untuk Kevin, I untuk Indra, R untuk Ryan, J untuk Jun, F untuk Fang, dan N untuk Nadya. Kemudian setiap simpul kita hubungkan dengan sisi yang memiliki bobot dengan jumlah dari aspek keterhubungan yang terpenuhi. Maka, seperti yang sudah dijelaskan dalam pembuatan graf dengan modifikasi arsitektur *Graph Search* model langkah satu, kita akan mendapatkan graf berbobot dengan bentuk sebagai berikut.



Gambar 12. Graf studi kasus hasil model langkah pertama

Kemudian kita mengubah setiap bobot menjadi nilai negatifnya seperti sudah dijelaskan dalam pembuatan graf dengan modifikasi arsitektur *Graph Search* langkah kedua. Hasil manipulasi grafnya akan menjadi sebagai berikut.



Gambar 13. Graf studi kasus hasil model langkah kedua

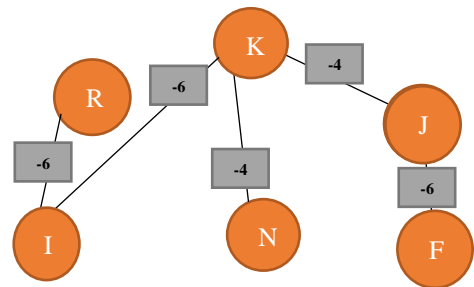
Dengan menggunakan algoritma Prim, maka kita akan mencari pohon merentang minimum yang bisa dibentuk dari graf di atas. Berikut ini adalah langkah-langkah pembentukan pohon merentang minimum dengan menggunakan algoritma Prim.

Tabel 2. Tabel pengolahan algoritma Prim dalam mencari pohon merentang minimum model langkah ketiga

No	Sisi	Bobot	Pohon Rentang
1.	(I,R)	-6	
2.	(I,K)	-6	
3.	(K,N)	-4	

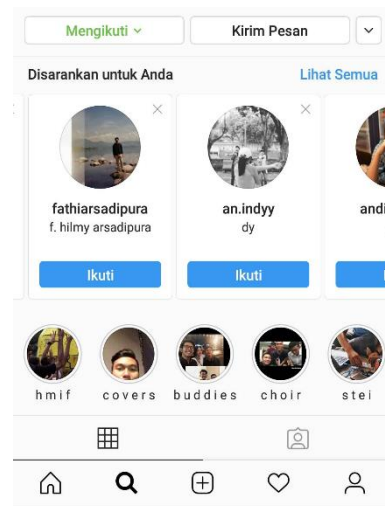
4.	(K,J)	-4	
5.	(J,F)	-6	

Maka berdasarkan algoritma Prim, kita telah berhasil mendapatkan pohon merentang minimum. Berikut adalah pohon merentang minimum yang dihasilkan dari algoritma Prim.



Gambar 14. Pohon merentang minimum hasil model langkah ketiga

Berdasarkan pohon merentang minimum di atas, kita bisa menarik kesimpulan bahwa Ryan memiliki kedekatan yang kuat dengan Indra, Kevin dengan Nadya, Kevin dengan Jun, dan Jun dengan Fang. Data-data ini kemudian dapat digunakan oleh perusahaan sosial media untuk memberikan saran pertemanan kepada individu-individu yang bersangkutan jika belum terhubung atau mengikuti (*follow*) satu sama lainnya di media sosial.



Gambar 15. Fitur *suggestion friend list* di Instagram (<https://www.instagram.com/indrafngroho/>)

Hal yang sama tentunya diimplementasikan oleh sosial media berskala besar untuk memberikan *suggestion friend* atau seperti *friend you may know*. Fitur ini akan menyarankan suatu teman yang memiliki banyak kesamaan kepada suatu individu agar bisa *connect* atau berhubungan satu sama lainnya. Aspek atau indikator yang menentukan keterhubungan antarindividu di setiap media sosial biasanya berbeda. Dengan menghitung jumlah atau kuantitas dari keterhubungannya, kita bisa melihat orang-orang yang sebenarnya dekat dengan kita, namun kita belum *connect* atau mengikuti satu sama lain. Modifikasi penerapan arsitektur *Graph Search* berbasis Algoritma Prim bisa menjadi satu dari banyak cara yang dipakai oleh media sosial untuk menyarankan pertemanan kepada pengguna-penggunanya.

## V. KESIMPULAN

Modifikasi Arsitektur *Graph Search* berbasis Algoritma Prim dapat digunakan sebagai alternatif untuk membangun fitur rekomendasi pertemanan atau *suggestion friend list* di sosial media berskala besar. Parameter atau indikator keterhubungan antarindividu pada setiap sosial media biasanya berbeda. Namun, perusahaan sosial media biasanya memiliki indikator keterhubungan antarentitas yang sangat baik sehingga individu yang disarankan kepada kita agar kita berteman di sosial media adalah orang yang kita kenal dan tepat sasaran. Oleh karena itu, sistem rekomendasi pertemanan sosial media memiliki algoritma-algoritma tambahan yang lebih kompleks dalam menyarankan pertemanan.

## VI. UCAPAN TERIMA KASIH

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas berkat dan kasih-Nya, penulis dapat menyelesaikan makalah yang berjudul "Modifikasi Arsitektur *Graph Search* dan Penggunaan Algoritma Prim sebagai Fitur Rekomendasi Pertemanan Media Sosial".

Penulis juga berterima kasih kepada dosen penulis, Dra. Harlili S., M.Sc., Dr. Ir. Rinaldi Munir, MT., dan Drs. Judi Santoso, M.Sc., yang telah membimbing penulis selama perkuliahan matematika diskrit sehingga penulis dapat menyelesaikan makalah ini.

Penulis juga berterima kasih kepada kedua orang tua penulis yang selalu memberika dukungan dan motivasi kepada penulis sehingga perkuliahan ini dapat berjalan dengan lancar. Selain itu, penulis juga berterima kasih kepada teman-teman penulis yang mendukung penulis selama perkuliahan matematika diskrit ini.

## REFERENSI

- [1] Munir, Rinaldi. 2004. Matematika Diskrit. Bandung: Informatika.
- [2] "Facebook Announces Its Third Pillar "Graph Search" That Gives You Answers, Not Links Like Google". TechCrunch. 15 January 2013. Retrieved 29 November 2019.
- [3] Tsukayama, Hayley. "Facebook introduces social search feature". The Washington Post. 15 January 2013. Retrieved 30 November 2019.
- [4] <https://www.facebook.com/notes/facebook-engineering/under-the-hood-building-out-the-infrastructure-for-graph-search/10151347573598920>. Retrieved 29 November 2019.

## 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, 2 Desember 2019



Kevin Austin Stefano / 13518104