

Aplikasi Algoritma Greedy pada Optimasi Pemilihan Anggota di Suatu Kepanitiaan atau Organisasi

Muhammad Akmal 13517028
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
akmalmuhammad51@gmail.com

Abstract—Di era yang serba kolaborasi ini, banyak sekali masalah yang dapat diselesaikan dengan cara bekerjasama baik yang cakupannya kecil dari 1 – 5 orang sampai yang cakupannya besar, sampai 150 orang bahkan lebih. Diperlukan suatu cara agar kolaborasi ini yang dapat ditampung sebagai kepanitiaan atau organisasi dapat berjalan dengan sangat baik di setiap tahap perkembangannya. Salah satu cara yang dapat digunakan adalah dengan menggunakan Algoritma Greedy untuk mengoptimasi pemilihan anggota di suatu kepanitiaan atau organisasi tersebut.

Keywords—Algoritma Greedy, Kepanitiaan, Organisasi

I. PENDAHULUAN

Kolaborasi saat ini telah menjadi hal penting dalam menyelesaikan suatu masalah. Kolaborasi juga dapat menjadikan sesuatu yang tadinya hanya mencakup pada hal yang kecil menjadi hal yang sangat besar. Dapat kita lihat pada 10 April yang lalu bahwa telah didapatkannya sebuah foto dari salah satu *Black Hole* yang tentunya mengguncang dunia. Ya, adalah kolaborasi yang menjadikan dapat terjadinya penemuan seperti itu. Mereka bekerjasama antar ilmuwan di banyak universitas di banyak negara.

Dengan berkembangnya permasalahan di dunia kita ini, akan sangat bermanfaat jika kolaborasi diterapkan pada banyak sudut kehidupan mulai dari mengerjakan tugas, menciptakan *startup*, sampai menemukan penemuan yang sangat hebat. Namun, pada nyatanya kolaborasi yang ideal tidak dapat dicapai jika kolaborasi tersebut tidak dipersiapkan matang-matang.

Dengan berkembangnya ilmu pengetahuan, telah banyak pula berbagai cara untuk bisa menjadikan kolaborasi menjadi suatu hal yang tidak ditakuti karena kekhawatiran akan kesuksesannya, namun menjadi suatu hal yang memudahkan kita dalam menyelesaikan segala permasalahan yang ingin kita temukan solusi optimalnya.

Salah satu strategi dalam menentukan kolaborasi yang ideal, dalam hal ini penentuan anggota pada suatu kepanitiaan atau organisasi, algoritma Greedy merupakan suatu hal yang menarik untuk diimplementasikan dikarenakan kemampuannya untuk bisa menjadi salah satu cara untuk mengestimasi nilai optimal dalam suatu permasalahan yang dalam hal ini permasalahan tersebut adalah untuk penentuan anggota.

II. TEORI DASAR ALGORITMA GREEDY

A. Definisi Algoritma Greedy

Dalam kehidupan sehari-hari, banyak terdapat persoalan yang menuntut pencarian solusi optimum. Persoalan tersebut dinamakan persoalan optimasi (*optimization problems*). Persoalan optimasi adalah persoalan yang tidak hanya mencari sekedar solusi, tetapi mencari solusi terbaik (*best*). Solusi terbaik adalah solusi yang bernilai minimum atau maksimum dari sekumpulan alternatif solusi yang mungkin. Contohnya menentukan lintasan terpendek dalam sebuah graf, menentukan total keuntungan maksimum dari pemilihan beberapa objek, dan sebagainya. Pada persoalan optimasi, kita diberi sekumpulan **kendala** (*constraint*) dan fungsi optimasi. Solusi yang memenuhi semua kendala adalah **solusi layak** (*feasible solution*). Solusi layak yang mengoptimalkan fungsi optimasi disebut **solusi optimum**.

Algoritma Greedy membentuk solusi langkah per langkah (*step by step*). Terdapat banyak pilihan yang perlu dieksplorasi pada setiap langkah solusi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Keputusan yang telah diambil pada suatu langkah tidak dapat diubah lagi pada langkah selanjutnya. Sebagai contoh, jika kita menggunakan algoritma Greedy untuk menempatkan komponen di atas papan sirkuit (*circuit board*), sekali sebuah komponen telah ditetapkan posisinya, komponen tersebut tidak dapat dipindahkan lagi.

Pendekatan yang digunakan di dalam algoritma Greedy adalah membuat pilihan yang “tampaknya” memberikan perolehan terbaik, yaitu dengan membuat pilihan **optimum lokal** (*local optimum*) pada setiap langkah dengan harapan bahwa sisanya mengarah ke solusi **optimum global** (*global optimum*).

Berangkat dari pendekatan yang disebutkan di atas, kita dapat mendefinisikan algoritma Greedy sebagai berikut:

Algoritma Greedy adalah algoritma yang memecahkan masalah langkah per langkah, pada setiap langkah:

1. Mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan (prinsip “*take what you can get now!*”)
2. Berharap bahwa dengan memilih optimum lokal (*local optimum*) pada setiap langkah akan berakhir dengan optimum global (*global optimum*)

Pada setiap langkah di dalam algoritma Greedy kita baru memperoleh optimum lokal (*local optimum*) menjadi optimum global (*global optimum*). Algoritma Greedy mengasumsikan bahwa optimum lokal merupakan bagian dari optimum global.

B. Skema Umum Algoritma Greedy

Persoalan optimasi dalam konteks algoritma Greedy disusun oleh elemen-elemen sebagai berikut:

1. Himpunan kandidat, C .
Himpunan ini berisi elemen-elemen pembentuk solusi. Contohnya adalah himpunan koin, himpunan *job* yang akan dikerjakan, himpunan simpul di dalam graf, dan lain-lain. Pada setiap langkah, satu buah kandidat diambil dari himpunannya.
2. Himpunan solusi, S .
Himpunan ini berisi kandidat-kandidat yang terpilih sebagai solusi persoalan. Dengan kata lain, himpunan solusi adalah himpunan bagian dari himpunan kandidat.
3. Fungsi seleksi
Dinyatakan dengan predikat SELEKSI, yaitu fungsi yang pada setiap langkah memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya. Biasanya setiap kandidat, x , di-assign sebuah nilai numerik, dan fungsi seleksi memilih x yang mempunyai nilai terbesar atau memilih x yang mempunyai nilai tekecil.
4. Fungsi kelayakan (*feasible*)
Dinyatakan dengan predikat LAYAK, yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala (*constraints*) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.
5. Fungsi obyektif
Fungsi yang memaksimumkan atau meminimumkan nilai solusi (misalnya panjang lintasan, keuntungan, dan lain-lain).

Dengan kata lain, persoalan optimasi yang diselesaikan dengan algoritma Greedy melibatkan pencarian sebuah himpunan bagian S , dari himpunan kandidat C ; yang dalam hal ini, S harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan S dioptimisasi oleh fungsi objektif.

Kata "optimisasi" dapat berarti **minimisasi** atau **maksimasi**, bergantung pada persoalan yang dipecahkan.

Semua algoritma Greedy mempunyai skema umum yang sama. Secara umum, skema algoritma Greedy dapat kita rumuskan sebagai berikut:

1. Inisialisasi S dengan kosong,
2. Pilih sebuah kandidat (dengan fungsi SELEKSI) dari C ,
3. Kurangi C dengan kandidat yang sudah dipilih dari langkah 2 di atas,
4. Periksa apakah kandidat yang dipilih tersebut bersama-sama dengan himpunan solusi membentuk solusi yang layak atau *feasible* (dilakukan oleh fungsi LAYAK). Jika

YA, masukkan kandidat tersebut ke dalam himpunan solusi; Jika TIDAK, buang kandidat tersebut dan tidak perlu dipertimbangkan lagi.

5. Periksa apakah himpunan solusi sudah memberikan solusi yang lengkap (dengan menggunakan fungsi SOLUSI). Jika YA, berhenti (selesai); Jika TIDAK, ulangi lagi dari langkah 2.

C. Deskripsi Lanjut Algoritma Greedy

Pada sebagian masalah, algoritma Greedy tidak selalu berhasil memberikan solusi yang benar-benar optimum. Meskipun demikian, jika jawaban terbaik mutlak (benar-benar optimum) tidak diperlukan, maka algoritma Greedy sering berguna sebagai solusi yang menghampiri (*approximation*) optimum, daripada menggunakan algoritma yang lebih kompleks untuk menghasilkan solusi yang eksak.

Bila algoritma Greedy berhasil menemukan solusi yang eksak, maka algoritma tersebut harus dapat dibuktikan secara matematis menghasilkan solusi optimum. Jika pembuktian matematis dapat ditunjukkan, maka secara tipikal algoritma Greedy tersebut menjadi metode pilihan untuk kelas masalah yang diberikan.

Jika algoritma Greedy tidak berhasil menemukan solusi optimum untuk suatu masalah, maka sebagai alternatifnya, kita dapat menggunakan metode *exhaustive search* terhadap semua kemungkinan solusi untuk menemukan solusi optimum. Algoritma Greedy biasanya lebih cepat dibandingkan *exhaustive search* karena ia tidak mempertimbangkan seluruh alternatif solusi.

III. IMPLEMENTASI ALGORITMA GREEDY PADA OPTIMASI PENENTUAN ANGGOTA KEPANITIAAN ATAU ORGANISASI

A. Kondisi Awal

Pada permasalahan ini, penulis akan memberikan kondisi awal agar pembaca bisa melihat permasalahan ini sebagai permasalahan yang khusus, bukan umum. Kondisi tersebut antara lain:

1. Anda berperan sebagai pengguna algoritma Greedy yang dalam hal ini Anda sudah menjadi anggota suatu kepanitiaan atau organisasi dan memegang peranan penting seperti Ketua Umum, Sekretaris Jenderal, Ketua Sektor, Ketua Bidang, maupun Ketua Divisi.
2. Hal selanjutnya yang akan Anda pertimbangkan adalah hanya pada *scope* pemilihan anggota Anda, misalnya jika Anda adalah seorang ketua divisi, maka Anda hanya akan fokus pada penentuan anggota divisi Anda. Dengan kata lain, pada permasalahan yang akan penulis bahas, penentuan ini bersifat individual dari Anda sendiri tanpa ada campur tangan dari pihak di luar Anda.
3. Anda ingin merekrut sesedikit mungkin anggota dengan maksimal n anggota.
4. Anda sudah mengerti pekerjaan apa yang akan Anda dan anggota Anda lakukan pada kepanitiaan atau organisasi tersebut.
5. Faktor-faktor yang akan menjadi bahan pertimbangan akan dikuantifikasi berdasarkan subjektivitas Anda untuk mempermudah implementasi algoritma Greedy. Faktor-faktor tersebut antara lain:
 - Kesibukan

Anda pasti menginginkan anggota yang dapat memberikan kontribusinya semaksimal mungkin, salah satu faktor terbesarnya ada pada kesibukan mereka yang berujung pada prioritas mereka masing-masing terhadap kepanitiaan atau organisasi tersebut. Dikarenakan Anda sudah menentukan linimasa (*timeline*) pekerjaan Anda, maka kuantifikasi untuk faktor ini adalah sebagai berikut:

- Setiap ada satu kesibukan lain dari calon anggota Anda yang Anda rasa dapat menurunkan kinerja mereka kurangi poin mereka sebanyak 2 poin
- Setiap ada kesibukan mereka yang Anda rasa dapat menunjang kinerja mereka menjadi lebih baik lagi, tambahkan poin mereka sebanyak 3 poin.
- Pengalaman
Tidak dapat kita pungkiri, mereka yang sudah sering akan latihan di suatu hal akan lebih mahir dalam melakukannya lagi. Maka, pengalaman merupakan suatu faktor yang penting untuk penentuan anggota dalam suatu kepanitiaan atau organisasi. Kuantifikasi untuk faktor ini adalah sebagai berikut:
 - Setiap ada satu pengalaman bekerja yang Anda rasa dapat meningkatkan produktivitas mereka dalam bekerja bersama Anda, tambahkan poin mereka sebanyak 4 poin.
- Karakter
Kepribadian seseorang akan cukup berdampak dalam suatu kolaborasi. Maka dari itu, diperlukan strategi khusus agar Anda bisa melakukan kolaborasi secara baik berdasarkan karakter masing-masing anggota Anda. Maka, faktor ini juga cukup penting dalam penentuan anggota. Kuantifikasi untuk faktor ini adalah sebagai berikut:
 - Setiap ada satu karakter/sifat yang menurut Anda dapat membuat kolaborasi Anda bersama mereka lebih baik, tambahkan poin mereka sebanyak 5 poin.
 - Setiap ada satu karakter/sifat yang menurut Anda dapat membuat kolaborasi Anda bersama mereka tidak lebih baik, kurangi poin mereka sebanyak 4 poin

B. Skema Umum

Berdasarkan penjelasan skema umum algoritma Greedy serta kondisi awal di atas, kita dapat menentukan skema umum bagi permasalahan optimasi penentuan anggota kepanitiaan atau organisasi.

1. Himpunan kandidat, C

Himpunan kandidat dibangun oleh bermacam-macam orang yang mengajukan diri sebagai anggota Anda

yang akan Anda tentukan siapa saja yang akan menjadi anggota Anda.

2. Himpunan solusi, S

Himpunan ini dibangun oleh orang-orang yang akan Anda percayai untuk menjadi Anggota Anda yang memenuhi bahwa anggota tidak lebih dari n anggota. Pada permasalahan ini, Anda akan mendapatkan sebanyak 7 buah himpunan solusi yang keputusannya akan dikembalikan kembali kepada Anda. 7 buah himpunan solusi itu antara lain himpunan yang optimal pada faktor:

- a. Kesibukan saja
- b. Pengalaman saja
- c. Karakter saja
- d. Kesibukan dan Pengalaman
- e. Pengalaman dan Karakter
- f. Kesibukan dan Karakter
- g. Kesibukan, Pengalaman, dan Karakter

3. Fungsi seleksi

Untuk setiap himpunan solusi yang telah disebutkan sebelumnya yang berjumlah 7, ambil anggota yang memiliki jumlah dari faktor-faktor yang ingin dioptimasi terbesar.

4. Fungsi kelayakan (*feasible*)

Memeriksa apakah banyak anggota yang telah diambil memenuhi nilai minimal a , b dan c serta jumlah anggota tidak melebihi n .

5. Fungsi obyektif

Meminimumkan jumlah anggota pada setiap himpunan solusi yang akan dibentuk.

C. Persiapan Sebelum Implementasi Algoritma Greedy

Terdapat beberapa kondisi yang harus Anda capai dalam mengimplementasikan Algoritma Greedy berdasarkan Skema Umum yang telah dijelaskan sebelumnya sebagai berikut:

1. Anda telah melakukan wawancara atau pengumpulan data terkait ketiga faktor penentuan yang telah dijelaskan sebelumnya yaitu Kesibukan, Pengalaman serta Karakter. Penjelasan setiap faktor dijelaskan sebagai berikut:
 - Untuk faktor kesibukan, dapat Anda lakukan wawancara langsung terhadap masing-masing calon anggota dengan harapan mendapatkan fakta yang sebenarnya
 - Untuk faktor pengalaman, dapat Anda lakukan wawancara langsung dengan masing-masing calon anggota, juga dengan minimal 1 teman kerja anggota tersebut dengan harapan dapat memberikan penilaian yang tidak begitu subjektif karena menggunakan 2 sumber.
 - Untuk faktor karakter, dapat Anda lakukan wawancara juga terhadap masing-masing calon anggota, juga melakukan wawancara kepada minimal 2 teman dekat anggota tersebut dengan harapan dapat memberikan objektivitas yang cukup baik.

2. Anda menentukan nilai a , b dan c yaitu variabel-variabel batasan nilai setiap faktor yang ingin anggota Anda kelompok Anda dengan anggota Anda capai. Misalnya $a = 100$, $b = 75$, $c = 85$. Maka, Anda ingin setidaknya jumlah nilai kesibukan mereka minimal 100 poin, pengalaman mereka 75 poin, dan karakter mereka 85 poin.
3. Anda harus menuliskan *list* kebutuhan dari masing-masing faktor yaitu Kesibukan, Pengalaman, dan Karakter
4. Anda sudah menentukan linimasa (*timeline* pekerjaan di dalam *scope* Anda
5. Anda telah membuat tabel yang kolomnya berisi poin pada masing-masing faktor yaitu Kesibukan, Pengalaman, serta Karakter dan juga setiap baris menunjukkan nama dari masing-masing calon anggota.

1	Nama	Kesibukan	Pengalaman	Karakter
2	Adit	4	4	5
3	Rara	-1	8	1
4	Budi	6	0	6
5	Agus	4	4	-4
6	Indah	3	4	7
7	Fani	2	4	3
8	Febrian	4	8	4
9	Cahyo	-1	4	5
10	Anugrah	6	0	7
11	Anggun	9	4	3
12	Ariana	7	8	5

Gambar 1. Tabel Hasil Wawancara setelah Diolah

D. Contoh Implementasi

Kondisi Awal:

Anda adalah seorang Ketua Divisi *Research and Development* dalam suatu Organisasi berbasis IT. Tugas dari divisi ini adalah untuk memastikan setiap acara yang diadakan Organisasi ini mendapatkan feedback yang objektif baik dari penyelenggara yaitu panitianya, maupun dari peserta acara tersebut. Nilai n yang akan Anda gunakan adalah 4.

Persiapan:

1. Anda telah melakukan wawancara terhadap semua calon anggota divisi Anda, serta telah melakukan wawancara juga dengan 1 teman kerja setiap calon anggota serta melakukan wawancara terhadap 2 teman dari masing-masing calon anggota.
2. Anda telah menentukan masing-masing nilai a , b dan c yaitu 15, 10, dan 14 secara berturut-turut.
3. List kebutuhan masing-masing faktor:
 - Kesibukan
Anda ingin agar di dalam divisi Anda setidaknya dalam linimasa kapanpun terdapat 3 orang yang bisa bekerja secara penuh.
 - Pengalaman
Anda membutuhkan orang yang sudah cukup berpengalaman dalam mengolah data, orang jurusan Statistika, serta orang yang sudah memiliki pengalaman dalam mengatur tim
 - Karakter
Anda ingin agar di dalam divisi Anda terdapat orang yang memiliki sifat kepala dingin, ulet dan tekun, serta berkomitmen tinggi akan janjinya.
4. Dari hasil wawancara serta pengolahan data berdasarkan persiapan sebelumnya, maka diperoleh data sebagai berikut:

Dengan menggunakan Algoritma Greedy yang telah dijelaskan sebelumnya, maka akan didapatkan 7 buah tabel sebagai berikut:

1. Kesibukan saja

Nama	Kesibukan	Pengalaman	Karakter
Anggun	9	4	3
Ariana	7	8	5

Gambar 2. Tabel Optimasi Berdasarkan Faktor Kesibukan

2. Pengalaman saja

Nama	Kesibukan	Pengalaman	Karakter
Adit	4	4	5
Rara	-1	4	1
Indah	3	4	7

Gambar 3. Tabel Optimasi Berdasarkan Faktor Pengalaman

3. Karakter saja

Nama	Kesibukan	Pengalaman	Karakter
Anugrah	6	0	7
Indah	3	4	7
Budi	6	0	6

Gambar 4. Tabel Optimasi Berdasarkan Faktor Karakter

4. Kesibukan dan Pengalaman

Nama	Kesibukan	Pengalaman	Karakter
Anggun	9	4	3
Budi	6	0	6
Agus	4	4	-4
Indah	3	4	7

Gambar 5. Tabel Optimasi Berdasarkan Faktor Kesibukan dan Pengalaman

5. Pengalaman dan Karakter

Nama	Kesibukan	Pengalam	Karakter
Anugrah	6	0	7
Indah	3	4	7
Adit	4	4	5
Rara	-1	4	1

Gambar 6. Tabel Optimasi Berdasarkan Faktor Pengalaman dan Karakter

REFERENSI

[1] Munir, R. (2012). *Matematika Diskrit*. Bandung: Informatika Bandung.

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, 26 April 2018

6. Kesibukan dan Karakter

Nama	Kesibukan	Pengalam	Karakter
Anggun	9	4	3
Budi	6	0	6
Cahyo	-1	4	5
Anugrah	6	0	7

Gambar 7. Tabel Optimasi Berdasarkan Faktor Kesibukan dan Karakter



7. Kesibukan, Pengalaman, dan Karakter

Nama	Kesibukan	Pengalam	Karakter
Anggun	9	4	3
Indah	3	4	7
Anugrah	6	0	7
Adit	4	4	5

Gambar 8. Tabel Optimasi Berdasarkan Faktor Kesibukan, Pengalaman dan Karakter

Muhammad Akmal 13517028

V. KESIMPULAN

Ilmu pengujian perangkat lunak merupakan ilmu yang cukup penting dalam bingkai keilmuan pengembangan perangkat lunak. Hal ini dikarenakan dibutuhkan suatu sistem yang dapat memastikan keamanan, kualitas, kebertahanan, hingga efisiensi dari perangkat lunak yang dikembangkan dan ilmu pengujian ini sangat bermanfaat untuk hal tersebut. Salah satu ilmu yang dipakai dalam pengembangan ilmu pengujian perangkat lunak adalah graf yang salah satunya memunculkan konsep graf aliran kontrol atau graf terstruktur. Konsep ini membawa penguji agar bisa memvisualisasikan serta mendapatkan data tes yang sangat mumpuni dalam pengujian baik itu dalam pengujian kode sumber, elemen desain, spesifikasi, hingga kasus penggunaan.

VII. UCAPAN TERIMA KASIH

Penulis mengucapkan syukur kepada Allah SWT yang telah memberikan kemampuan kepada penulis untuk bisa menyelesaikan makalah ini tepat pada waktunya. Penulis juga mengucapkan terima kasih terhadap orangtua yang telah mendukung saya dalam berkembang dan berkuliah di Teknik Informatika ITB serta Ibu Nur Ulfa Maulidevi yang telah memberikan pengajaran terhadap perkuliahan IF 2211 Strategi Algoritma terutama pada pengajaran materi Algoritma Greedy yang saya angkat pada tema makalah kali ini.