Penerapan Algoritma *Branch and Bound* pada Penentuan *Staffing* Organisasi dan Kepanitiaan

Mikhael Artur Darmakesuma - 13515099
Program Studi Teknik Informaitka
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13515099@std.stei.itb.ac.id; mikhael.artur.d@gmail.com

***Abstrak*—Algortima *branch and bound* adalah algoritma untuk persoalan optimasi yang luas pemakaiannya. Makalah ini akan membahasa bagaimana penerapan algoritma tersebut dalam penentuan *staffing* organisasi dan kepanitiaan berdasarkan urutan pilihan divisi. Biasanya penentuan *staffing* tersebut dilakukan secara manual sehingga cukup menghabiskan waktu dan tenaga.**

Kata Kunci—staffing, organisasi, kepanitiaan, branch and bound

I. PENDAHULUAN

Istilah *open recruitment* yang seringkali disingkat menjadi *oprec* mungkin sudah bukan istilah yang asing lagi. Hampir setiap organisasi atau kepanitiaan melakukan *open recruitment* untuk mencari anggota organisasi atau kepanitiaan terkait. *Open recruitment* biasanya dilakukan dengan penyebaran form untuk diisi. Form yang disebar tersebut biasanya berisi urutan pilihan divisi dengan jumlah tertentu.

Setelah *open recruitment* ditutup, akan ada panitia yang menentukan hasil penempatan organisasi atau kepanitiaan yang biasa disebut *staffing*. Panitia tersebut biasanya merupakan kepala divisi manajemen sumber daya manusia (MSDM) yang belum tentu mengenal masing-masing pendaftar pada *open recruitment* tersebut sehingga penentuan hanya dapat dilakukan dengan melihat urutan pilihan divisi dan kebutuhan divisi tanpa melihat kemampuan pendaftar tersebut.

Untuk organisasi atau kepanitiaan dengan jumlah anggota kurang dari 50 orang, hasil *staffing* akan mudah dibuat oleh kepala divisi MSDM. Namun organisasi atau kepanitiaan dengan jumlah anggota lebih dari 100 atau bahkan 200 orang tentu sulit untuk dilakukan secara manual seorang diri. Kenyataannya, banyak organisasi atau kepanitiaan yang anggotanya lebih dari 100 seperti himpunan, acara-acara himpunan dan khususnya kaderisasi himpunan.

Oleh karena itu, pembuatan program yang menerapkan algoritma *branch and bound* untuk menentukan hasil *staffing* dapat membantu meringankan pekerjaan kepala divisi MSDM tersebut.

II. DASAR TEORI

A. Graf

Graf adalah sebuah struktur diskrit yang terdiri dari simpul yang dilambangkan dengan V dan sisi yang menghubungkan simpul-simpul dilambangkan dengan E. Graf pertama kali digunakan untuk menyelesaikan masalah jembatan Konigsberg pada tahun 1736. Secara matematis, sebuah graf G dapat didefinisikan sebagai pasangan dua buah himpunan (V, E) di mana:

1. V merupakan himpunan tidak kosong dari simpul-simpul graf.
2. E merupakan himpunan sisi yang menghubungkan sepasang simpul.

Berdasarkan ada tidaknya gelang atau sisi ganda pada suatu graf, graf dapat dikelompokkan menjadi:

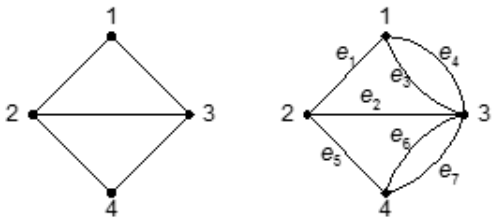
1. Graf sederhana di mana graf tidak memiliki gelang maupun sisi ganda
2. Graf tak-sederhana di mana graf memiliki gelang atau sisi ganda.

Berdasarkan jumlah simpul pada suatu graf, graf dapat dibagi menjadi:

1. Graf berhingga di mana jumlah simpulnya n, berhingga
2. Graf tak-berhingga yaitu graf yang jumlah simpulnya tak berhingga.

Berdasarkan orientasi sisi pada suatu graf. graf dapat dibagi menjadi:

1. Graf tak-berarah yang sisinya tidak memiliki orientasi
2. Graf berarah yang sisinya memiliki orientasi.



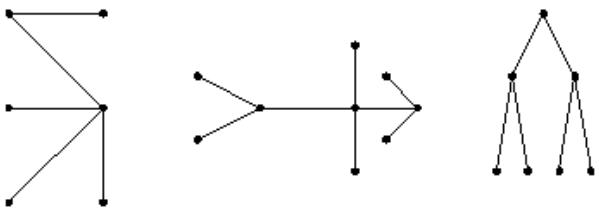
Gambar 2.1 Graf sederhana tak-berarah (kiri) dan graf tak-sederhana tak-berarah (kanan)

Selain definisi, ada beberapa terminologi yang perlu diketahui untuk memahami graf:

1. Bertetangga
   Dua buah simpul dikatakan bertetangga pada graf tak-berarah apabila kedua simpul tersebut dihubungkan langsung oleh sebuah sisi.
2. Bersisian
   Sebuah simpul N dikatakan bersisian dengan sisi E jika sisi E menghubungkan simpul N dengan simpul lain.
3. Simpul Terpencil
   Simpul terpencil adalh simpul yang tidak mempunyai simpul lain yang bertetangga dengan simpul tersebut.
4. Graf Kosong
   Graf yang tidak mempunyai sisi sehingga himpunan sisinya merupakan himpunan kosong adalah graf kosong. Semua simpul yang dimiliki graf kosong adalah simpul terpencil.
5. Derajat
   Derajat sebuah simpul adalah jumlah sisi yang bersisian dengan simpul tersebut.
6. Lintasan
   Lintasan adalah barisan berselang-seling antara simpul dan sisi yang berawal dan berakhir pada sebuah simpul.
7. Sirkuit
   Sirkuit adalah lintasan yang berawal dan berakhir pada simpul yang sama.
8. Terhubung
   Terhubung berarti terdapat lintasan yang menghubungkan sebuah simpul dengan simpul lainnya. Sebuah graf dikatakan terhubung jika seluruh simpulnya terhubung satu sama lain.
9. Upagraf
   Upagraf dari sebuah graf adalah graf yang terdiri dari himpunan simpul dan sisi yang masing masing merupakan himpunan bagian dari himpunan simpul dan sisi graf awal.

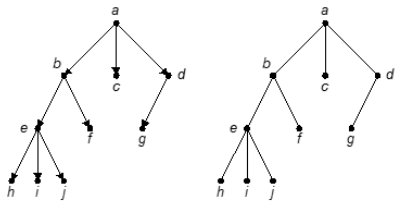
B. Pohon

Pohon adalah graf tak-berarah terhubung yang tidak memiliki sirkuit di mana jumlah sisinya satu lebih sedikit daripada jumlah simpulnya dan seluruh simplnya terhubung dengan sebuah lintasan tunggal. Selain itu, seluruh sisi dalam sebuah pohon merupakan jembatan yang berarti penghapusan sisi tersebut akan menyebabkan pohon awal terbagi menjadi dua buah pohon.



Gambar 2.2 Contoh pohon

C. Pohon Berakar



Gambar 2.3 Pohon berakar, panah pada sisi pohon berakar dapat tidak digambarkan

Pohon berakar adalah pohon yang salah satu simpulnya diperlakukan sebagai akar dan setiap sisinya diberi arah. Untuk memahami pohon berakar, ada beberapa terminologi yang perlu diketahui:

1. Anak
   Sebuah simpul dikatakan anak dari simpul lain jika simpul lain memiliki sisi ke simpul tersebut.
2. Orangtua
   Sebuah simpul dikatakan orangtua dari simpul lain jika simpul lain merupakan anak dari simpul tersebut.
3. Akar
   Akar adalah sebuah simpul yang bukan merupakan anak dari simpul manapun.
4. Keturunan
   Sebuah simpul dikatakan keturunan dari simpul lain jika simpul lain memiliki lintasan ke simpul tersebut.
5. Leluhur
   Sebuah simpul dikatakan leluhur dari simpul lain jika simpul lain merupakan keturunan dari simpul tersebut.
6. Saudara Kandung
   Sebuah simpul dikatakan saudara kandung dari simpul lain jika simpul-simpul tersebut memiliki orangtua yang sama.
7. Upapohon
   Upapohon dari sebuah pohon berakar adalah sebuah upagraf di mana upagraf tersebut memiliki sebuah simpul di dalam pohon berakar dan semua simpul keturunannya.
8. Derajat
   Berbeda dengan derajat dalam graf, derajat dalam pohon berakar berarti jumlah anak dari simpul tersebut.
9. Daun
   Daun adalah simpul yang tidak memiliki anak.
10. Simpul Dalam
    Simpul dalam adalah simpul yang memiliki anak.
11. Aras
    Aras merupakan panjang lintasan dari akar ke simpul tersebut. Aras dari akar adalah 0.
12. Tinggi
    Tinggi adalah aras maksimum sebuah pohon.

D. *Branch and Bound*

Algoritma *branch and bound* adalah algoritma yang mirip dengan algoritma runut-balik karena sama-sama membangun sebuah pohon ruang status. Namun pembentukan pohon ruang statusnya dibangun dengan BFS (Breadth First Search) disertai LCS (*Least Cost Search*).

Selain itu, perbedan lain dapat dilihat dari jenis persoalan yang diselesaikan masing-masing algoritma. Algoritma *branch and bound* biasanya digunakan untuk persoalan optimasi sementara algoritma runut-balik biasanya digunakan untuk persoalan non-optimasi.

Setiap simpul yang dibangun pada pohon ruang status *branch and bound* diberikan bobot yang dinyatakan dengan lambang $\hat{c}$. Bobot tersebut akan dipakai untuk melakukan ekspansi simpul-simpul berikutnya, tidak berdasarkan urutan pembangkitan seperti BFS murni namun dengan mencari simpul dengan bobot terkecil (LCS).

Bobot tersebut dicari dengan sebuah fungsi yang dinamakan fungsi pembatas yang biasanya ditulis sebagai berikut

$$\hat{c} = \hat{f} + \hat{g}$$

di mana

$\hat{c}$ = ongkos untuk mencapai simpul tersebut

$\hat{f}$ = ongkos mencapai simput tersebut dari akar

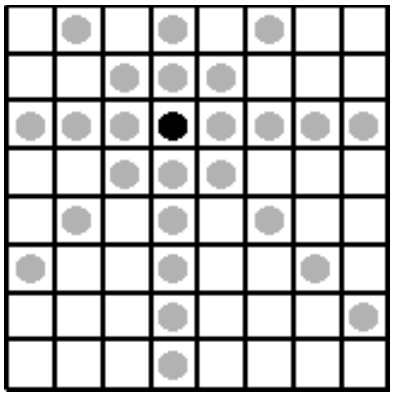
$\hat{g}$ = ongkos mencapai simpul tujuan dari simpul tersebut

Nilai $\hat{c}$ tersebut digunakan untuk menentukan bobot minimum untuk melanjutkan pencarian seperti yang sudah dijelaskan sebelumnya. Secara umum, langkah langkah penyelesaian *branch and bound* adalah sebagai berikut:

1. Masukkan simpul akar ke dalam antrian Q. Jika simpul tersebut adalah solusi, maka solusi telah ditemukan dan proses berhenti.
2. Jika Q kosong maka solusi tidak ditemukan dan proses berhenti.
3. Jika Q tidak kosong, keluarkan simpul dengan $\hat{c}$ minimum dari Q. Jika ada 2 simpul dengan $\hat{c}$ minimum, pilih secara acak atau sesuai perjanjian yang disepakati (misalnya sesuai abjad).
4. Jika simpul tersebut adalah simpul solusi, berarti solusi ditemukan. Jika simpul tersebut bukan simpul solusi, bangkitkan anak-anaknya. Jika simpul tersebut tidak memiliki anak, kembali ke langkah 2.
5. Untuk setiap anak dari simpul tersebut hitung bobotnya dengan fungsi pembatas dan masukkan semua simpul anak ke dalam Q.
6. Kembali ke langkah 2.

Banyak persoalan yang dapat diselesaikan dengan algoritma *branch and bound*. Beberapa di antaranya adalah persoalan n buah ratu dalam permainan catur, permainan 15-puzzle, integer knapsack problem, *Travelling Salesperson Problem* (TSP) dan beberapa persoalan lainnya.

Dalam persoalan TSP, ada beberapa algoritma *branch and bound* yang lazim digunakan untuk menentukan bobot sebuah simpul. Algoritma tersebut adalah algoritma *Reduced Cost Matrix* dan Bobot Tur Lengkap.



Gambar 2.3 Persoalan n-ratu dengan n = 8

E. *Staffing*

Staffing adalah fungsi manajemen menempatkan orang pada sebuah struktur organisasi dan menjaga agar tidak terjadi kekosongan pada struktur organisasi tersebut dengan tujuan utama menempatkan orang yang tepat pada posisi yang tepat. Namun, pada makalah ini, definisi *staffing* lebih dititikberatkan pada penempatannya saja.

Kebutuhan		
Keamanan	Medik	Mentor
3	1	2

Tabel 2.1 Kebutuhan panitia

Staffing pada makalah ini dibatasi lingkupnya pada pendaftaran kepanitiaan atau organisasi dengan sistem *open recruitment* yang sudah dijelaksan sebelumnya.

Nama	Pilihan 1	Pilihan 2
Andi	Keamanan	Medik
Budi	Mentor	Medik
Cindy	Medik	Mentor
Dono	Keamanan	Medik
Erni	Mentor	Medik
Fahmi	Mentor	Keamanan

Tabel 2.2 Hasil *open recruitment*

Definisi hasil *staffing* optimal pada makalah ini adalah hasil *staffing* yang paling sesuai dengan pilihan pendaftar dan mencukupi kebutuhan kepanitiaan atau organisasi.

Nama	Divisi
Andi	Keamanan
Budi	Mentor
Cindy	Medik
Dono	Keamanan
Erni	Mentor
Fahmi	Keamanan

Tabel 2.3 Contoh tabel hasil *staffing*

III. *BRANCH AND BOUND* UNTUK STAFFING

Dari tabel 2.1 dan 2.2 pada bab sebelumnya, kita dapat membuat struktur tabel yang lebih mudah diimplementasikan dalam prrogram yaitu dengan memisalkan pilihan "Keamanan" sebagai 0, "Medik" sebagai 1 dan "Mentor" sebagai 2. Pilihan-pilihan tersebut diubah bentuknya menjadi sebuah kolom beserta sebuah kolom nama. Sementara untuk masing-masing pendaftar, akan diisi 1 pada kolom sesuai prioritas pilihan pertama, 2 pada kolom sesuai prioritas pilihan ke dua dan 99 untuk merepresentasikan pilihan yang tidak dipilih. Sehingga didapat tabel baru sebagai berikut:

Nama	Keamanan	Medik	Mentor
Andi	1	2	99
Budi	99	2	1
Cindy	99	1	2
Dono	1	2	99
Erni	2	99	1
Fahmi	99	2	1

Tabel 3.1 Konversi hasil *open recruitment*

Selain itu, tabel 2.1 yang berisi kebutuhan panitia dapat diubah menjadi sebuah *queue* berisi kebutuhan panitia sesuai pemisalan yang sudah dibuat sebelumnya. *Queue* yang akan terbentuk dari tabel 2.1 adalah 1-1-1-2-3-3 yang berarti 3 opsi 1 (Keamanan), 1 opsi 2 (Medik) dan 2 opsi 3 (Mentor). *Queue* tersebut nantinya akan berguna untuk menghitung ongkos untuk mencapai suatu simpul dari akar.

Untuk persoalan ini dibuat fungsi pembatas tersendiri sebagai berikut

$$\hat{c} = \hat{f} + \hat{g}$$

di mana

$\hat{c}$ = bobot untuk simpul tersebut
$\hat{f}$ = jumlah prioritas pendaftar yang sudah dimasukkan
$\hat{g}$ = jumlah langkah yang diperlukan untuk mencapai solusi.

Misalkan untuk sebuah kasus dengan *queue* kebutuhan 1-2-3-3 dan sebuah simpul sudah mengambil Andi kemudian Budi. Karena Andi memilih pilihan 1 (Keamanan) pada prioritas 1 dan Budi memilih pilihan 2 (Medik) pada prioritas 2, maka $\hat{f}$ simpul tersebut adalah 1 + 2 = 3. Untuk kasus hanya dibutuhkan 4 orang panitia sesuai *queue* yang tadi sudah disebutkan maka langkah yang masih perlu dilakukan adalah

$$(Kebutuhan\ panitia) - (Panitia\ yang\ sudah\ diambil)$$

yaitu 4 – 2 = 2. Sehingga $\hat{g}$ dari simpul tersebut adalah 2 yang berarti $\hat{c}$ simpul tersebut adalah 3 + 2 = 5.

Secara umum, langkah-langkah yang perlu ditempuh untuk memperoleh hasil *staffing* yang paling optimal adalah sebagai berikut:

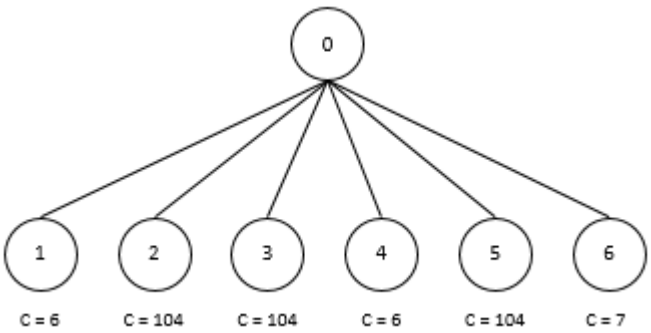
1. Mulai dengan memasukkan simpul akar yang mewakili kondisi belum menempatkan pendaftar sama sekali ke dalam Q. Di mana $\hat{f}$ = 0 karena belum ada yang ditempatkan dan $\hat{g}$ = n di mana n adalah jumlah kebutuhan panitia. Pada kasus ini simpul akar tidak mungkin merupakan simpul solusi.
2. Setiap simpul pada aras ke-i menyatakan pendaftar yang terambil untuk mengisi posisi ke-i pada *queue* kebutuhan.
3. Kemudian keluarkan dan ekspansi simpul tersebut. Hitung pula bobotnya untuk masing-masing pendaftar yang belum ditempatkan sehingga simpul akar memiliki m buah anak dan simpul pada aras ke-i memiliki m – i buah anak di mana m adalah jumlah pendaftar. Masukkan anak-anak simpul tersebut ke Q.
4. Ulangi langkah ke 3 untuk simpul dengan jumlah bobot ($\hat{c}$) minimum pada pohon tersebut hingga didapat simpul solusi, jika ada simpul non-solusi yang belum diekspansi dan bobotnya lebih kecil dari simpul daun, ulangi langkah 3 dan 4 dengan simpul tersebut. Jika tidak ada maka simpul solusi dengan bobot terkecil adalah solusi optimal.

IV. CONTOH KASUS

Contoh kasus diambil dari tabel 2.1 dan 2.2 pada bab 2 di mana dibutuhkan 3 orang Keamanan, 1 orang Medik dan 2 orang Mentor dengan hasil *open recruitment* seperti pada tabel 2.2. Dalam contoh kasus ini, untuk simpul yang bobotnya sama akan diambil simpul yang arasnya lebih besar, untuk simpul yang bobot dan arasnya sama akan diambil simpul yang lebih dulu dibangkitkan.

Pertama-tama dibuat *queue* pilihan yang dibutuhkan kepanitiaan atau organisasi yaitu 1-1-1-2-3-3 di mana 1 mewakili pilihan keamanan, 2 mewakili pilihan medik dan 3 mewakili pilihan mentor.

Kemudian dibangkitkan simpul akar 0 dan dilakukan ekspansi untuk simpul 0 sehingga menghasilkan 6 simpul anak yaitu simpul 1 hingga simpul 6 yang memiliki bobot berturut-turut 6, 104, 104, 6, 104 dan 7.



Gambar 4.1 Anak-anak simpul akar

Bobot didapatkan demikian dari fungsi pembatas yang ditentukan sebelumnya pada bab 3. Untuk simpul 1 dan 4, pendaftar memilih keamanan pada pilihan pertamanya, oleh karena itu nilai dari $\hat{f}$ simpul tersebut adalah 1.

Untuk simpul 2, 3 dan 5, pendaftar tidak memilih keamanan pada opsi pilihannya sehingga nilai dari $\hat{f}$ simpul tersebut adalah 99.

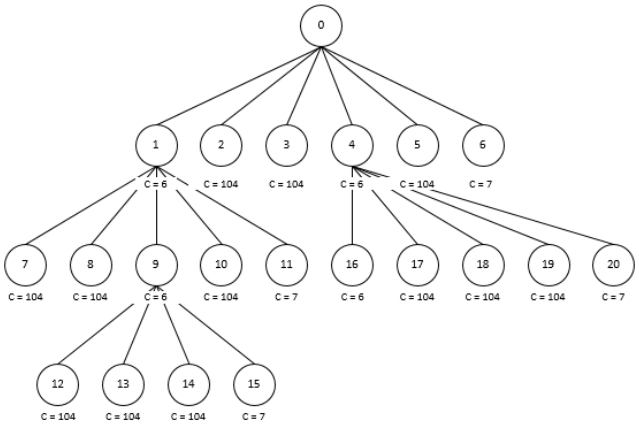
Untuk simpul 6, pendaftar memilih keamanan pada opsi ke 2 sehingga nilai dari $\hat{f}$ simpul tersebut adalah 2. Untuk aras 1 dan jumlah kebutuhan panitia 6, nilai $\hat{g}$ seluruh simpul adalah 5. Sehingga didapatkan bobot berturut-turut 6, 104, 104, 6, 104 dan 7.

Simpul 1 dan 4 memiliki aras dan bobot yang sama. Sesuai kesepakatan di awal, diambil simpul yang dibangkitkan terlebih dahulu yaitu simpul 1. Anak-anak dari simpul 1 dibangkitkan yaitu simpul 7 hingga 11. Bobot minimum pada anak-anak simpul tersebut adalah 6 oleh simpul 9.

Bobot simpul 4 sama dengan simpul 9. Sesuai kesepakatan, untuk bobot sama dan aras berbeda, diekspansi simpul yang arasnya lebih besar yaitu simpul 9.

Hasil ekspansi dari simpul 9 menghasilkan 4 simpul anak yaitu simpul 12 hingga 15. Bobot minimum pada anak-anak simpul 9 adalah 7 oleh simpul 15.

Karena masih ada simpul dengan bobot lebih kecil yaitu simpul 4, maka simpul 4 diekspansi terlebih dahulu sehingga menghasilkan simpul anak 16 hingga 20 yang memiliki bobot minimum 6 oleh simpul 16.



Gambar 4.2 Ekspansi dilakukan pada simpul dengan bobot terkecil

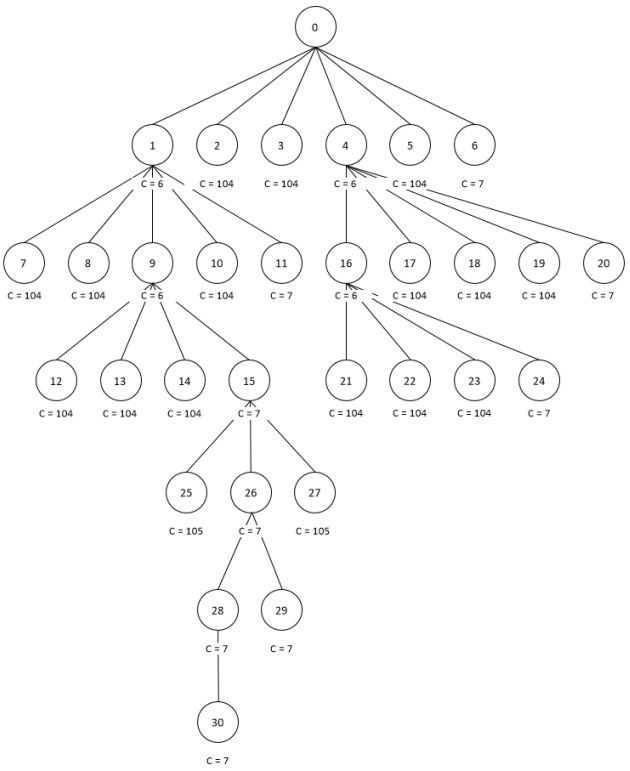
Karena bobot minimum dalam Q adalah 6, maka ekspansi simpul 16 sehingga memperoleh simpul 21 hingga 24 yang memiliki bobot minimum 7 oleh simpul 24.

Bobot minimum dimiliki oleh simpul 15 dan 24 yang memiliki aras sama dengan 2, sesuai kesepakatan diekspansi simpul yang dibangkitkan terlebih dahulu yaitu simpul 15.

Simpul 15 diekspansi dan memiliki 3 simpul anak yaitu 25, 26 dan 27.

Simpul 26 merupakan simpul dengan bobot minimum 7 dan aras terbesar sehingga simpul tersebut diekspansi dan menghasilkan 2 simpul anak yaitu 28 dan 29 yang sama-sama memiliki bobot 7 di mana 7 adalah bobot minimum.

Sesuai kesepatakan, ekspansi simpul yang lebih dulu dibangkitkan yaitu simpul 28 yang menghasilkan 1 simpul anak yaitu simpul 30 dengan bobot 7. Karena tidak ada simpul lain yang memungkinkan simpul solusi dengan bobot lebih kecil dari 7, maka simpul 30 merupakan simpul solusi.



Gambar 4.4 Pohon ruang status untuk kasus tabel 2.1 dan 2.2. Didapat hasil staffing optimal 1-4-6-3-2-5 untuk *queue 1-1-1-2-3-3*

V. KESIMPULAN

Algoritma *branch and* bound dapat digunakan untuk menyelesaikan berbagai macam persoalan. Namun, algoritma *branch and bound* tidak selalu merupakan solusi terbaik bagi sebuah permasalahan dalam konteks kompleksitas waktu dan memori.

Penentuan *staffing* optimal dapat dilakukan dengan algoritma *branch and bound*, namun hanya efektif untuk mencari satu solusi saja. Hal tersebut disebabkan *staffing* merupakan persoalan kombinasi bukan permutasi sehingga urutan tidak mempengaruhi untuk masing-masing divisi yang sama.

Misalnya hasil untuk contoh kasus pada bab sebelumnya adalah 1-4-6-3-2-5 untuk *queue* 1-1-1-2-3-3. Maka 3 urutan terdepan dapat ditukar-tukar menjadi 1-6-4, 4-6-1, 4-1-6, 6-4-1 dan 6-1-4. Hal yang sama juga dapat dilakukan untuk 2 urutan terakhir yaitu menjadi 5-2.

Pada kasus tersebut, jika dilakukan pencarian seluruh kemungkinan maka akan terdapat 3! x 1! x 2! solusi yang sebenarnya sama satu sama lain hanya berbeda urutannya. Maka untuk n buah pilihan dan k(n) adalah kebutuhan anggota pilihan ke-n, akan ada

$$p = k(1)! \times k(2)! \times \ldots \times k(n-1)! \times k(n)!$$

kali pengulangan untuk setiap solusi. Sehingga bila didapatkan x buah solusi, maka solusi sebenarnya hanya ada $\frac{x}{p}$ buah solusi.

UCAPAN TERIMA KASIH

Pertama-tama penulis mengucapkan syukur kepada Tuhan yang Maha Esa, atas bimbingan-Nya penulis dapat menyelesaikan makalah ini. Penulis juga berterima kasih kepada, Dr. Nur ulva Maulidevi, ST., MT. dan Dr. Ir. Rinaldi Munir, MT. atas bimbingannya dalam mata kuliah IF2211 sehingga ilmu yang didapatkan dalam perkuliahan dapat digunakan untuk menyelesaikan makalah ini. Penulis juga berterima kasih kepada teman-teman yang sudah membantu mulai dari penentuan topik hingga selesai.

REFERENSI

[1] Munir, Rinaldi. *Matematika Diskrit*, Bandung : Informatika, 2006.

[2] Munir, Rinaldi. *Strategi Algoritma*. Bandung : Informatika, 2006.

[3] Harold D. Koontz dan Cyril J. O'Donnell. *Principles of Management: An Analysis of Managerial Function*, Tokyo: McGraw-Hill Kogakusha, 1972.

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, 19 Mei 2017

Mikhael Artur Darmakesuma - 13515099