Penggunaan Algoritma Branch and Bound dalam Pencarian Jalur Penagihan Pelanggan

Richard Wellianto (13514051)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
richard.wellianto@hotmail.com

***Abstract*—Makalah ini akan membahas bagaimana menggunakan algoritma Branch and Bound pada persoalan graf lokasi penagihan dan titik awal dan akhir penagihan (markas). Algoritma Branch and Bound dapat digunakan untuk menyelesaikan permasalahan graf dengan optimal. Graf yang dipakai adalah graf yang memilki restriksi pada simpul, yang membuat simpul hanya dapat dikunjungi dalam suatu kondisi**

Keywords—Branch and Bound, Penagihan, Graf, Restriksi

I. PENDAHULUAN

Persoalan penagihan adalah salah satu persoalan yang penulis miliki saat berada di kampung halaman ketika membantu orang tua. Bagaimana cara mendapatkan jalur penagihan terbaik yang memperhatikan profit dan waktu? Setelah mengikuti kuliah Strategi Algoritma, penulis mendapatkan ide untuk membuat lokasi penagihan menjadi model graf.

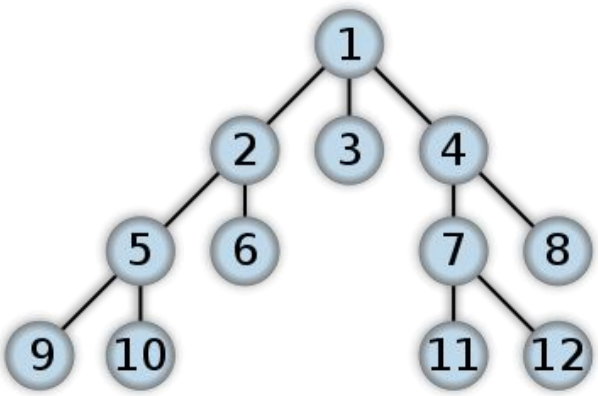
Persoalan graf adalah persoalan mencari jalur pada graf untuk menyelesaikan sebuah masalah sehingga jalur yang didapatkan menjadi jalur yang optimal. Persoalan ini dapat diselesaikan dengan beberapa jenis algoritma seperti algoritma BFS, algoritma DFS, algoritma Backtracking, dan algoritma Branch and Bound. Dalam hal ini penulis menggunakan algoritma Branch and Bound karena jalur penagihan lebih baik didapatkan menggunakan algoritma yang melebar, dan dikurangi pelebarannya dengan menggunakan fungsi pembatas atau bobot.

II. DASAR TEORI

A. Algoritma BFS

Algoritma BFS adalah salah satu algoritma yang digunakan dalam penelusuran sebuah graf. BFS adalah singkatan dari Breadth First Search, yang merupakan teknik penelusuran graf secara melebar. Algoritma ini bersifat blind-search, sehingga tidak ada informasi tambahan yang dapat digunakan untuk mempermudah pencarian.

Berikut adalah graf yang merepresentasikan BFS



Gambar Langkah-langkah Algoritma BFS
(Sumber : http://www.programmerinterview.com/index.php/data-structures/dfs-vs-bfs/)

Graf yang dapat ditelusuri menggunakan algoritma BFS dapat berupa graf statis atau graf dinamis. Graf statis adalah graf yang terbentuk sebelum dilakukan pencarian dan tidak berubah-ubah selama pencarian berlangsung. Graf dinamis adalah graf yang terbentuk saat proses pencarian dilakukan. Graf dinamis dapat berubah-ubah sesuai simpul dalam graf yang dikunjungi.

Untuk mengecek apakah sebuah algoritma pencarian graf lebih baik daripada yang lain, digunakan evaluasi teknik pencarian. Evaluasi ini terdiri dari 4 faktor, yaitu:

- Completeness : apakah algoritma yang dipakai dapat menjamin ditemukannya solusi jika graf memang memiliki solusi.
- Optimality : apakah teknik yang dipakai dapat menjamin ditemukannya solusi optimal pada graf.
- Kompleksitas waktu : lama waktu yang dibutuhkan untuk menentukan solusi
- Kompleksitas ruang : memory yang digunakan untuk melakukan pencarian

Kompleksitas waktu dan ruang dapat ditentukan dengan beberapa variabel, yaitu branching factor (b), yang merupakan

maksimum percabangan yang mungkin dari satu simpul, depth (d), yang merupakan kedalaman dari solusi terbaik (cost terendah), dan m, yang merupakankedalaman maksimum dari ruang status, m bisa saja bernilai tak hingga.

Dari evaluasi teknik pencarian, BFS memiliki completeness selama nilai b terbatas, artinya cabang yang digenerate memiliki nilai maksimum. BFS juga memiliki Optimality, jika banyak langkah yang dibutuhkan untuk menemukan solusi dapat dianggap sebagai cost. Kompleksitas waktu BFS membesar sebanyak polinomial karena dari 1 node dapat bercabang menjadi b subnode, lalu dari 1 subnode tersebut dapat bercabang menjadi b sub-subnode yang membuat munculnya maksimum b^2 sub-subnode, dst. Sehingga kompleksitas waktu dari BFS adalah $1 + b + b^2 + b^3 + ... + b^d = O(b^d)$. Dengan alasan yang sama dengan kompleksitas waktu, BFS juga memiliki kompleksitas ruang $O(b^d)$. Disini dapat dilihat bahwa BFS memakan ruang cukup banyak karena meng-*generate* cabang yang tidak perlu.

B. Algoritma Branch & Bound

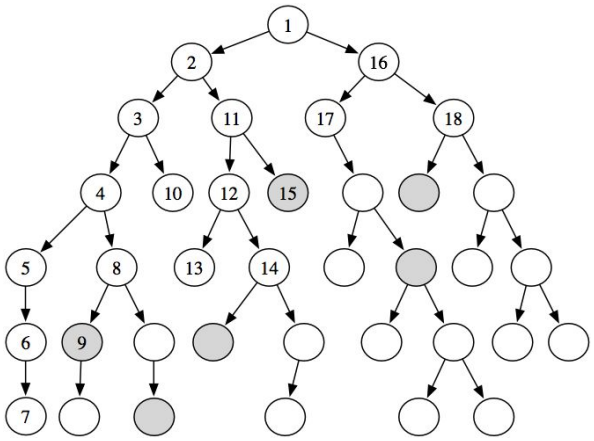
Branch and Bound adalah algoritma pencarian solusi pada graf yang menggunakan suatu bobot pada Breadth First Search (BFS). Bobot, yang bisa diibaratkan sebagai fungsi, ditentukan oleh pencari solusi dari masalah yang ingin diselesaikan. Bobot harus bisa diukur sehingga dapat dibandingkan satu sama lain.

Misalnya untuk persoalan knapsack, bobot dapat memiliki cara pendapatan yang berbeda tergantung dari solusi optimal yang diinginkan. Jika solusi yang diinginkan adalah semakin banyak jenis barang yang dibawa, maka bobot hanya perlu mempertimbangkan jumlah jenis barang dan sisa ruang pada knapsack tersebut. Tetapi jika solusi yang diinginkan adalah profit sebanyak mungkin, maka yang perlu dipertimbangkan adalah profit dan sisa ruang pada knapsack tersebut. Biarpun objek yang digunakan sama, bobot yang dimiliki bisa berbeda.

Nama Branch and Bound sendiri sudah sedikit memberikan bayangan bagaimana algoritma ini bekerja. Seperti yang sudah dijelaskan sebelumnya, algoritma Branch and Bound menggunakan algoritma BFS dalam mencari solusi yang diinginkan. BFS adalah algoritma dimana dilakukan pembuatan cabang baru untuk mengecek setiap solusi yang mungkin. Dari sini dapat disimpulkan bahwa “Branch” berasal dari BFS yang digunakan.

Sedangkan Bound, berasal dari bobot yang sudah dijelaskan sebelumnya. Dalam algoritma ini, bobot memegang peran penting karena urutan pengecekan simpul didapatkan dari seberapa dekat bobot simpul dengan nilai tujuan yang diinginkan. Semakin dekat bobot simpul dengan nilai tujuan yang diinginkan, atau yang bisa disebut solusi terbaik (belum tentu terjadi), maka simpul tersebut akan dicek terlebih dahulu dan dibangkitkan cabang-cabangnya terlebih dahulu. Jika simpul yang baru dibangkitkan tersebut memiliki bobot yang lebih mendekati ke solusi terbaik dibandingkan simpul yang belum dicek, simpul tersebut akan dicek terlebih dahulu biarpun memiliki kedalamn yang lebih dalam dari simpul yang belum dicek. Dari sini dapat disimpulkan bahwa “Bound” berasal dari bobot yang digunakan.

Berikut adalah contoh sebuah graf yang merepresentasikan Branch and Bound



Gambar 2 Penggunaan algoritma Branch and Bound untuk menyelesaikan masalah.
(Sumber : http://artint.info/figures/ch03/sgraph_bb.png)

Dari Gambar 2 dapat dilihat bahwa nomor pada simpul menunjukkan urutan pengecekan pada simpul. Hal ini menunjukkan bahwa simpul di bagian kiri pada gambar tersebut memiliki bobot yang mendekati solusi dibanding simpul di bagian kanan. Perlu diingatkan lagi bahwa urutan pembangkitan tidak berpengaruh banyak pada algoritma Branch and Bound. Urutan hanya diperhatikan ketika 2 simpul yang berbeda memiliki bobot yang sama.

Jika 2 simpul yang berbeda memiliki bobot yang sama, maka simpul yang akan dicek terlebih dahulu adalah simpul yang dibangkitkan lebih dulu. Biasanya simpul yang berada di bagian kiri dianggap sebagai simpul yang dibangkitkan lebih awal dibandingkan simpul di bagian kanan, biarpun simpul-simpul tersebut dibangkitkan secara bersama-sama dari simpul sebelumnya. Pencarian berakhir jika kondisi untuk memenuhi solusi telah tercapai atau tidak ada solusi terbaik lain yang mungkin dalam permasalahan tersebut (kasus terburuk).

III. PENYELESAIAN MASALAH PENAGIHAN

Langkah-langkah penyelesaian masalah penagihan adalah sebagai berikut

A. Penentuan bobot simpul

Dalam permasalahan ini, solusi yang diinginkan adalah sebisa mungkin profit yang didapatkan sebanding dengan waktu yang dibutuhkan untuk mendapatkannya. Selain itu, akan lebih baik jika profit yang didapat nilainya sebanyak mungkin. Dari sini disimpulkan bahwa bobot dipengaruhi oleh 3 hal. Pertama adalah total profit yang didapat. Kedua adalah total waktu yang dibutuhkan untuk mencapai profit tersebut berdasarkan jalur yang telah dilalui. Ketiga adalah banyak simpul yang masih dapat dilalui dari simpul tersebut. Poin ketiga diperlukan untuk memperbanyak jumlah simpul yang akan dicek. Dari ketiga parameter ini, dapat disimpulkan fungsi bobot yang dipakai menjadi

$$f = \frac{n\sum profit}{\sum waktu} \quad \text{.....(1)}$$

n pada (1) adalah banyak simpul yang dapat dikunjungi dari simpul saat ini. Rumus ini dipikirkan penulis karena akan membuat bobot memiliki tendensi menurun antara simpul pembangkit dan simpul yang dibangkitkan

Kedalaman maksimum dari graf ini adalah banyak simpul. Cabang akan dibangkitkan jika simpul berikutnya dapat dilalui dari simpul saat ini. Selalu ada cabang untuk kembali ke simpul awal, tapi karena simpul awal adalah markas, maka profitnya adalah 0, yang membuat simpul ini hampir tidak mungkin dilewati kecuali kembali ke markas adalah pilihan yang paling memungkinkan untuk mencapai profit yang sepadan dengan waktu yang dibutuhkan.

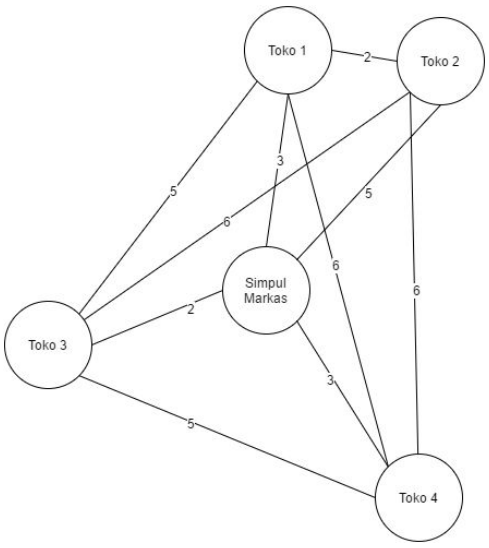
Dalam permasalahan ini hanya ada satu sirkuit yang dilalui (dari simpul awal dan kembali ke simpul awal) sehingga jika penagih telah mencapai simpul awal maka penagih telah keluar dari masalah yang baru diselesaikan.

Jika penagih mengecek simpul markas, maka bobot hanya ditentukan oleh total profit dan waktu yang dipakai untuk melakukan penagihan. Fungsi bobot pada simpul markas :

$$f = \frac{\sum profit}{\sum waktu} \quad \text{.....(2)}$$

B. Graf yang dipakai

Dalam masalah penagihan suatu tempat dapat didatangi dari tempat-tempat yang lain, yang membuat graf yang dipakai adalah graf lengkap yang memiliki simpul sebanyak jumlah tempat penagihan ditambah dengan 1, yang merupakan simpul markas atau simpul awal. Untuk memudahkan penentuan jalan, sebuah simpul hanya dapat dilewati sebanyak satu kali. Simpul awal tidak memiliki restriksi waktu sehingga dapat dilewati kapan saja.



Gambar 3 Graf jarak antar toko dan simpul markas

Simpul	Restriksi	Profit
Toko 1	4 - 8, 10 - 11	Rp 500.000
Toko 2	1 - 5, 7 - 9, 11 - 12	Rp. 300.000
Toko 3	2 - 5, 6 - 8, 10 - 13	Rp. 750.000
Toko 4	9 - 11	Rp. 2.000.000
Markas	Dapat dikunjungi kapan saja	-

Tabel 1 Restriksi dan profit pada simpul

Angka pada cabang pada graf di gambar 3 menandakan waktu yang dibutuhkan untuk mencapai simpul yang dihubungkan oleh cabang tersebut dengan simpul saat ini. Saat ini tidak ada satuan pada angka tersebut, tapi nantinya angka tersebut dapat memiliki satuan waktu (misalnya menit). Simpul simpul menandakan tempat-tempat yang dapat dikunjungi. Simpul-simpul tersebut terdiri atas simpul markas yang merupakan awal dan akhir perjalanan dan simpul toko yang akan dikunjungi untuk mendapatkan profit.

Restriksi pada tabel 1 menandakan waktu dimana toko tersebut dapat dikunjungi. Waktu tersebut akan dibandingkan dengan total waktu yang telah ditempuh untuk mencapai simpul tersebut. Penagih tak dapat mengunjungi simpul tersebut jika total waktu yang telah ditempuh tidak termasuk dalam restriksi tersebut, yang menandakan toko tersebut sedang tutup. Misalnya, penagih dari simpul markas ingin mengunjungi toko 1 terlebih dahulu. Total waktu yang dipakai untuk mengunjungi toko 1 adalah 3, sedangkan restriksi pada toko 1 adalah 4 - 8, 9 - 11, yang menandakan toko tidak buka pada waktu 3, yang membuat penagih tak dapat mendatangi toko tersebut.

Contoh lain adalah penagih setelah mengunjungi toko 2 dan toko 1, akan mengunjungi toko 3. Total waktu yang dipakai untuk mengunjungi toko 3 dari toko 2 yang sebelumnya ditempuh menggunakan jalur Simpul Markas - toko 1 - toko 2 adalah 4 + 2 + 5 = 11. Restriksi pada toko 3 adalah 2 - 5 , 6 - 8, 9 - 13. Karena 11 berada di antara 9 dan 13, maka simpul tersebut dapat dikunjungi oleh penagih. Karena markas dapat dikunjungi kapan saja untuk pulang menyetor uang, maka markas tidak memiliki restriksi waktu.

Profit adalah total uang yang didapat saat menagih toko. Tujuan dari penagihan adalah mendapatkan profit sebanyak-banyaknya dengan total waktu sesedikit mungkin, yang membuat penagihan menjadi lebih efisien. Dalam hal ini, waktu yang dibutuhkan untuk mendapatkan profit pada simpul, atau waktu tunggu pada toko, tidak diperhitungkan mengingat kondisi pada setiap toko berbeda-beda dan dapat berubah pada waktu yang lain. Kondisi dimana toko buka dan tutup tidak pada waktunya juga tidak diperhitungkan pada graf tersebut, sehingga toko selalu buka dan tutup tepat waktu.

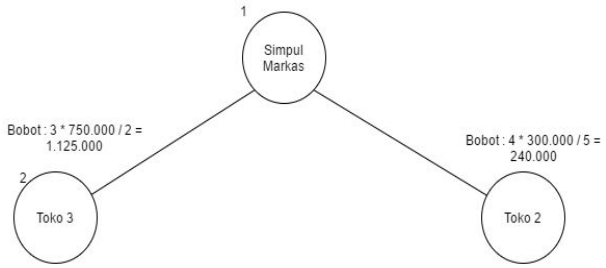
Asumsi lain yang dipakai adalah waktu yang digunakan untuk menempuh simpul lain dari simpul saat ini sama dengan waktu yang digunakan untuk menempuh simpul saat ini dari simpul lain tersebut, yang membuat graf yang dibuat tidak berarah. Normalnya, ada perbedaan antara kedua waktu tersebut, bisa karena jalan satu arah (contohnya ketika toko 1

berada di pangkal jalan, dan toko 2 berada pada ujung jalan, yang membuat penagih mudah berjalan dari toko 1 ke toko 2, tetapi harus memutar untuk ke toko 1 dari toko 2), atau perbedaan kondisi jalan (ada yang menikah di pinggir jalan, atau ada perbaikan jalan). Selain itu, diasumsikan juga penagih tidak istirahat sepanjang jalan, entah untuk makan dan/atau mengisi bensin dan/atau ada kerusakan pada kendaraan.

Kondisi jalan juga diasumsikan telah diperbaharui sesuai dengan kondisi terbaru yang ada pada jalan, sehingga tidak ada perubahan kondisi secara tiba-tiba ketika sedang melakukan penagihan. Misalnya jalan X tiba-tiba amblas yang membuat penagih tak bisa mendatangi toko Y dengan jalur yang dilalui, sehingga waktu yang dibutuhkan untuk mencapai toko Y menjadi berbeda dari waktu sebelumnya. Jalan juga dianggap tidak memiliki *rush hour* dan lampu lalu lintas tidak merubah waktu yang dibutuhkan untuk mengunjungi simpul.

C. Penggunaan algoritma Branch & Bound

Awalnya, penagih berada pada simpul markas, dengan total profit yang telah didapatkan dan total waktu yang telah ditempuh adalah 0. Penagih tak bisa ke toko 1 karena ketika penagih mencapai toko 1 dari simpul markas, total waktu yang digunakan adalah 3, yang tidak berada pada restriksi pada toko 1 (4 - 8, 10 - 11). Penagih juga tak bisa ke toko 4 karena ketika penagih mencapai toko 4 dari simpul markas, total waktu yang digunakan adalah 3, yang tidak berada pada restriksi pada toko 4 (9 - 11). Sehingga penagih hanya bisa pergi ke toko 2 dan toko 3.

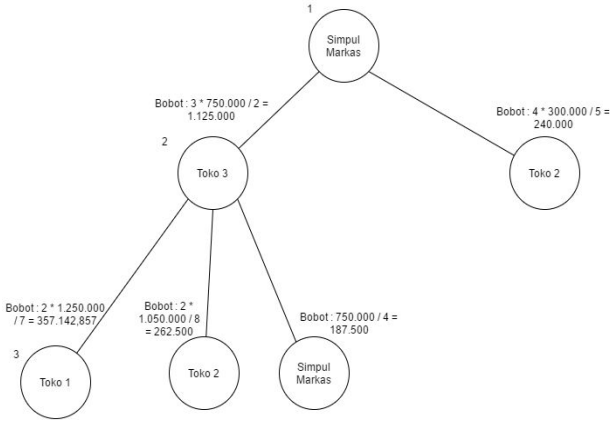


Gambar 4 Pembangkitan pertama

Penagih dapat pergi ke toko 2 karena ketika penagih mencapai toko 2 dari simpul markas, total waktu yang digunakan adalah 5, yang berada pada restriksi pada toko 2 (1 - 5, 7 - 9, 11 - 12) mengingat 5 berada di antara 1 dan 5. Penagih dapat pergi ke toko 3 karena ketika penagih mencapai toko 3 dari simpul markas, total waktu yang digunakan adalah 2, yang berada pada restriksi pada toko 3 (2 - 5, 6 - 8, 10 - 13). Selanjutnya, penagih akan mengecek toko 3 terlebih dahulu karena bobot toko 3 lebih besar dari toko 2, seperti pada gambar 4. Simpul tersebut akan dinamai simpul 2 mulai saat ini.

Profit total saat mencapai simpul 2 adalah 750.000. Waktu total saat mencapai simpul 2 adalah 2. Dari simpul 2 (toko 3), penagih dapat pergi ke simpul markas yang selalu bisa dikunjungi, toko 1 karena total waktu yang digunakan untuk mencapai toko 1 dari simpul 2 adalah 2 + 5 = 7, dan 7 berada pada restriksi toko 1 (4 - 8, 10 - 11) mengingat 7 berada di antara 4 dan 8, dan toko 2 karena total waktu yang digunakan untuk mencapai toko 2 dari simpul 2 adalah 2 + 6 = 8, dan 8 berada pada restriksi toko 2 (1 - 5, 7 - 9, 11 - 12) mengingat 8 berada di antara 7 dan 9. Sehingga banyak toko yang dapat dicapai dari simpul 2 adalah 3.

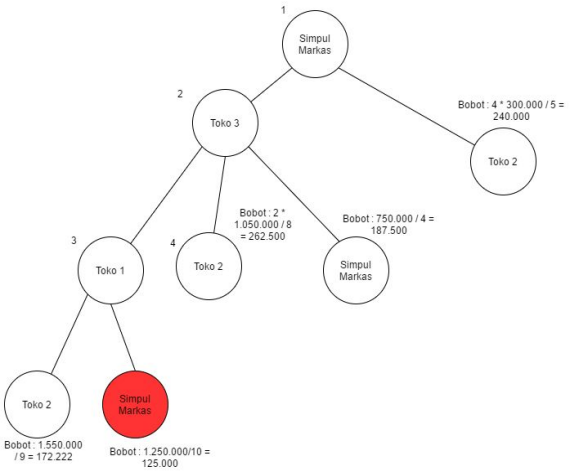
Selanjutnya dibangkitkan 3 simpul, yaitu simpul toko 1, simpul toko 2, dan simpul markas. Penagih akan mengecek toko 1 yang dibangkitkan dari simpul 2 terlebih dahulu karena bobot toko 1 tersebut lebih besar dari bobot toko yang lain, seperti pada gambar 5. Simpul tersebut mulai saat ini akan disebut simpul 3.



Gambar 5 Pembangkitan kedua

Profit total saat mencapai simpul 3 adalah 1.250.000. Waktutotal saat mencapai simpul 3 adalah 7. Dari simpul 3 (toko 1), penagih dapat pergi ke simpul markas yang selalu bisa dikunjungi dan toko 2 karena total waktu yang digunakan untuk mencapai toko 2 dari simpul 3 adalah 7 + 2 = 9, dan 9 berada pada restriksi toko 2 (1 - 5, 7 - 9, 11 - 12) mengingat 9 berada di antara 7 dan 9.

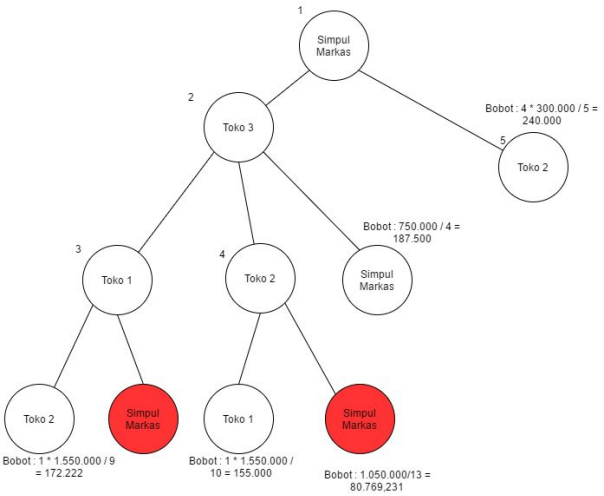
Selanjutnya dibangkitkan 2 simpul, yaitu simpul toko 2 dan simpul markas. Penagih akan mengecek toko 2 yang dibangkitkan dari simpul 2 terlebih dahulu karena bobot toko 2 tersebut lebih besar dari bobot toko yang lain, seperti pada gambar 6. Simpul tersebut mulai saat ini akan disebut simpul 4. Perhatikan simpul markas yang dibangkitkan dari simpul 3 diberikan warna merah pada gambar 6. Hal ini disebabkan bobot pada simpul markas adalah solusi yang dicari, dan hanya bobot terbesar yang akan dijadikan solusi permasalahan ini. Sehingga simpul markas yang tidak mungkin dijadikan solusi (yang bukan simpul markas berbobot terbesar) diberi tanda sebagai simpul mati.



Gambar 6 Pembangkitan ketiga

Profit total saat mencapai simpul 4 adalah 1.050.000. Waktu total saat mencapai simpul 4 adalah 8. Dari simpul 4 (toko 2), penagih dapat pergi ke simpul markas yang selalu bisa dikunjungi dan toko 1 karena total waktu yang digunakan untuk mencapai toko 1 dari simpul 4 adalah 8 + 2 = 10, dan 10 berada pada restriksi toko 1 (4 - 8, 10 - 11) mengingat 10 berada di antara 10 dan 11.

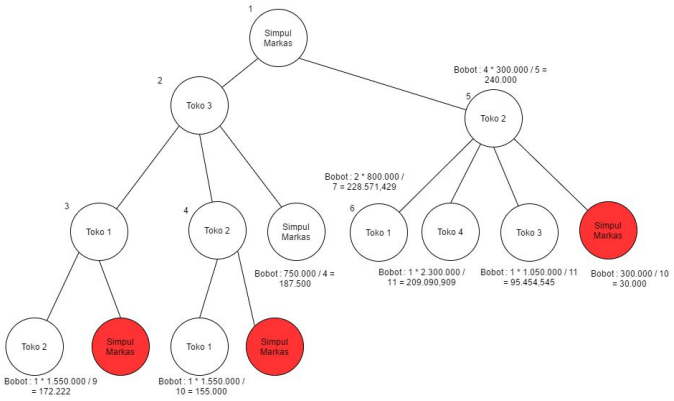
Selanjutnya dibangkitkan 2 simpul, yaitu simpul toko 1 dan simpul markas. Penagih akan mengecek toko 2 yang dibangkitkan dari simpul 1 karena bobot toko 2 tersebut lebih besar dari bobot toko yang lain, seperti pada gambar 7. Simpul tersebut mulai saat ini akan disebut simpul 5. Perhatikan simpul markas yang dibangkitkan dari simpul 4 diberikan warna merah pada gambar 7. Hal ini disebabkan bobot pada simpul markas adalah solusi yang dicari, dan hanya bobot terbesar yang akan dijadikan solusi permasalahan ini. Sehingga simpul markas yang tidak mungkin dijadikan solusi (yang bukan simpul markas berbobot terbesar) diberi tanda sebagai simpul mati.



Gambar 7 Pembangkitan keempat

Profit total saat mencapai simpul 5 adalah 300.000. Waktu total untuk mencapai simpul 5 adalah 5. Dari simpul 5 (toko 2), enagih dapat pergi ke simpul markas yang selalu bisa dikunjungi, toko 1 karena total waktu yang digunakan untuk mencapai toko 1 dari simpul 5 adalah 5 + 2 = 7, dan 7 berada pada restriksi toko 1 (4 - 8, 10 - 11) mengingat 7 berada di antara 4 dan 8, toko 3 karena total waktu yang digunakan untuk mencapai toko 3 dari simpul 5 adalah 5 + 6 = 11, dan 11 berada pada restriksi toko 3 (2 - 5, 6 - 8, 10 - 13) mengingat 11 berada di antara 10 dan 13, dan toko 4 karena total waktu yang digunakan untuk mencapai toko 4 dari simpul 5 adalah 5 + 6 = 11, dan 11 berada pada restriksi toko 4 (9 - 11) mengingat 11 berada di antara 9 dan 11.

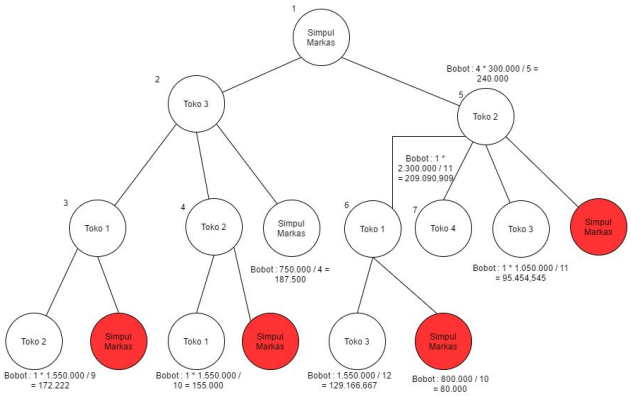
Selanjutnya dibangkitkan 4 simpul, yaitu simpul toko 1, toko 4, toko 3, dan simpul markas. Penagih akan mengecek toko 1 yang dibangkitkan oleh simpul 5 karena bobot toko 1 tersebut lebih besar dari bobot toko yang lain, seperti pada gambar 8. Simpul tersebut mulai saat ini akan disebut simpul 6. Perhatikan simpul markas yang dibangkitkan dari simpul 5 diberikan warna merah pada gambar 8. Hal ini disebabkan bobot pada simpul markas adalah solusi yang dicari, dan hanya bobot terbesar yang akan dijadikan solusi permasalahan ini. Sehingga simpul markas yang tidak mungkin dijadikan solusi (yang bukan simpul markas berbobot terbesar) diberi tanda sebagai simpul mati.



Gambar 8 Pembangkitan kelima

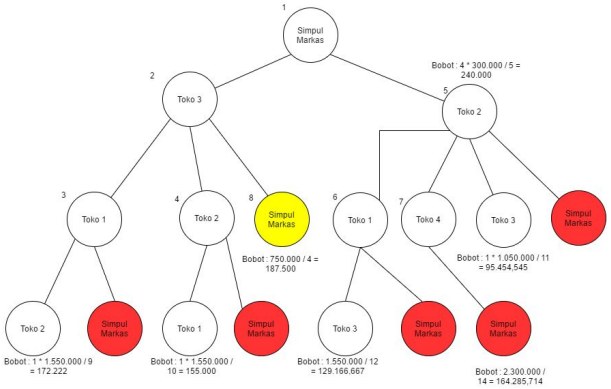
Profit total saat mencapai simpul 6 adalah 800.000. Waktu total saat mencapai simpul 6 adalah 7. Dari simpul 6 (toko 1), penagih dapat pergi ke simpul markas yang selalu bisa dikunjungi dan toko 3 karena total waktu yang digunakan untuk mencapai toko 3 dari simpul 6 adalah 7 + 5 = 12, dan 12 berada pada restriksi toko 3 (2 - 5, 6 - 8, 10 - 13) mengingat 12 berada di antara 10 dan 13.

Selanjutnya dibangkitkan 2 simpul, yaitu simpul toko 3 dan simpul markas. Penagih akan mengecek toko 4 yang dibangkitkan dari simpul 5 karena bobot toko 4 tersebut lebih besar dari bobot toko yang lain, seperti pada gambar 9. Simpul tersebut mulai saat ini akan disebut simpul 7. Perhatikan simpul markas yang dibangkitkan dari simpul 6 diberikan warna merah pada gambar 9. Hal ini disebabkan bobot pada simpul markas adalah solusi yang dicari, dan hanya bobot terbesar yang akan dijadikan solusi permasalahan ini. Sehingga simpul markas yang tidak mungkin dijadikan solusi (yang bukan simpul markas berbobot terbesar) diberi tanda sebagai simpul mati.



Gambar 9 Pembangkitan keenam

Profit total saat mencapai simpul 7 adalah 2.300.000. Waktu total saat mencapai simpul 7 adalah 11. Dari simpul 7 (toko 4) penagih hanya dapat pergi ke simpul markas karena toko lain tak bisa dikunjungi (waktu total jika mencapai toko lain sudah tidak berada pada restriksi toko tersebut.). Sehingga selanjutnya hanya akan dibangkitkan 1 simpul, yaitu simpul markas. Bobot pada simpul markas tersebut ternyata kurang dari bobot simpul markas yang dibangkitkan dari simpul 2, seperti pada gambar 10. Karena hanya bobot terbesar yang dijadikan solusi permasalahan ini, simpul markas yang dibangkitkan dari simpul 7 menjadi simpul mati (warna merah)



Gambar 7 Pembangkitan ketujuh dan terakhir

Selanjutnya simpul yang dicek adalah simpul markas yang dibangkitkan oleh simpul 2, karena simpul tersebut memiliki bobot terbesar saat ini. Simpul tersebut mulai saat ini disebut simpul 8. Simpul 8 merupakan simpul markas yang dicek pertama kali. Karena tidak ada simpul lain yang memiliki bobot lebih besar dari simpul 8, dan bobot simput hasil bangkitan memiliki tendensi menurun jika dibandingkan dengan simpul sebelumnya (yang membangkitkan), maka dapat ditetapkan simpul 8 sebagai solusi.

Sehingga, solusi terbaik untuk mendapatkan profit / waktu yang maksimal adalah dengan mendatangi toko 3 dan kembali ke markas. Total profit yang didapat mungkin tidak seberapa, tetapi waktu yang dibutuhkan hanya sedikit, sehingga jalur yang didapat dapat dikatakan sebagai jalur optimum.

IV. KESALAHAN YANG MUNGKIN TERJADI

Kebetulan pada kasus di gambar 3, bobot memiliki tendensi menurun antara simpul pembangkit dan simpul baru. Pada kasus toko 2 memiliki profit hanya 200.000, simpul toko 4 yang dibangkitkan dari simpul 5 akan memiliki bobot yang lebih besar dari simpul 5. Hal ini membuat simpul 5 tidak akan dicek karena memiliki nilai lebih kecil dari simpul 8, yang membuat simpul 7 yang memiliki nilai lebih besar dari simpul 8 tidak akan dicek. Dari sini dapat disimpulkan bahwa rumus yang penulis pikirkan mungkin masih dapat diperbaiki sehingga bobot memiliki tendensi menurun antara simpul pembangkit dan simpul baru.

V. KESIMPULAN

Algoritma Branch and Bound dapat digunakan dalam pencarian rute terbaik dalam masalah penagihan. Kompleksitasnya cukup besar karena banyaknya restriksi yang harus dicek setiap akan membangkitkan simpul. Tetapi, dengan menggunakan algoritma Branch and Bound, jumlah simpul yang dibangkitkan dapat dibuat menjadi lebih sedikit.

UCAPAN TERIMA KASIH

Pertama-tama saya mengucapkan syukur kepada Tuhan Yang Maha Esa yang telah melimpahkan berkatnya selama pengerjaan makalah ini sampai selesai. Selanjutnya ucapan terima kasih juga saya ucapkan pada orang tua saya yang memberi kesempatan saya membantu toko sehingga saya dapat memikirkan masalah ini. Saya juga mengucapkan terima kasih kepada bapak Rinaldi Munir dan ibu Nur Ulfa Maulidevi yang telah memberikan materi mata kuliah Strategi Algoritma yang membantu dalam proses penyelesaian makalah ini.

REFERENCES

[1] Slide Kuliah IF 2211 : Algoritma BFS dan DFS. Diakses pada 08/05/2016 pada pukul 20.13 WIB.

[2] Slide Kuliah IF 2211 : Algoritma Branch and Bound. Diakses pada 08/05/2016 pada pukul 21.32 WIB.