

Penentuan Beban Maksimum Truk Pickup dengan Algoritma Backtracking

Rafidika Samekto - 13519207
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
13519207@std.stei.itb.ac.id

Abstraksi—Pengiriman barang menjadi salah satu hal krusial di era modern ini. Membuat kendaraan pengirim barang seperti truk menjadi optimal adalah salah satu cara agar barang dapat sampai ke tangan konsumen dengan cepat. Penggunaan algoritma *backtracking* dalam penentuan barang mana yang akan dikirimkan dalam truk yang sama dapat menjadi alternatif yang cukup mangkus dalam menyelesaikan masalah ini. Dengan fungsi pembatas bahwa beban dan dimensi dari total barang yang masuk tidak boleh melebihi kapasitas truk, persoalan ini dapat diselesaikan dengan cepat.

Kata Kunci— *backtracking*; truk



Gambar 1. Truk Pickup [2]

I. PENDAHULUAN

Pada era modern ini, kebutuhan akan pengiriman barang yang cepat menjadi hal yang tak dapat lagi dipisahkan. Kecepatan melakukan pembelian melalui sederet aplikasi *online shop* membuat pengguna ingin barang yang dipesannya segera sampai. Tujuan tanpa terkecuali. Salah satu solusi yang ditawarkan adalah pengiriman banyak barang ke region atau wilayah yang sama. Hal ini dapat dicapai dengan menggunakan kendaraan-kendaraan berukuran besar seperti truk, kapal, atau pesawat terbang.

Untuk penggunaan jasa logistik melalui jalur darat, misalnya, akan dibutuhkan beberapa kendaraan besar yang dalam hal ini dapat memuat banyak barang seperti truk. Sebuah truk memiliki dimensi dan kapasitasnya masing-masing sehingga dalam persoalan ini masalah yang ingin diselesaikan adalah bagaimana sebuah truk dapat memaksimalkan "potensi" yang dimilikinya agar tidak ada atau hanya tersisa sedikit ruang yang kosong di dalam truk. Ini merupakan masalah optimasi dan pada makalah ini, penulis akan menyelesaikan persoalan ini menggunakan algoritma *Backtracking*.

Dalam dunia nyata, masalah seperti ini dapat dialami oleh pihak selain penyedia jasa logistik. Namun, dalam makalah ini, penulis mengasumsikan bahwa pihak yang "bermasalah" adalah penyedia jasa logistik yang seharusnya sudah memiliki sendiri kendaraan pengangkut yang dibutuhkan sehingga penulis tidak akan mempertimbangkan *cost* atau harga yang harus dikeluarkan oleh penyedia jasa.

II. DASAR TEORI

A. Algoritma Backtracking

Algoritma *backtracking* adalah salah satu metode yang mangkus untuk menyelesaikan masalah-masalah optimasi maupun non-optimasi. Algoritma ini juga dapat dipandang sebagai sebuah tahap dalam algoritma *Depth-First Search* (DFS) karena alur bekerjanya yang mirip dengan DFS. Jika dibandingkan dengan *exhaustive search*, algoritma ini berjalan lebih efisien karena ia hanya akan "menggali lebih dalam" pilihan yang mengarah ke solusi, tidak seperti *exhaustive search* yang menelusuri seluruh kemungkinan yang ada. Dalam *backtracking*, pilihan yang tidak mengarah ke solusi akan dipangkas sehingga dalam perjalanannya, tidak akan menelusuri pilihan-pilihan di bawah pilihan yang dipangkas tersebut.

Algoritma *backtracking* memiliki sejumlah properti umum, yaitu:

1. Solusi Persoalan

- Solusi dinyatakan sebagai vektor dengan n-tuple: $X = (x_1, x_2, \dots, x_n)$, $x_i \in S_i$
- Lazimnya $S_1 = S_2 = \dots = S_n$.
- Contohnya, $S_i = \{0, 1\}$, $x_i = 0$ atau 1.

2. Fungsi Pembangkit Nilai x_k

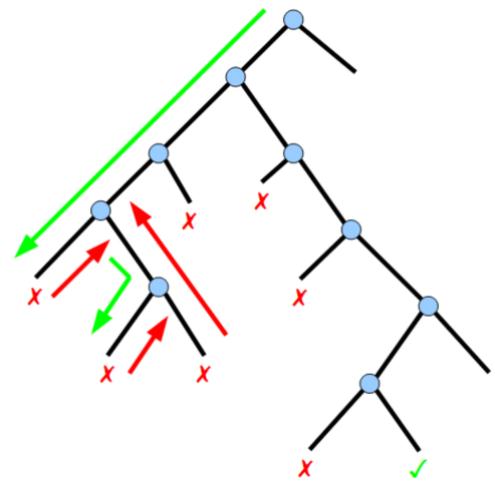
- Dinyatakan sebagai predikat $T()$

- $T(x[1], x[2], \dots, x[k-1])$ akan membangkitkan nilai untuk x_k yang merupakan komponen vektor solusi.

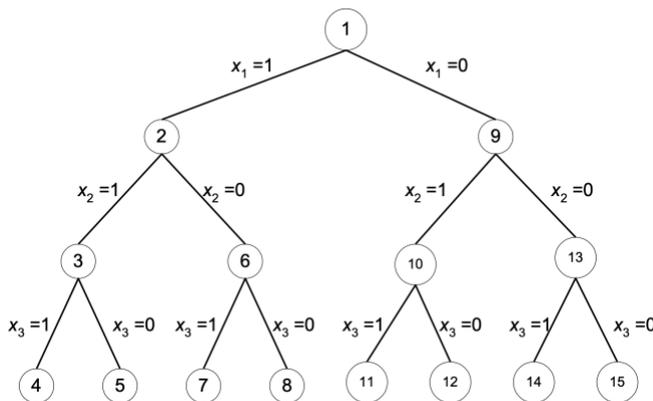
3. Fungsi Pembatas (*Bounding Function*)

Fungsi ini bekerja dengan mematikan pilihan yang tidak mengarah ke solusi. Setiap pilihan yang (akan) di-*expand* dilakukan evaluasi apakah mengarah ke solusi melalui fungsi pembatas ini. Jika tidak, pilihan tersebut akan dimatikan dan seluruh anaknya tidak akan dievaluasi.

Dalam algoritma ini, semua kemungkinan solusi masuk ke dalam sesuatu yang disebut ruang solusi. Ruang solusi diorganisasikan ke dalam suatu struktur pohon berakar. Simpul pada pohon tersebut menyatakan status persoalan sedangkan sisi dilabeli dengan nilai-nilai x_i . Seluruh lintasan dari akar ke daun dalam pohon ruang status menyatakan seluruh kemungkinan solusi dan termasuk ke dalam ruang solusi.



Gambar 3. Contoh alur pergerakan pemeriksaan simpul [1]



Gambar 2. Contoh pohon ruang status problem *knapsack* 1/0 [1]

Solusi dalam ruang solusi dibangkitkan dengan menggunakan aturan pembangkitan simpul pada *Depth-First Search* (DFS) sehingga menghasilkan lintasan dari akar ke daun. Pada algoritma ini, simpul yang sudah dibangkitkan dinamakan simpul hidup dan simpul yang sedang diperluas (dievaluasi) dinamakan simpul-E.

Dalam proses perluasan simpul-E, fungsi pembatas yang telah didefinisikan sebelumnya memiliki andil untuk "mengizinkan" simpul tersebut meluas atau tidak. Jika menurut fungsi pembatas simpul-E tidak akan mengarah ke solusi, simpul-E dimatikan dan tidak akan dianggap sebagai simpul hidup lagi. Hal ini juga berarti simpul-simpul anak yang seharusnya bisa dibangkitkan dari simpul-E tidak akan dilakukan evaluasi sehingga simpul yang harus diperiksa akan semakin berkurang.

Jika hasil dari evaluasi menyatakan bahwa simpul-E tidak berjalan ke arah solusi, pencarian solusi akan dilakukan *backtrack*, yaitu kembali satu simpul ke belakang dan menelusuri simpul lain yang menjadi anak dari simpul tersebut. Jika sudah tidak ada simpul anak yang bisa ditelusuri, dilakukan kembali *backtrack* ke simpul sebelumnya dan proses ini akan terus berjalan hingga ditemukan solusi yang diinginkan atau seluruh simpul sudah selesai diperiksa.

III. IMPLEMENTASI

Masalah yang akan diselesaikan seperti sudah disebutkan sebelumnya adalah bagaimana memasukkan barang-barang yang akan dikirim sehingga truk atau kendaraan pengangkut lainnya bermuatan se penuh mungkin. Penuh dalam hal ini yaitu seluruh ruang yang tersedia dalam truk terpakai dan tidak ada yang "terbuang" sia-sia.

Persoalan ini mengasumsikan muatan yang akan dimasukkan selalu berbentuk balok (kardus, box, dsb.). Karena itu setiap barang pasti memiliki dimensi, yaitu panjang, lebar, dan tinggi. Misalkan kita memiliki barang i , maka

- p_i adalah panjang barang i
- l_i adalah lebar barang i
- t_i adalah tinggi barang i

Sebuah barang juga memiliki berat. Berat barang i dilambangkan dengan w_i .

Di sisi lain, sebuah truk memiliki dimensi dan berat muatan maksimum yang dapat dibawanya. Seperti sebuah barang, dimensi dan berat dilambangkan dengan

- P , yaitu panjang truk
- L , yaitu lebar truk
- T , yaitu tinggi truk
- W , yaitu total berat muatan maksimum yang dapat dibawa truk

Urutan pemasukkan barang tidak diperhatikan. Dengan kata lain, persoalan ini tidak memedulikan apakah barang 1 dimasukkan terlebih dahulu kemudian barang 2 atau sebaliknya. Pohon ruang status yang akan dibentuk berupa pohon biner dengan simpul anak bagian kiri menyatakan barang dimasukkan dan simpul anak bagian kanan menyatakan barang tidak dimasukkan. Sisi pohon dilabeli dengan nilai x_i dimana nilai dari x_i adalah 1 atau 0 yang menyatakan barang dimasukkan atau tidak.

Karena sebuah truk memiliki kapasitas dan dimensi maksimum, maka kita tidak bisa begitu saja memasukkan seluruh barang ke dalam truk. Hal ini yang mendasari fungsi pembatas pertama, yaitu total berat barang yang dimasukkan ke dalam truk tidak boleh melebihi beban maksimum yang dapat dibawa truk atau dengan kata lain:

$$w = \sum_{i=1}^k w_i x_i \leq W$$

Hal lain yang perlu diperhatikan adalah dimensi barang yang tidak boleh melebihi dimensi truk sehingga fungsi pembatas lain yang berlaku adalah:

$$p = \sum_{i=1}^k p_i x_i \leq P$$

$$l = \sum_{i=1}^k l_i x_i \leq L$$

$$t = \sum_{i=1}^k t_i x_i \leq T$$

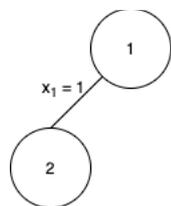
IV. STUDI KASUS

Misalkan terdapat sebuah truk akan digunakan untuk suatu kebutuhan logistik. Truk memiliki kapasitas beban maksimum sebesar 10000 kg dan dimensi 1200 cm x 1000 cm x 800 cm. Terdapat 3 buah barang yang akan dimasukkan ke dalam truk dengan spesifikasi berikut:

No Barang	p_i	l_i	t_i	w_i
1	600 cm	500 cm	400 cm	4000 kg
2	300 cm	250 cm	200 cm	2000 kg
3	900 cm	750 cm	600 cm	6000 kg

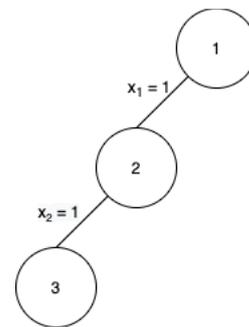
Tabel 1. Spesifikasi barang yang akan dimasukkan

Pencarian pertama dimulai dengan mengekspan simpul pertama untuk menentukan apakah barang 1 dimasukkan.



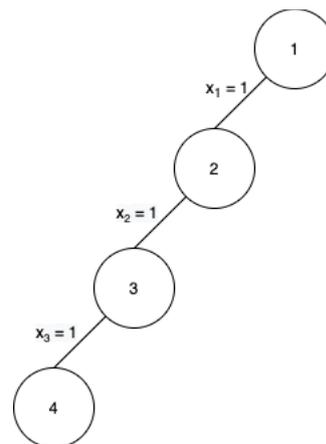
Gambar 4. Pohon ruang status awal

Simpul 2 pada gambar 4 menyatakan salah satu jalan menuju solusi, yaitu jika barang 1 dimasukkan. Barang 1, berdasarkan spesifikasi pada tabel 1, akan mengambil *space* sebesar 600 cm x 500 cm x 400 cm dan kapasitas maksimum truk berkurang dari 10000 kg menjadi 6000 kg. Menurut fungsi pembatas, simpul ini tidak melanggar *constraint* sehingga dibiarkan tetap hidup. Penelusuran berlanjut ke simpul 3, yaitu perluasan simpul 2.



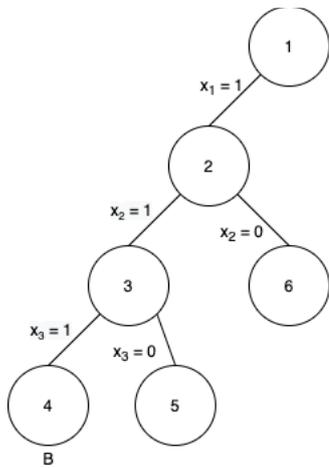
Gambar 5. Perluasan simpul 2

Simpul 3 melambangkan kondisi saat barang 1 dimasukkan dan barang 2 dimasukkan. Dari tabel 4, penjumlahan panjang, lebar, dan tinggi barang 1 dan 2 berturut-turut adalah 900 cm, 750 cm, 600 cm. Berat total yang ada di dalam truk jika keduanya dimasukkan adalah 6000 kg. Pengecekan oleh fungsi pembatas menyatakan simpul ini tidak melanggar *constraint* sehingga masih dibiarkan hidup hingga saat ini. Simpul 3 kemudian dapat di-*expand* ke simpul 4.



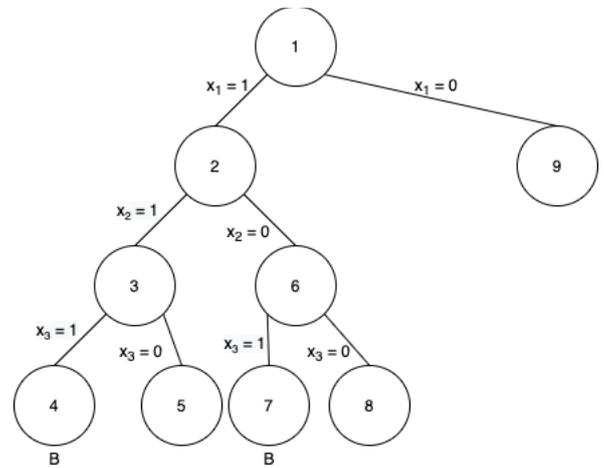
Gambar 6. Perluasan simpul 3

Simpul 4 yang juga merupakan simpul daun, menggambarkan situasi jika barang 1, 2, dan 3 dimasukkan ke dalam truk. Total panjang, lebar, dan tinggi dimensi barang yang dimasukkan secara berturut-turut menjadi 1800 cm, 1500 cm, dan 1200 cm. Berat total yang dimasukkan juga bertambah menjadi 12000 kg. Kedua kondisi ini melanggar *constraint* dan fungsi pembatas akan mematikan simpul ini. Perjalanan kembali dilanjutkan dengan *backtrack* ke simpul 3 lalu memperluas simpul tersebut menjadi simpul 5. Simpul 5 menyatakan saat dimana barang 1 dan 2 dimasukkan ke dalam truk tetapi barang 3 tidak. Hal ini akan menghasilkan perhitungan yang sama dengan yang terjadi pada simpul 3 sehingga fungsi pembatas tidak mematikan simpul ini. Karena sudah mencapai simpul daun, lintasan dari simpul akar hingga simpul 5 menjadi solusi terbaik untuk saat ini.



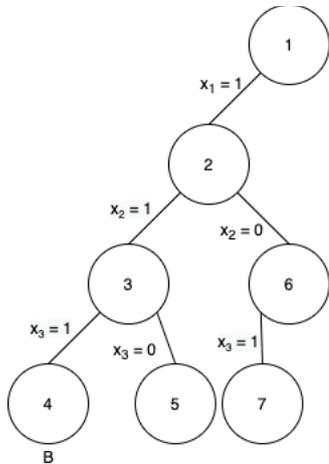
Gambar 7. Perluasan kedua simpul 2

Penelusuran selanjutnya dilakukan dengan *backtrack* hingga ke simpul 2 dan memperluas kemungkinan kedua simpul 2, yaitu simpul 6 yang menyatakan keadaan saat barang 1 dimasukkan namun barang 2 tidak. Berat barang yang dimasukkan dan dimensi truk yang digunakan sama dengan yang terjadi pada simpul 2 dan fungsi pembatas tidak akan mematikan simpul tersebut serta penelusuran dapat dilanjutkan melalui simpul 6.



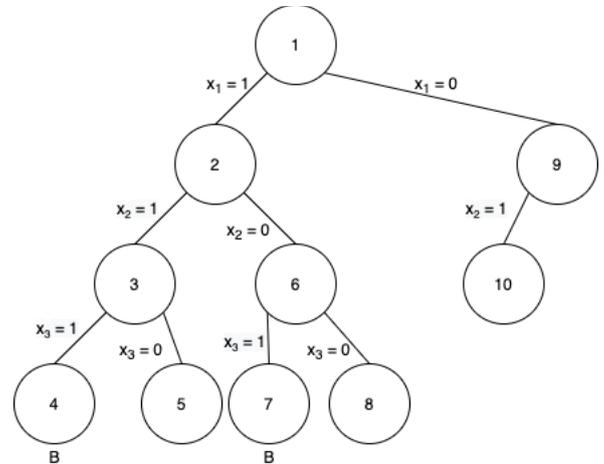
Gambar 9. Perluasan kedua simpul 1

Seluruh kemungkinan anak dari simpul 2 sudah selesai di-*expand*, namun dari simpul akar ada satu simpul lagi yang belum diperluas sehingga *backtrack* dilakukan hingga ke simpul 1 kemudian perluas menjadi simpul 9, barang 1 tidak diambil. Karena belum ada yang diambil, tidak ada beban tambahan ke dalam truk serta truk kosong sehingga fungsi pembatas memperbolehkan ekspansi simpul 9.



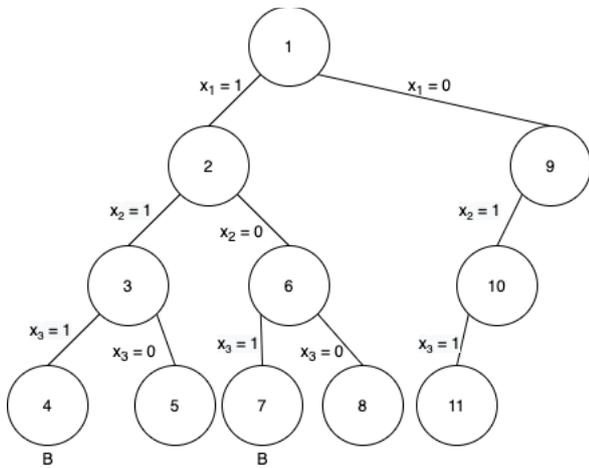
Gambar 8. Perluasan simpul 6

Simpul 7 menyatakan kondisi saat barang 1 dan 3 diambil namun barang 2 tidak. Jika dilakukan perhitungan beban total, akan didapat berat total yang dimasukkan 10000 kg. Sementara itu, panjang, lebar, dan tinggi total yang digunakan barang adalah 1500 cm, 1250 cm, 1000 cm. Ini melanggar *constraint* dan fungsi pembatas tidak menjadikan simpul 7 sebagai salah satu solusi. Kemudian, dilakukan *backtracking* ke simpul 6 dan ekspansi simpul tersebut sehingga didapat simpul 8 yang melambangkan saat dimana barang 1 dimasukkan namun barang 2 dan 3 tidak.



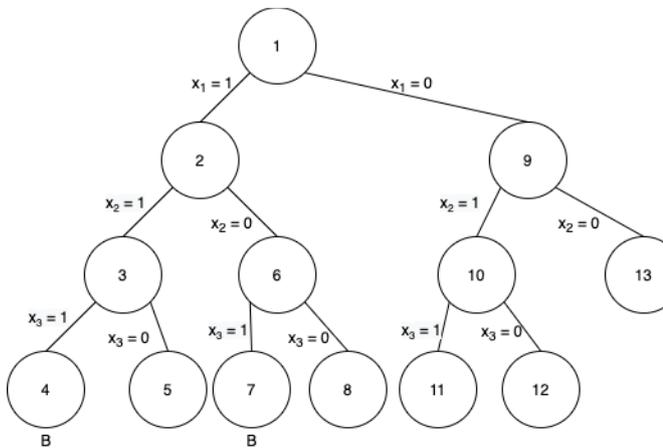
Gambar 10. Perluasan simpul 9

Ekspansi selanjutnya adalah simpul 10, yaitu saat barang 1 tidak diambil dan barang 2 diambil. Total berat muatan saat kondisi ini adalah 2000 kg sedangkan muatan memakan ruangan sebesar 300 cm x 250 cm x 200 cm. Kedua kondisi ini tidak melanggar *constraint* sehingga oleh fungsi pembatas diperbolehkan untuk mengekspansi dirinya sendiri.



Gambar 11. Perluasan simpul 11

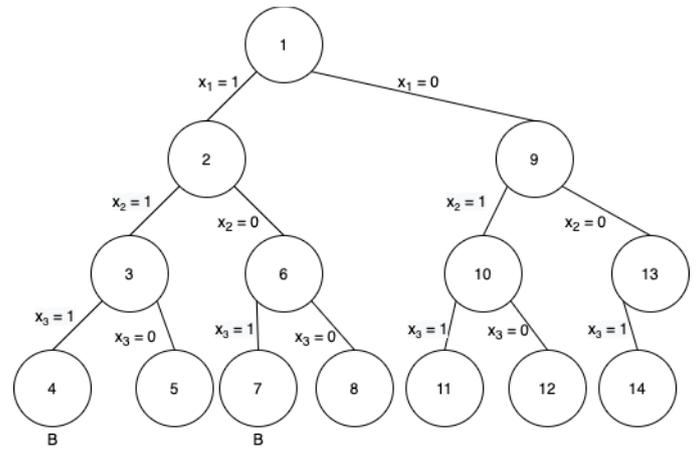
Gambar 11 menunjukkan simpul 11 yang menyatakan saat truk memuat barang 2 dan 3, namun tidak barang 1. Pada saat ini, berat muatan dalam truk menjadi 8000 kg dan total panjang, lebar, dan truk yang terpakai secara berturut-turut menjadi 1200 cm, 1000 cm, dan 800 cm. Dimensi yang terpakai mencapai batas maksimum dan karena itu, simpul masih diperbolehkan untuk "ada". Dibandingkan dengan solusi sebelumnya, yaitu simpul 5, solusi ini lebih baik karena barang yang dimasukkan mencakup seluruh isi truk. Oleh karena itu, simpul 11 menjadi solusi terbaik sementara.



Gambar 12. Perluasan kedua simpul 9

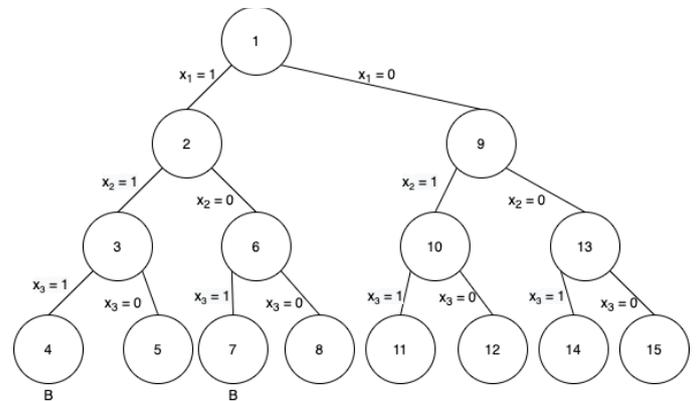
Kembali dilakukan backtracking ke simpul 10 untuk kemudian memperluas simpul tersebut lagi menjadi simpul 12, yaitu jika barang ketiga tidak diambil. Kondisi ini sama seperti simpul 10 sehingga fungsi pembatas tidak mematikan simpul ini. Namun, karena solusi yang ditawarkan tidak lebih baik ketimbang yang ditawarkan simpul 11, tidak ada yang menggantikan posisi simpul 11 sebagai solusi terbaik untuk sekarang.

Dilakukan lagi backtrack ke simpul 9 untuk memperluasnya menjadi simpul 13 yang menyatakan tidak ada dari barang 1 dan 2 yang diambil untuk dimasukkan ke truk. Perhitungan dapat dilanjutkan semenjak fungsi pembatas tidak menemukan hal yang melanggar constraint.



Gambar 13. Perluasan simpul 13

Hasil perluasan simpul 13, yaitu simpul 14, menyatakan saat dimana barang 1 dan 2 tidak diambil namun barang 3 dimasukkan. Dalam situasi ini, berat truk berkurang sebesar 6000 kg dan hal ini masih "aman". Sementara itu, ruang truk yang digunakan adalah sebesar barang 3, yaitu 900 cm x 750 cm x 600 cm. Pemakaian ruang dalam truk masih memenuhi constraint. Walaupun menjadi salah satu solusi, simpul 14 tidak juga menyingkirkan singgasana simpul 11 sebagai solusi optimum.



Gambar 14. Pohon ruang status lengkap

Backtrack dilakukan sekali lagi ke simpul 13. Seperti sebelumnya, simpul 13 di-expand menjadi simpul 15 yang menyatakan barang 1, 2, dan 3 tidak diletakkan dalam truk. Walaupun tetap memenuhi constraint, simpul 15 tidak dapat menggantikan simpul 11 dengan alasan yang sudah jelas.

Gambar 14 menunjukkan pohon ruang status lengkap dari permasalahan yang dibahas. Karena pencarian solusi sudah selesai, dapat disimpulkan bahwa solusi optimum dari masalah ini adalah simpul 11, yaitu barang 2 dan 3 dimasukkan ke dalam truk dan barang 1 tidak.

KESIMPULAN

Algoritma Backtracking merupakan salah satu algoritma yang dapat digunakan untuk menyelesaikan masalah optimasi maupun non-optimasi. Fleksibilitas algoritma membuat lebih banyak masalah dapat diselesaikan menggunakan algoritma ini.

Hal ini dapat terlihat dari studi kasus yang telah dijabarkan sebelumnya. Sebuah truk yang digunakan untuk mengangkut banyak barang dapat diatur sedemikian sehingga ruangan dalam kontainer truk dapat diisi semaksimal mungkin oleh barang yang akan ditempatkan di dalamnya.

PENUTUP

Akhir kata, penulis mengucapkan rasa syukur yang sedalam-dalamnya kepada Tuhan Yang Maha Esa yang telah memberikan kemudahan, kelancaran, serta inspirasi bagi penulis dalam menyelesaikan makalah ini. Tidak lupa juga penulis ucapkan terima kasih yang sebesar-besarnya kepada dosen mata kuliah Strategi Algoritma Bapak Rinaldi Munir yang telah memberi materi kuliah yang cukup lengkap dan dengan sangat baik sehingga penulis bisa menyelesaikan makalah ini. Penulis berharap makalah ini dapat digunakan oleh orang banyak untuk kemudian dipelajari dan dikembangkan.

REFERENCES

- [1] Rinaldi Munir, Algoritma Runut-Balik (Backtracking) [Online] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-backtracking-2021-Bagian1.pdf> (Diakses 9 Mei 2021)
- [2] Gambar Mobil dan Motor Keren [Online] <https://gambarotomotif.blogspot.com/2020/05/gambar-mobil-mitsubishi-box.html>

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 2021



Rafidika Samekto 13519207