

# Penerapan Algoritma Backtracking untuk Peletakan Barang pada Kapal Kontainer

Chandrika Azharyanti / 13518001  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
13518001@std.stei.itb.ac.id

**Abstract**— Stabilitas kapal menjadi hal yang krusial dalam keamanan saat berlayar. Oleh karena itu, tata letak cargo pada kapal yang mengangkut banyak barang-barang berat menjadi hal yang penting karena merupakan salah satu faktor internal yang mempengaruhi keseimbangan kapal. Makalah ini bertujuan untuk menjelaskan bagaimana algoritma *backtracking* dapat digunakan sebagai salah satu solusi untuk menyelesaikan permasalahan peletakan barang pada kapal kontainer.

**Kata kunci**—*backtracking*; tata letak; keseimbangan; berat

## I. PENDAHULUAN

Kapal Container adalah kapal kargo yang membawa semua dari mereka beban dalam kemasan antar moda truk-ukuran, dalam pengepakan teknik yang disebut. Mereka membentuk sarana umum antar moda transportasi barang komersial. Industri pelayaran peti kemas modern baru berusia 60 tahunan. Dalam usianya yang masih relatif singkat – dibanding sistem non peti kemas yang menguasai perdagangan global selama berabad-abad, jenis transportasi ini sudah begitu dominan.

Menurut data World Shipping Council, pelayaran sistem peti kemas menguasai lebih dari 60% nilai perdagangan global via laut. Hal itu disebabkan oleh bongkar muat yang dapat dilakukan hanya dalam beberapa jam dibandingkan dengan hari di sebuah kapal kargo tradisional. Ini, selain biaya tenaga kerja pemotongan, telah mengurangi pengiriman kali antara poin untuk sebagian besar, misalnya butuh beberapa minggu, bukan bulan untuk konsinyasi untuk dikirim dari India ke Eropa dan sebaliknya.



Gambar 1. Kapal container dengan muatan

Sumber : pinterest

Stabilitas kapal adalah kesetimbangan kapal pada saat diapungkan, tidak miring kekiri atau kekanan, demikian pula pada saat berlayar, pada saat kapal diolengkan oleh ombak atau angin, kapal dapat tegak kembali. Salah satu penyebab kecelakaan kapal di laut, baik yang terjadi di laut lepas maupun ketika di pelabuhan, adalah peranan dari para awak kapal yang tidak memperhatikan perhitungan stabilitas kapalnya sehingga dapat mengganggu kesetimbangan secara umum yang akibatnya dapat menyebabkan kecelakaan fatal seperti kapal tidak dapat dikendalikan, kehilangan kesetimbangan dan bahkan tenggelam

Ada beberapa perangkat yang digunakan untuk meningkatkan stabilitas kapal yaitu:

### a. Sirip lambung

Sirip lambung berfungsi untuk meningkatkan friksi melintang kapal sehingga lebih sulit untuk terbalik. Biasanya digunakan pada kapal dengan bentuk lambung V.

### b. Tangki penyeimbang

Merupakan tangki yang berfungsi menstabilkan posisi kapal dengan mengalirkan air balast dari kiri ke kanan kalau kapal miring kekiri dan sebaliknya kalau miring kekanan.

### c. Sirip stabilizer

Sirip stabiliser merupakan sirip di lunas kapal yang dapat menyesuaikan posisinya pada saat kapal oleng

Untuk keamanan selama pengiriman tentu saja stabilitas kapal menjadi hal yang sangat penting untuk diperhatikan. Secara umum hal-hal yang mempengaruhi keseimbangan kapal dapat dikelompokkan kedalam dua kelompok besar yaitu:

a. Faktor internal yaitu tata letak barang/cargo, bentuk ukuran kapal, kebocoran karena kandas atau tubrukan

b. Faktor eksternal yaitu berupa angin, ombak, arus dan badai

Seperti yang diketahui, tata letak barang merupakan salah satu pendukung kestabilan kapal. Oleh karena itu, makalah ini berfokus pada faktor internal dimana total berat pada bagian kiri dan bagian kanan harus memiliki selisih sesedikit mungkin untuk menghindari kemiringan kapal pada satu sisi. Walaupun kapal sudah memiliki perangkat penyeimbang, namun dengan lebih meratanya pembagian beban pada kapal akan

meningkatkan keamanan pelayaran. Berdasarkan hal tersebut, penulis akan menentukan peletakan barang pada kapal kontainer menggunakan pendekatan algoritma *backtracking* untuk persoalan 0/1 knapsack.

## II. LANDASAN TEORI

### A. Backtracking

*Backtracking* adalah salah satu metode pemecahan masalah yang mangkus, terstruktur, dan sistematis. *Backtracking* juga dikenal sebagai algoritma hasil pengembangan dari Depth First Search yang dilakukan pruning, dimana kandidat solusi yang tidak memenuhi tujuan akan dihapus dari solusi penyelesaian. Algoritma ini juga merupakan perbaikan dari exhaustive search, dimana pada exhaustive search semua kemungkinan solusi dieksplorasi satu per satu, berbeda dengan *backtracking* yang melakukan pemangkasan.

#### 1. Properti Umum *Backtracking*

##### a. Solusi persoalan

Solusi dinyatakan sebagai vektor dengan n- tuple:

$$X = (x_1, x_2, \dots, x_n), x_i \in S_i.$$

Dimana, mungkin saja  $S_1 = S_2 = \dots = S_n$ .

Contoh:  $S_i = \{0, 1\}, x_i = 0$  atau 1

##### b. Fungsi Pembangkit nilai $x_k$

Dinyatakan sebagai  $T(k)$

$T(k)$  berfungsi sebagai pembangkit nilai untuk  $x_k$ , yang merupakan komponen vektor solusi.

##### c. Fungsi pembatas

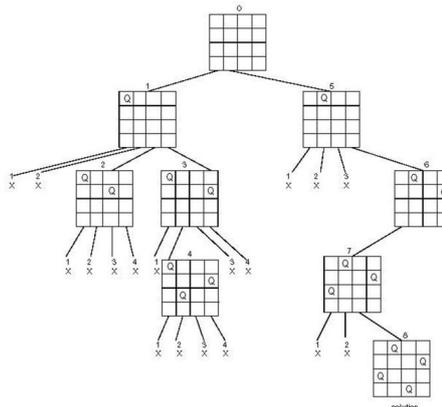
Fungsi ini berlaku membatasi / menentukan kandidat solusi mana yang sudah tidak cocok sehingga dapat dipangkas. Fungsi ini biasanya mengembalikan nilai Boolean, dimana hasil true berarti mengarah ke solusi dan jika false maka dibuang.

#### 2. Pengorganisasian Solusi

Semua kemungkinan solusi dari persoalan disebut ruang solusi. Ruang solusi ini dibentuk menjadi struktur pohon, dimana pada tiap simpulnya dinyatakan status persoalan dan cabang-cabangnya dilabeli dengan nilai  $x_i$

Lintasan dari akar ke daun menyatakan solusi yang mungkin, dan seluruh lintasan yang ada akan membentuk ruang solusi.

State-Space Tree of the 4-Queens Problem



Gambar 2. Contoh pohon ruang status untuk N-Queen problem

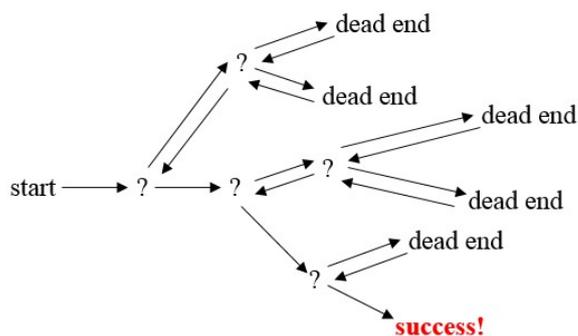
Sumber : <https://slideplayer.com/slide/14740688/>

#### 3. Prinsip Pencarian Solusi dengan *Backtracking*

Solusi dicari dengan cara membangun lintasan dari akar ke daun, pembangunan dilakukan mengikuti aturan depth-first order, dimana dilakukan perluasan kedalam terlebih dahulu.

Terdapat 3 jenis simpul, yaitu simpul hidup (simpul yang sudah dilahirkan), simpul-E (simpul hidup yang sedang diperluas), dan simpul mati (simpul-E yang tidak mengarah ke solusi). Fungsi yang digunakan untuk membunuh simpul-E adalah fungsi pembatas, jika simpul sudah mati maka simpul tersebut tidak akan diperluas lagi.

Jika saat pembentukan lintasan, diakhiri dengan simpul mati maka akan dilakukan backtrack kesimpul sebelumnya hingga menemukan simpul yang belum di expand, lalu simpul tersebut menjadi simpul-E yang baru. Pencarian dihentikan jika telah ditemukan solusi / goal dicapai,



Gambar 3. Skema *Backtracking*

Sumber : <https://slideplayer.info/slide/2775238/>

## B. Konsep 0/1 Knapsack

Seperti artinya, knapsack merupakan karung atau kantung yang digunakan untuk memuat sesuatu, dan tentunya tidak semua objek dapat dtampung di dalam karung. Karung memiliki kapasitas yang biasanya bertolak ukur berat (weight).

Persoalan 0/1 knapsack adalah persoalan memilih objek-objek yang dimasukkan ke dalam karung sehingga keuntungan yang didapatkan maksimum dengan syarat total berat objek yang dimasukkan kedalam knapsack tidak boleh melebihi kapasitas knapsack.

Solusi persoalan dinyatakan sebagai vektor n-tupel:

$$X = \{x_1, x_2, \dots, x_n\}$$

$x_i = 1$  jika objek ke- $i$  dimasukkan ke dalam knapsack,

$x_i = 0$  jika objek ke- $i$  tidak dimasukkan.

Fungsi pembatas pada persoalan ini adalah total jumlah objek yang akan dimasukkan pada suatu state tidak melebihi kapasitas knapsack dikurangi dengan total berat objek yang sudah dimasukkan pada state sebelumnya. Jika sudah melebihi maka simpul akan dimatikan.

Untuk mendapatkan himpunan solusi dengan profit paling besar, maka untuk setiap perluasan simpul akan dituliskan jumlah profit yang didapat sejauh ini. Jika pada saat membangun lintasan berakhir pada simpul mati maka kita akan mengecek apakah profit yang didapat lebih besar dibanding profit yang ada sebelumnya, jika iya maka profit tertinggi akan diubah dan simpul daun yang memiliki profit lebih kecil dibanding profit tertinggi sekarang akan dimatikan.

## III. METODE PENYELESAIAN MASALAH

Berikut strategi penyelesaian untuk permasalahan pemilihan tata letak barang menggunakan algoritma *backtracking* serta dengan pendekatan persoalan 0/1 knapsack:

### A. Strategi 0/1 knapsack

Seperti yang diketahui, bahwa tujuan dalam memecahkan permasalahan ini adalah dengan cara membari beban-beban cargo menjadi dua partisi, dimana selisih antara dua partisi tersebut haruslah minimum.

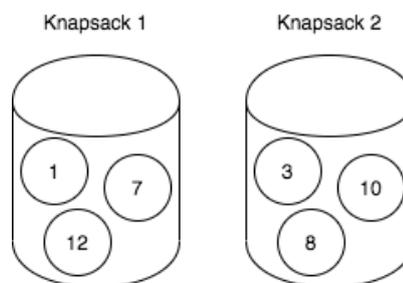
Tujuan beban-beban dibagi menjadi dua partisi adalah untuk menyeimbangi sisi bagian kanan dan kiri kapal, sehingga ide awal penyelesaiannya adalah dengan cara melakukan pendekatan dengan persoalan 0/1 knapsack. Anggap salah satu sisi kapal sebagai knapsack, dimana maksimal total beban pada sisi tersebut adalah setengah dari total beban yang akan dimasukkan kedalam kapal.

Hal itu didasari oleh ide bahwa dua partisi akan memiliki selisih minimum jika total berat keduanya merupakan / mendekati setengah total berat, namun kita tidak perlu meninjau kedua sisi, karena pasti sisi satunya lagi akan berisi barang-barang yang tidak ditaruh pada knapsack.

Keuntungan yang di cek pada persoalan knapsack ini bukanlah profit yang biasa terdapat pada persoalan-persoalan knapsack biasa, disini keuntungan yang diperhitungkan merupakan minimum selisih berat antara dua sisi kapal, perhitungan selisih dilakukan selama pembuatan ruang solusi

persoalan. Lintasan solusi (dari simpul akar hingga daun) yang ada akan dibandingkan dengan minimum selisih berat yang ada sekarang, jika lebih besar maka lintasan tersebut akan dihapus dari set solusi persoalan.

Berikut contoh pembagian knapsack untuk beban = {3,8,7,10,1,12}. Dimana 0/1 knapsack dilakukan pada knapsack 1 dengan total beban < setengah total beban yang ada (di contoh ini adalah  $20.5 = 20$ ), dan knapsack 2 mengambil beban-beban yang tidak dimasukkan kedalam knapsack 1.



Gambar 4. Contoh hasil akhir knapsack

### B. Strategi *Backtracking*

Dikarenakan disini dilakukan penyelesaian permasalahan 0/1 menggunakan *backtracking*, maka hasil solusi persoalan merupakan tupel yang berisikan 0/1 dimana 1 berarti objek dimasukkan ke knapsack dan 0 tidak.

Fungsi pembatas disini adalah total beban pada knapsack tidak melebihi batas (setengah total beban yang akan dimasukkan ke kapal), jika beban sudah melebihi maka simpul tidak akan diperluas lagi / menjadi simpul mati. Lintasan yang menuju simpul sebelumnya (lintasan yang valid adalah lintasan yg belum melanggar fungsi pembatas) akan menjadi salah satu kandidat solusi, dimana setiap ditemukan simpul mati akan di cek apakah selisih beratnya (lower bound), jika lebih besar maka akan dibuang dari list kandidat dan jika lebih kecil maka simpul mati yang baru ditemukan tidak akan dimasukkan ke list kandidat solusi.

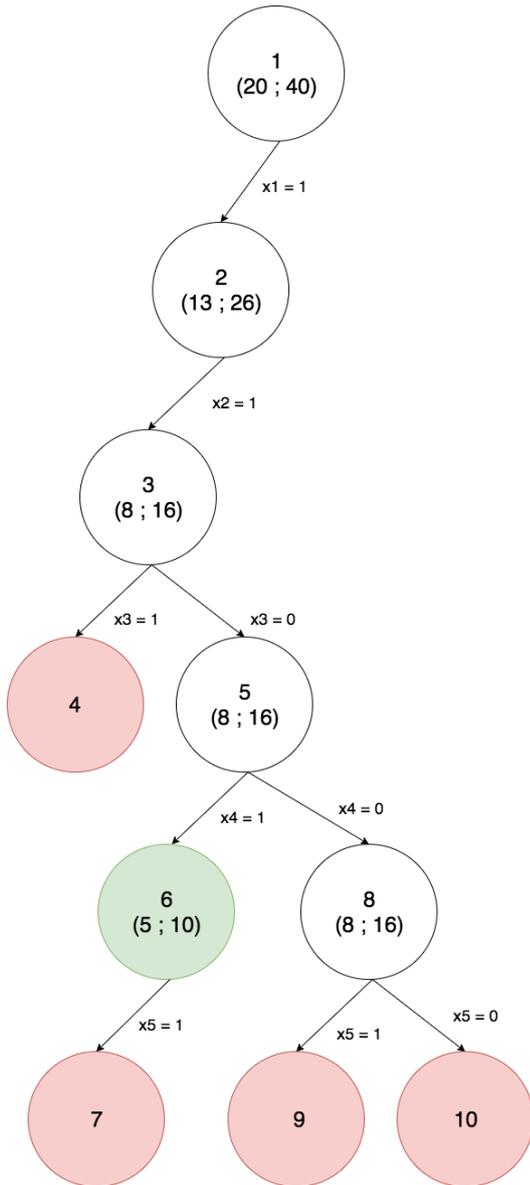
## IV. STUDI KASUS

Pada bab ini akan dibahas mengenai langkah-langkah pengaplikasian algoritma *backtracking* dalam menentukan tata letak barang yang akan dimasukkan kedalam kapal kontainer. Pengaplikasian algoritma menggunakan contoh set beban = {7, 5, 11, 3, 14}

### A. Perhitungan Fungsi Pembatas

Total Berat seluruh beban adalah 40, maka maksimal beban pada knapsack 1 adalah 20. Lower bound di set menjadi 40 (dengan anggapan seluruh beban berada pada satu sisi), jika ditemukan simpul mati bandingkan dengan lower bound, ubah lower bound jika selisih pada simpul tersebut lebih kecil dibanding lower bound. Maka fungsi penyelesaiannya adalah :  $\sum w_i * x_i \leq 20$  (i menyatakan index barang)

B. Tahap *Backtracking*



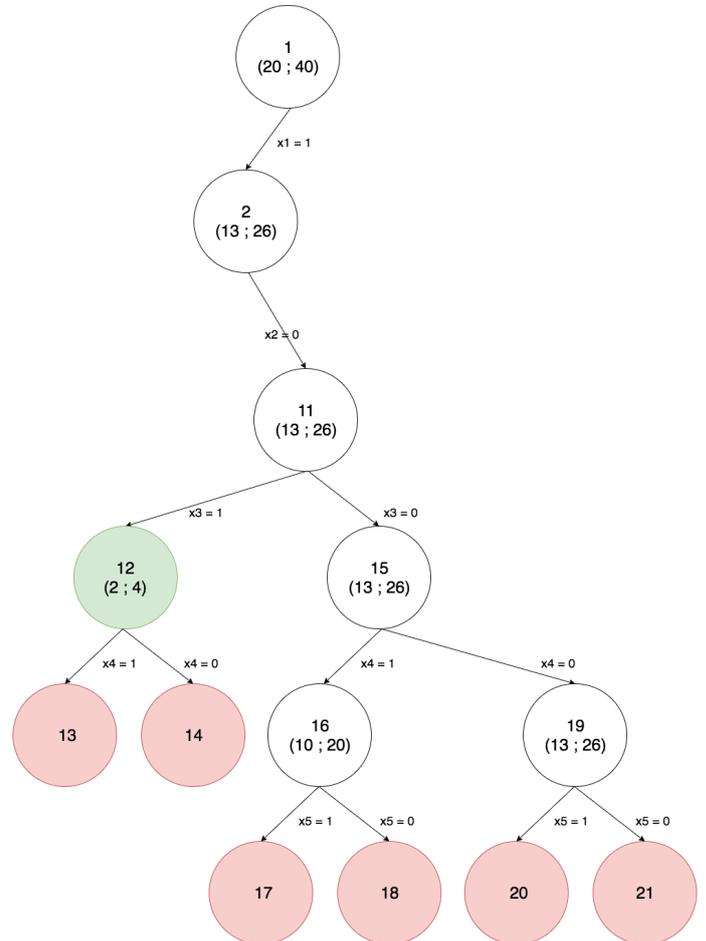
Gambar 5. Tahap 1 *backtracking*

Dapat dilihat dari sebagian hasil *backtracking* diatas, angka pada bagian atas simpul menyatakan urutan pembangkitan simpul, dua angka dibawahnya menyatakan (sisa kapasitas knapsack ; selisih beban antara 2 knapsack).

Warna pada simpul menyatakan bahwa simpul merupakan simpul daun / simpul mati dimana hal tersebut terjadi bila pada state tersebut dilakukan pemasukan beban yang menyebabkan kelebihan kapasitas.

Saat simpul 4 ditemukan, lower bound berubah menjadi 16 dan simpul 3 masuk kedalam kandidat solusi. Lalu dilakukan *backtracking* dan simpul 5 diperluas, namun saat simpul 7 ditemukan, simpul 6 tersebut memiliki selisih yang lebih kecil ( $10 < 16$ ) maka lower bound diubah menjadi 10 dan simpul 3 dikeluarkan dari kandidat solusi, digantikan oleh simpul 6. Dilanjutkan dengan *backtracking* ke simpul 2.

Dilakukan *backtracking* kembali dan simpul 8 diperluas. Saat simpul 9\* dan 10\* ditemukan, simpul 8 memiliki selisih yang lebih besar dari lower bound, maka kedua simpul tersebut tidak dimasukkan kedalam himpunan kandidat solusi. Se jauh ini simpul 6 merupakan kandidat solusi terbaik (ditandai dengan warna hijau).



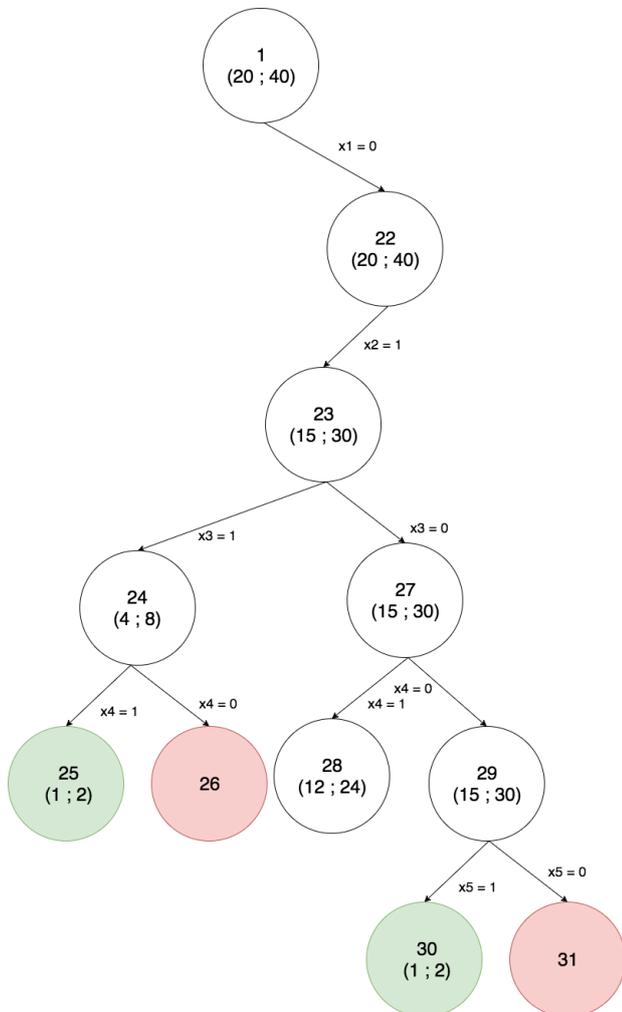
Gambar 6. Tahap 2 *backtracking*

Pada tahap ini, ditemukan simpul 13\* dan 14\* dimana selisihnya lebih kecil dibanding lower bound ( $4 < 10$ ) maka lower bound diubah dan simpul 6 dikeluarkan dari kandidat solusi digantikan oleh simpul 12. Dilanjutkan dengan *backtracking* ke simpul 11 dan simpul 15 diperluas. Setelah itu ditemukan kembali simpul mati lainnya seperti 17\* dan 18\*, namun selisih mereka lebih besar dari lower bound maka tidak dijadikan kandidat solusi. Begitupun juga dengan simpul 20\* dan 21\*. Dilanjutkan dengan backtrack ke simpul 1.

Perhatikan gambar 7, dimana ditemukan simpul 25 yang memiliki selisih lebih kecil dibanding lower bound, maka simpul 12 dikeluarkan dari kandidat solusi dan digantikan oleh simpul 25 (disini yang merupakan kandidat solusi bukanlah simpul sebelumnya dikarenakan simpul 25 tidak melanggar fungsi pembatas dan index barang merupakan yang terakhir). Setelah itu dilakukan backtrack ke simpul 24 dan membangun simpul 26, dimana simpul 26 bukan merupakan kandidat solusi.

- \*simpul dibangun hanya untuk menunjukkan bahwa jika suatu simpul memiliki kapasitas < beban selanjutnya dan beban selanjutnya adalah beban terakhir, tidak perlu lagi diperluas

Simpul 28 tidak perlu di ekspan karena sisa kapasitasnya lebih kecil dibanding beban selanjutnya dan beban selanjutnya merupakan beban terakhir namun karena selisih yang dimiliki simpul tersebut lebih besar dari lower bound, maka simpul tidak masuk kedalam kandidat solusi. Dilanjutkan dengan backtrack ke simpul 27, dan ditemukan simpul 30 yang memiliki selisih sama dengan lower bound, oleh karena itu simpul tersebut dimasukkan juga kedalam kandidat solusi bersama simpul 25, sehingga saat ini terdapat dua kandidat solusi yaitu simpul 25 dan simpul 30. Dilanjutkan dengan backtrack ke simpul 22.



Gambar 7. Tahap 3 *backtracking*

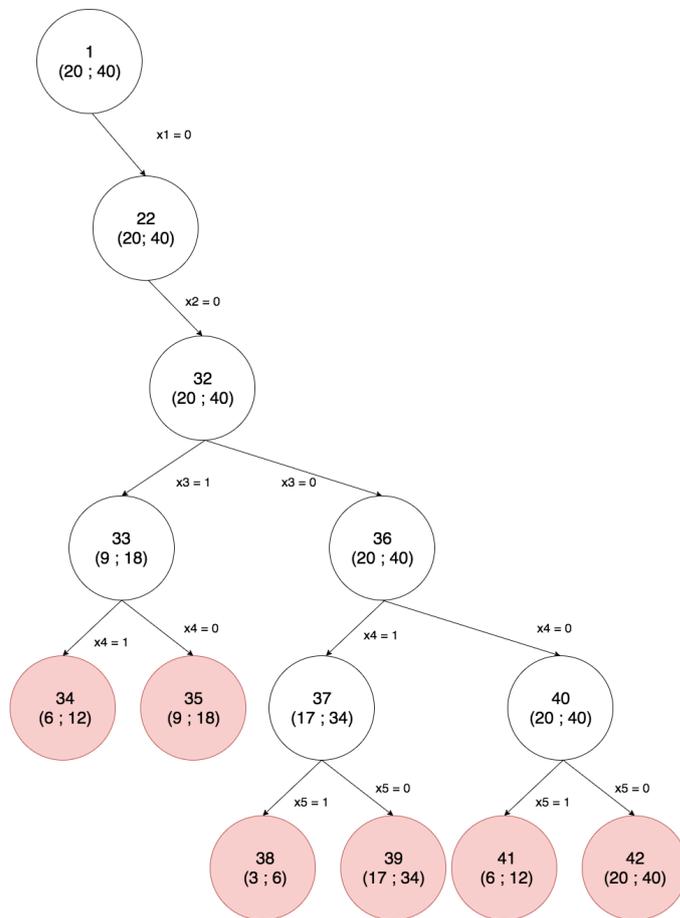
Perhatikan Gambar 8, ditemukan simpul 34, namun selisihnya masih lebih besar dibanding lower bound, maka simpul tidak dimasukkan kedalam kandidat solusi, ditemukan juga simpul 38, 39, 41, dan 42 namun selisih pada semua simpul tersebut masih lebih besar dibanding lower bound sehingga tidak ada yang dimasukkan kedalam kandidat solusi.

Setelah seluruh simpul dibangun, untuk setiap simpul daun yang merupakan kandidat solusi kita lakukan penelusuran sepanjang lintasannya, dan didapatkan 2 tupel jawaban, yaitu  $\{0,1,1,1,0\}$  dan  $\{0,1,0,0,1\}$ .

Dari studi kasus ini, hasil *backtracking* menunjukkan bahwa terdapat dua solusi untuk tata letak barang pada kapal container ini. Solusi yang pertama adalah, satu sisi memiliki beban 19

(dengan rincian beban =  $\{5,11,3\}$ ) dan disisi lainnya memiliki beban 21 (dengan rincian beban =  $\{7, 14\}$ ).

Solusi yang kedua adalah satu sisi memiliki beban 19 (dengan rincian beban =  $\{5, 14\}$ ) dan sisi lainnya memiliki beban 21 dengan rincian beban =  $\{7, 11, 13\}$ ). Kedua solusi ini menyediakan selisih beban terkecil antara bagian kiri dan kanan kapal, dengan nilai 2.



Gambar 8. Tahap 4 *backtracking*

## V. KESIMPULAN

Banyak hal yang dapat diterapkan pada kehidupan sehari-hari dari pembelajaran strategi algoritma. Salah satunya adalah pengaplikasian algoritma *backtracking*. Algoritma *backtracking* dapat digunakan sebagai salah satu *tools* untuk menyelesaikan permasalahan dalam penentuan tata letak barang pada kapal container. Terinspirasi dari perdagangan di dunia yang sebagian besar melakukan pengiriman menggunakan kapal container ini, penulis membuat makalah berisi penggunaan algoritma *backtracking* untuk menentukan peletakan barang yang paling baik sehingga selisih berat pada kiri dan kanan kapal minimal sehingga kemungkinan terjadi ketidakseimbangan kapal dapat diminimalisir dan keamanan barang selama pengiriman dapat ditingkatkan.

Pada makalah ini penulis hanya mempertimbangkan 1 hal untuk dijadikan kondisi penentuan. Selain berat beban barang, masih terdapat banyak hal-hal yang perlu dipertimbangkan pada persoalan ini, perlu diperhatikan pula tinggi tumpukan cargo pada setiap sisi serta faktor lainnya yang bersifat eksternal seperti kuat angin dan besar gelombang yang dapat mengganggu keseimbangan kapal. Walaupun kompleksitas waktu algoritma ini cenderung lama, namun diharapkan metode penyelesaian dengan pendekatan *backtracking* dan 0/1 knapsack ini dapat menjadi gambaran cara untuk penentuan tata letak cargo.

## VI. PENUTUP

Segala puji bagi Allah SWT yang telah memberikan penulis kemudahan sehingga dapat menyelesaikan makalah ini dengan tepat waktu. Penulis juga mengucapkan terima kasih kepada kedua orang tua serta teman-teman yang memberi dukungan secara moral maupun doa selama pengerjaan makalah ini. Tidak lupa juga penulis mengucapkan terima kasih kepada Ibu Dr. Masayu Leylia Khodra selaku dosen mata kuliah Strategi Algoritma yang telah memberi materi untuk penulisan makalah. Akhir kata, penulis menyadari masih terdapat kekurangan dan kesalahan kata dalam makalah ini, penulis berharap makalah ini dapat digunakan sebaik-baiknya dan dikembangkan sehingga lebih menghasilkan manfaat bagi masyarakat luas

## REFERENSI

- [1] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/Algoritma-Runut-balik-\(2020\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2019-2020/Algoritma-Runut-balik-(2020).pdf). Diakses pada 1 Mei 2020
- [2] <http://hugonano.blogspot.com/2015/05/dasar-dasar-stabilitas-kapal.html>. Diakses pada 1 Mei 2020
- [3] <https://www.kaskus.co.id/thread/54c5e6c31cbfaa3d228b456b/sejarah-kapal-kontainer/>. Diakses pada 2 Mei 2020

## LINK YOUTUBE

<https://youtu.be/deETdsILu0w>

## 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 Mei 2020



Chandrika Azharyanti - 13518001