

# Penerapan Algoritma Backtracking untuk Peletakan Barang di Gudang pada SCM

Karlsen Adiyasa Bachtiar 13519001  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
E-mail: karlsenab7@gmail.com

**Abstract**—Pada era Revolusi Industri ini, salah satu aspek penting dalam menjalankan sebuah bisnis adalah manajemen pendistribusian supply dari produsen ke distributor dan akhirnya ke customer. Oleh karena itu munculah Supply Chain Management (SCM) yang bertujuan untuk mengatur alur pendistribusian yang baik dari produsen hingga ke konsumen sehingga produk dapat dikirim dengan cepat dan dengan biaya yang lebih murah.

**Keywords**—Supply Chain Management, SCM, distribusi, gudang, warehouse, backtrack, knapsack

## I. PENDAHULUAN

Kata “Supply Chain Management”, disingkat SCM, pertama kali ditemukan oleh Keith Oliver pada tahun 1982 yang memiliki makna sangat penting pada dunia bisnis. Supply Chain Management adalah rangkaian kegiatan yang diperlukan untuk merencanakan, mengendalikan, dan menjalankan arus produk. Kegiatan ini meliputi proses perolehan bahan baku, proses produksi, hingga distribusi produk ke konsumen akhir dengan cara yang se-efisien dan se-hemat mungkin.

SCM yang baik dapat membantu perusahaan dalam menyesuaikan kebutuhan perusahaan dengan teknologi yang ada saat ini. Banyak perusahaan-perusahaan besar yang sudah memakai outsource organisasi yang lebih berpengalaman untuk menerapkan supply chain dalam perusahaannya, karena dapat menerapkan teknologi yang lebih baik sehingga akan lebih menghemat biaya. Tujuan dari SCM adalah untuk meningkatkan tingkat kredibilitas dan kerja sama antar mitra supply chain dan untuk mengintegrasikan proses antar system sehingga dapat meningkatkan kemudahan kerjasama kepada mitra *supply chain*. SCM sangatlah luas bagian-bagiannya sehingga pada makalah ini hanya akan dibahas bagaimana untuk mengoptimasi cara penempatan barang di gudang saja.



Gambar 1. Supply Chain Management

Sumber: <https://www.shsu.edu/programs/bachelor-ofbusiness-administration-in-supply-chain-management/>

(diakses pada 10 Mei 2021, 18.03 WIB)

## II. TEORI DASAR

### A. Backtracking

Backtracking adalah sebuah metode pemecahan masalah yang mangkus. Algoritma backtracking atau runut-balik dapat dipandang sebagai salah satu dari dua hal berikut:

1. Backtracking sebagai sebuah fase di dalam algoritma traversal DFS.
2. Backtracking atau runut-balik sebagai sebuah metode pemecahan masalah yang mangkus, terstruktur, dan sistematis, baik untuk persoalan optimasi maupun non-optimasi

Algoritma runut-balik merupakan perbaikan dari *exhaustive search*. Pada *exhaustive search*, semua kemungkinan solusi dieksplorasi dan dievaluasi satu per satu. Sedangkan pada algoritma *backtracking* hanya pilihan yang mengarah ke solusi yang dieksplorasi, pilihan yang tidak mengarah ke solusi tidak dipertimbangkan lagi sehingga dapat memangkas (pruning) simpul-simpul yang tidak mengarah ke solusi. Algoritma runut-balik pertama kali diperkenalkan oleh D. H. Lehmer pada

tahun 1950. Kemudian R.J Walker, Golomb, dan Baumert menyajikan uraian umum tentang algoritma runut-balik.

Properti umum algoritma runut-balik:

1. Solusi persoalan

Solusi dinyatakan sebagai vektor dengan n- tuple:

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

Dimana, umumnya  $S_1 = S_2 = \dots = S_n$ .

Contoh: Pada persoalan 1/0 knapsack  $S_i = \{0, 1\}$ ,  $x_i = 0$  atau  $1$

2. Fungsi Pembangkit nilai  $x_k$

Dinyatakan sebagai predikat  $T()$

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

3. Fungsi Pembatas (*bounding function*)

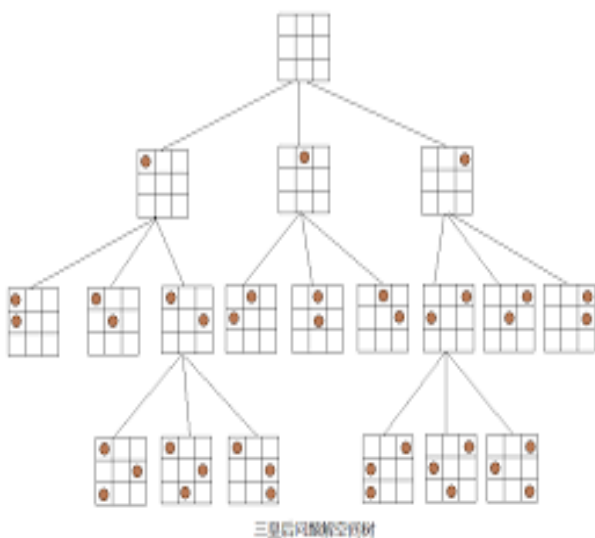
Dinyatakan sebagai predikat  $B(x_1, x_2, \dots, x_k)$

$B$  bernilai *true* jika  $(x_1, x_2, \dots, x_k)$  mengarah ke solusi. Mengarah ke solusi artinya tidak melanggar kendala (*constraints*).

Fungsi ini berlaku membatasi / menentukan kandidat solusi mana yang sudah tidak cocok sehingga dapat dipangkas. Fungsi ini biasanya mengembalikan nilai Boolean, dimana jika *true*, maka pembangkitan nilai untuk  $x_{k+1}$  dilanjutkan, tetapi jika *false*, maka  $(x_1, x_2, \dots, x_k)$  dibuang.

Semua kemungkinan solusi dari persoalan disebut ruang solusi. Ruang solusi ini diorganisasikan ke dalam struktur pohon berakar, dimana pada tiap simpulnya menyatakan status (state) 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. Pengorganisasian pohon ruang solusi diacu sebagai pohon ruang status (*status space tree*).



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

Sumber:

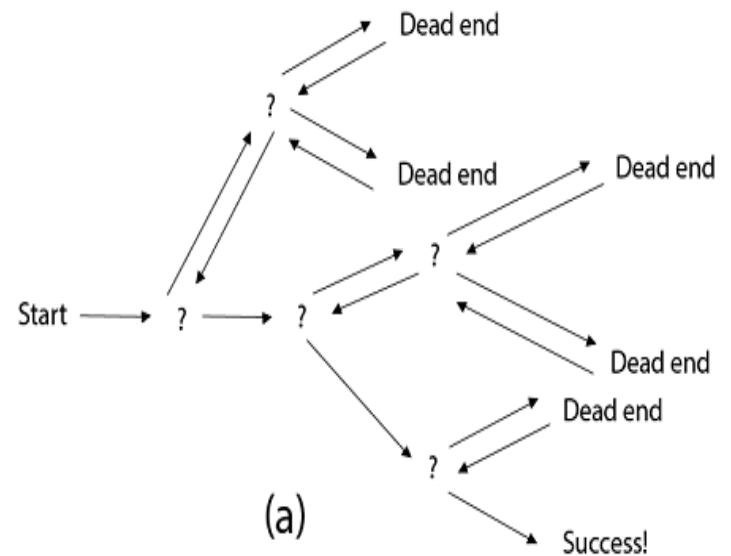
<https://www.programmersought.com/article/78674658709/>

(diakses pada 10 Mei 2021, 19.06 WIB)

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 / expand), dan simpul mati (simpul-E yang tidak mengarah ke solusi). Fungsi yang digunakan untuk membunuh simpul-E adalah fungsi pembatas atau bounding function, jika simpul sudah mati, maka simpul tersebut tidak akan diperluas lagi.

Jika saat pembentukan lintasan, diakhiri dengan simpul mati maka akan dilakukan backtrack ke simpul sebelumnya yang sudah pernah dilewati hingga menemukan simpul yang masih dapat dilakukan expand, lalu simpul tersebut menjadi simpul-E (simpul yang akan di-expand) yang baru. Pencarian akan dihentikan jika telah ditemukan solusi / goal sudah tercapai.



Gambar 3. Alur Backtracking

Sumber: <https://www.javatpoint.com/backtracking-introduction>

(Diakses pada 10 Mei 2021, 19.14 WIB)

B. 0/1 Knapsack

Knapsack merupakan karung atau kantung yang digunakan untuk menyimpan barang dan setiap karung memiliki kapasitas maksimumnya masing-masing sehingga kita tidak dapat memasukkan barang yang melebihi kapasitas maksimum dari karung yang akan diisi.

Persoalan 0/1 knapsack adalah persoalan memilih objek-objek (setiap objek memiliki keuntungan/profit masing-masing) yang dimasukkan ke dalam karung sehingga keuntungan yang didapatkan maksimum dengan syarat total berat (weight) objek yang dimasukkan ke dalam knapsack tidak boleh melebihi kapasitas knapsack.

Solusi persoalan dinyatakan sebagai vektor n-tuple:

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

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

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

Fungsi pembatas pada persoalan 0/1 knapsack 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.

Fungsi konstrain (*bounding function*):

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

Gambar 4. Bounding Function

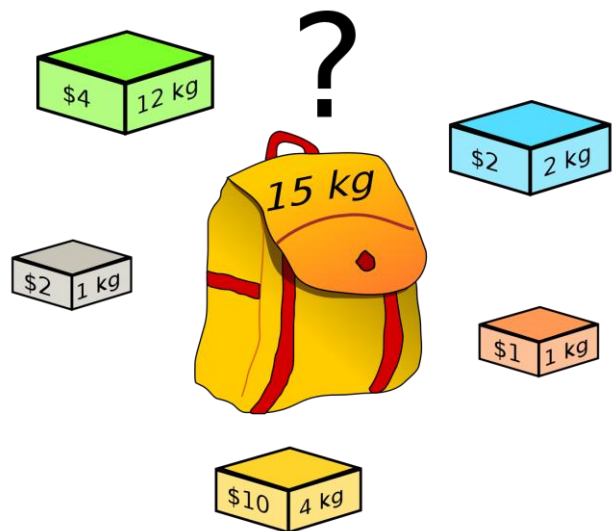
Sumber:

<https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-backtracking-2021-Bagian1.pdf>

(Diakses pada 10 Mei 2021, 21.52 WIB)

Untuk mendapatkan himpunan solusi dengan profit / keuntungan yang paling besar, maka untuk setiap perluasan simpul (simpul-E) akan dicatat jumlah profit yang didapat sejauh ini. Jika pada saat membangun lintasan dari akar ke daun berakhir pada simpul mati, maka kita akan mengecek apakah keuntungan yang didapat lebih besar jika dibandingkan dengan keuntungan yang sudah dicatat pada lintasan sebelumnya. Jika keuntungannya lebih tinggi, maka profit tertinggi akan diubah dan simpul daun yang memiliki profit yang lebih kecil dibandingkan dengan profit tertinggi sekarang akan dimatikan.

Jika jumlah simpul dalam pohon ruang status adalah  $2^n$  atau  $n!$ , maka pada kasus terburuknya, algoritma runut balik membutuhkan waktu / kompleksitas algoritma  $O(p(n)2^n)$  atau  $O(q(n)n!)$ , dengan  $p(n)$  dan  $q(n)$  adalah polinom derajat  $n$  yang menyatakan waktu komputasi setiap simpul.



Gambar 5. Contoh Knapsack Problem

Sumber: [https://en.wikipedia.org/wiki/Knapsack\\_problem](https://en.wikipedia.org/wiki/Knapsack_problem)

(Diakses pada 10 Mei 2021, 22.00 WIB)

### C. Supply Chain Management

Seperti yang sudah disinggung pada bagian pendahuluan, Supply Chain Management merupakan proses yang melibatkan banyak proses dan berbagai pihak. Sebuah produk atau layanan jasa bisa sampai ke tangan konsumen setelah melalui berbagai tahapan dalam Supply Chain Management. Tahapan-tahapan tersebut:

#### 1. Customer

Tahap pertama yang memulai Supply Chain Management adalah konsumen atau customer. Pada tahap ini, customer mengajukan pesanan atau order suatu produk yang ditawarkan oleh pihak produsen, atau dengan kata lain dapat diringkas menjadi demand.

#### 2. Planning

Setelah pesanan diterima oleh produsen, tim atau departemen perencanaan akan membuat perencanaan mengenai produksi produk yang diminta. Selain itu, tim perencanaan juga bertanggung jawab terhadap persiapan kebutuhan baku yang diperlukan.

#### 3. Purchasing

Tahap ini dilakukan oleh departemen pembelian setelah menerima rencana produksi yang dilakukan departemen perencanaan. Pada tahap ini, departemen pembelian akan menghubungi pemasok untuk melakukan pembelian bahan baku dan bahan pendukung. Selain melakukan pembelian, departemen ini juga menentukan tanggal penerimaan bahan dan jumlah bahan baku yang dibeli.

#### 4. Inventory

Setelah pesanan bahan baku telah datang, selanjutnya bahan baku tersebut akan dikirim ke pabrik untuk dilakukan

pemeriksaan kualitas. Apabila kualitas bahan baku dan jumlahnya telah memenuhi standar, bahan baku tersebut akan disimpan di gudang.

### 5. Produksi

Ketika bahan baku telah diperiksa kualitasnya dan disimpan, nantinya bahan baku tersebut akan diproses Bersama bahan pendukung menjadi bahan jadi yang dipesan oleh customer. Setelah selesai diproduksi, bahan baku yang telah menjadi bahan jadi akan kembali disimpan di gudang dan nantinya produk tersebut akan dikirim kepada customer.

### 6. Pengiriman

Tahap ini merupakan bagian terakhir dari Supply Chain Management. Pada tahap ini, produk yang disimpan di Gudang akan dikirimkan kepada customer sesuai tanggal pengiriman yang diminta. Proses pengiriman ini dapat dikirim ke distributor/agen ataupun langsung ke customer.

### D. Gudang (Warehouse)

Tipe dan jenis gudang dibedakan dalam dua kategori besar yaitu tipe secara umum serta tipe berdasarkan karakteristik penyimpanannya. Terdapat 3 jenis gudang dalam tipe umum sedangkan berdasarkan tipe karakteristik penyimpanannya, terdapat setidaknya 6 jenis gudang.

#### 1. Tipe/Jenis gudang secara umum:

- Gudang pribadi / swasta  
Biasanya digunakan oleh perusahaan dagang atau *reseller*.
- Gudang publik  
Ketika perusahaan swasta memerlukan ruang penyimpanan tambahan untuk barang mereka, maka gedung publik bisa disewa untuk jangka pendek.
- Gudang otomatis

Gudang ini memanfaatkan teknologi dalam proses pengangkutan hingga penyimpanan barang sehingga memerlukan lebih sedikit SDM.

#### 2. Tipe/Jenis gudang berdasarkan karakteristik penyimpanan:

- Gudang bahan baku  
Gedung tempat bahan baku disimpan biasanya dibangun sekitar pusat produksi.
- Gudang barang setengah jadi  
Barang setengah jadi disimpan di gudang ini.
- Gudang hasil produksi  
Barang atau produk perusahaan yang siap dijual atau didistribusikan akan disimpan di gudang hasil produksi.
- Gudang transit  
Gudang yang satu ini menjadi titik temu antara berbagai barang perusahaan sebelum dikirim ke konsumen atau bisa dibeli konsumen.
- Gudang cross docking

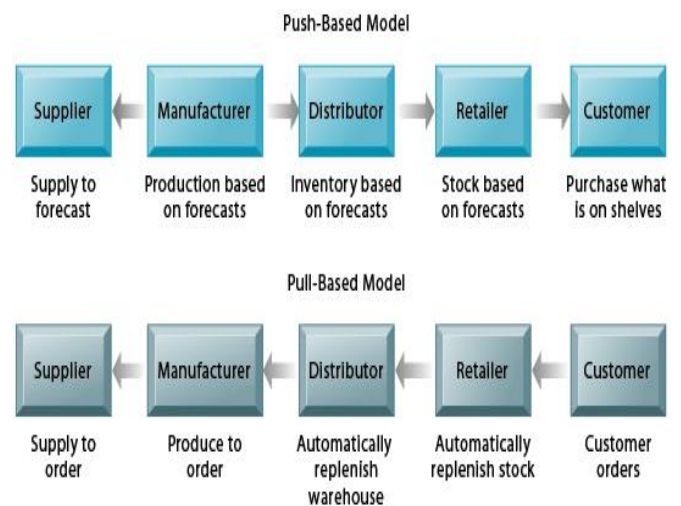
Gudang untuk barang-barang dengan perputaran cepat (buah, sayur, atau makanan dengan kadaluarsa cepat).

- Gudang kepentingan publik  
Contoh gudang publik adalah Gudang Bulog yang mana dikelola oleh negara sebagai tempat penyimpanan beras.

## III. PENERAPAN ALGORITMA BACKTRACKING DALAM SUPPLY CHAIN MANAGEMENT

### A. Overview

Sebagai gambaran kasar untuk mengaplikasikan algoritma backtracking pada proses peletakan barang di gudang, baiknya kita mengetahui bagaimana alur / flow yang terjadi pada *Supply Chain Management*.



Gambar 6. Push and Pull Model

Sumber: <https://paginas.fe.up.pt/~als/mis10e/ch9/chpt9-2bullettext.htm>

(Diakses pada 11 Mei 2021, 12.14 WIB)

Dapat diperhatikan bahwa alur dari *Supply Chain Management* ini dapat berjalan dua arah, yaitu Push-Based Model dan Pull-Based Model.

Push-Based Model (build-to-order) sering digunakan pada zaman dahulu sebelum internet sudah sepopuler sekarang. Dulu, perusahaan industry memproduksi barang terlebih dahulu, lalu menyimpannya di gudang, baru produk tersebut dijual saat ada customer yang ingin membeli. Namun metode ini kurang efektif karena perusahaan harus memprediksi besar kecilnya arus demand dari customer, serta hingga kapan produk tersebut akan trend. Hal ini dapat menjadi masalah jika perusahaan salah dalam melakukan prediksi. perusahaan dapat memproduksi lebih dari jumlah demand yang diinginkan, yang dapat meningkatkan biaya produksi secara sia-sia karena belum tentu produk yang dijual adalah produk yang dapat disimpan. Namun, cara ini tidak dapat dihindari karena tentu saja waktu



produksi suatu produk belum tentu cepat dan belum tentu customer mau menunggu waktu proses produksi yang dapat menyebabkan Makalah IF2120 Matematika Diskrit – Sem. I Tahun 2020/2021 customer tidak jadi membeli produk tersebut. Jadi, perusahaan lebih memilih rugi saat proses produksi daripada tidak ada customer yang membeli produknya

Oleh karena itu, setelah internet ditemukan dan mulai dikenal oleh masyarakat kegunaannya, terciptalah Supply Chain Management berjenis Pull-Based Model. Pull-Based Model (build-to-order) ini berkebalikan dengan Push-Based Model, dimana aksi customer melakukan order lah yang memicu terjadinya flow pada Supply Chain Management. Dengan bantuan internet, perusahaan industry dapat meminimalisir kerugian mereka karena jumlah demand dari customer akan dapat diprediksi secara lebih akurat.

Perbedaan “Push” dan “Pull” ini dapat diringkas melalui sebuah slogan yaitu

**“Make what we sell, not sell what we make”.**

### B. Prinsip Gudang (Warehouse)

Sebelum kita memecah masalah menjadi persoalan 0/1 knapsack dan menggunakan algoritma backtracking, kita perlu mengetahui apa itu *warehouse* dan prinsip-prinsip penting yang berkaitan dengan *warehouse*.

Bagi perusahaan yang bergerak dalam bidang retail atau perdagangan, memiliki *warehouse* yang baik menjadi sangat krusial. Gudang dengan manajemen baik akan mendukung pemasokan barang perusahaan secara rapi dan sistematis. Sebaliknya, gudang dengan manajemen buruk mampu memberikan dampak negatif kepada penjualan bahkan keuntungan karena tidak terkontrolnya *supply chain*.

Dalam proses perencanaan, ada beberapa prinsip Gudang yang perlu diperhatikan seperti kecepatan hingga efisiensi. Dengan mengetahui prinsip-prinsip tersebut, maka pembuatan gudang akan lebih terarah dan mampu menunjang kinerja perusahaan. Berikut ini 5 prinsip gudang atau *warehouse*:

#### 1. Proses yang cepat (Speedily)

Gudang harus mampu memenuhi permintaan pelanggan dengan cepat sekaligus menyanggupi terjadinya perubahan permintaan. Ketika pesanan masuk, *warehouse* harus mampu memproses pesanan tersebut melalui manajemen serta strategi yang baik.

#### 2. Proses yang efisien (Efficiently)

Sebuah gudang harus didesain dan diatur seefisien mungkin. Segala macam kegiatan yang berhubungan dengan *warehouse* seperti pengiriman, penerimaan, pencatatan, atau penyimpanan barang harus efisien melalui *supply chain* yang terintegrasi.

#### 3. Proses yang efektif (Effectiveness)

Memastikan pelanggan mendapatkan produk mereka dengan mudah atau membuat perusahaan menemukan produk pesanan dengan cepat adalah jenis efektivitas.

#### 4. Keandalan (Reliability)

Memastikan segala komponen perusahaan termasuk *warehouse* memiliki keandalan dalam proses komunikasi, bertukar informasi, sekaligus eksekusi. Dengan begitu, perpindahan barang menjadi lebih baik sehingga perusahaan mampu memperpendek *lead time* pelanggan.

#### 5. Akurat dalam penyediaan barang (Accuracy)

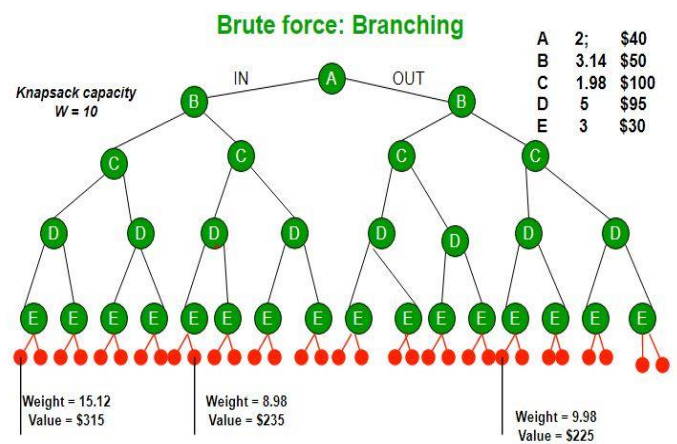
Warehouse haruslah akurat dalam menyediakan barang sehingga mampu memenuhi permintaan. Namun, ketika barang yang disimpan berlebihan, tentu menjadi tidak efisien sekaligus *costly*. Dengan begitu, sangat penting untuk mengatur jumlah barang tersimpan secara tepat dan akurat sesuai dengan kondisi pasar.

Dari lima prinsip di atas, makalah ini akan membahas prinsip kedua yaitu proses yang efisien, lebih detailnya penyimpanan barang secara efisien. Akan dibahas bagaimana algoritma backtracking dengan pendekatan persoalan 0/1 knapsack.

### C. Backtracking 0/1 Knapsack

Berat (*weight*) pada persoalan *knapsack* ini dapat berupa berat ataupun volume tergantung dari jenis barang dan jenis gudang. Keuntungan (*profit*) merupakan harga jual dari suatu barang.

Sebelum menunjukkan bagaimana algoritma backtracking dapat memangkas kompleksitas waktu yang dibutuhkan, akan ditunjukkan *tree* dari algoritma brute force.



Gambar 7. Tree 0/1 Knapsack dengan algoritma Brute Force

Sumber: <https://www.geeksforgeeks.org/0-1-knapsack-using-branch-and-bound/>

(Diakses pada 11 Mei 2021, 16.26 WIB)

Dapat dilihat pada gambar 7, ada 32 rute lintasan dari akar ke daun paling bawah. Algoritma brute force ini memeriksa semua kemungkinan lintasan lalu mengecek lintasan yang paling optimal. Algoritma ini walaupun memiliki kompleksitas algoritma yang lama/besar, tetapi selalu menghasilkan hasil yang optimum. Sekarang akan dilihat *tree* dari algoritma *Backtracking*.

#### IV. KESIMPULAN

Supply Chain Management sangatlah penting dalam dunia bisnis untuk mengurangi berbagai pengeluaran. Pada pengaplikasiannya, penggunaan algoritma backtracking pada proses peletakkan barang di gudang berperan penting dalam memulai penggunaan SCM secara baik. Walaupun Supply Chain Management ini sangat berguna, namun penerapannya di Indonesia masih belum maksimal karena masih terlalu banyak factor eksternal yang menghambat berjalannya alur SCM sehingga dalam implementasinya terbatas.

#### V. UCAPAN TERIMA KASIH

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena dengan rahmat dan berkat-Nya, makalah ini dapat diselesaikan tepat waktu. Penulis juga mengucapkan terima kasih kepada kedua orangtua, serta teman-teman yang telah memberikan dukungan selama pengerjaan makalah ini. Tidak lupa juga penulis mengucapkan terima kasih kepada Bapak Ir. Rila Mandala, M.Eng., Ph.D. selaku dosen K1 mata kuliah Strategi Algoritma yang telah memberi materi kepada penulis pada Semester keempat Tahun Ajar 2020/2021. Penulis meminta maaf jika terdapat kesalahan kata-kata dalam makalah ini, penulis berharap makalah ini dapat digunakan sebaik-baiknya.

VIDEO LINK AT YOUTUBE

<https://youtu.be/Q7m5S5sDkqw>

#### REFERENCES

- [1] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-backtracking-2021-Bagian1.pdf> , diakses pada 10 Mei 2021 pukul 19.00 WIB
- [2] <https://www.shsu.edu/programs/bachelor-of-business-administration-in-supply-chain-management/> , diakses pada 10 Mei 2021 pukul 18.03 WIB
- [3] <https://ukirama.com/id/blogs/prinsip-tipe-jenis-fungsi-gudang-serta-dan-atribut-yang-perlu-diperhatikan-pada-gudang-warehouse> , diakses pada 11 Mei 2021 pukul 14.16 WIB

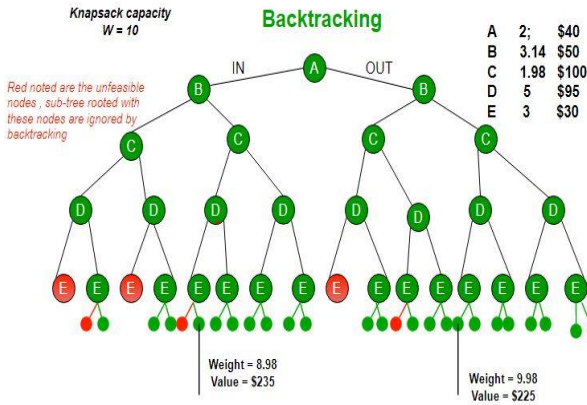
#### PERNYATAAN

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.

Tangerang, 11 Mei 2021



Karlsten Adiyasa Bachtiar 13519001



Gambar 8. Tree 0/1 Knapsack dengan algoritma Backtracking

Sumber: <https://www.geeksforgeeks.org/0-1-knapsack-using-branch-and-bound/>

(Diakses pada 11 Mei 2021, 16.26 WIB)

Dari gambar 8 dapat dilihat bahwa hanya ada 23 rute lintasan dari akar ke daun. Algoritma Backtracking ini akan meng-*ignore* simpul-simpul yang tidak mengarah ke *goal* sehingga dapat memangkas (*pruning*) kompleksitas algoritma. Algoritma backtracking juga sama seperti algoritma brute force, yaitu selalu menghasilkan hasil yang optimum. Algoritma ini merupakan algoritma yang umum karena dapat diaplikasikan pada banyak bidang dengan batasan / constraint yang sedikit.

#### D. Penerapan SCM di Indonesia

Di Indonesia, SCM masih belum dapat diterapkan dengan baik dikarenakan terlalu banyak factor eksternal yang dapat memengaruhi kinerja dari system yang dibuat. Contoh pada dunia nyata adalah ketika perusahaan sudah memasukkan produk ke dalam kendaraan pengirim dengan susunann yang sudah diatur sedemikian rupa, tetapi tiba-tiba ada salah satu mitra nya yang menginginkan produk yang sama namun dalam waktu secepatnya karena urgent. Di Indonesia, hubungan antar individu memiliki prioritas yang lebih signifikan daripada system yang sudah dibuat. Permintaan “urgent” ini dapat menunda alur pengiriman secara massive. Inilah alasannya mengapa Supply Chain Management kurang efektif jika dijalankan di Indonesia karena pengaplikasiannya terbatas oleh berbagai factor eksternal.

Namun menurut saya, Indonesia sudah harus mulai masuk pada era yang dapat memakai SCM dengan baik. Dimulai dari proses penyimpanan barang yang efisien sehingga proses memasok dan mengambil barannng dari gudang dappat berjalan dengan efisien. Walaupun memang ada faktor-faktor eksternal lainnya yang dapat menghambat ketepatan waktu pengiriman, seperti masih parahnya kemacetan yang ada di Indonesia.