

# Optimisasi Profit Penjualan dengan Menerapkan Algoritma Branch and Bound pada Penentuan Barang Dagangan

Rais Vaza Man Tazakka (13519060)

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
E-mail: raisvaza@gmail.com

**Abstrak**—Makalah ini bertujuan untuk menunjukkan cara mengoptimisasi profit penjualan dengan menerapkan algoritma Branch and Bound pada proses penentuan pembelian barang dagangan. Memperoleh profit dari penjualan merupakan tujuan utama kegiatan penjualan. Oleh karena itu, optimisasi profit penjualan menjadi penting bagi pelaku usaha (penjual). Kajian ini dilakukan berdasarkan studi pustaka. Algoritma Branch and Bound merupakan salah satu desain/strategi algoritma yang dapat digunakan untuk memaksimalkan profit penjualan pada penentuan pembelian barang dagangan secara optimal.

**Kata Kunci**—*optimisasi; profit; penjualan; Branch and Bound; Integer Knapsack Problem;*

## I. PENDAHULUAN

Algoritma Branch and Bound adalah salah satu strategi/desain algoritma yang dapat digunakan dalam memecahkan masalah optimisasi. Pada masalah optimisasi, seorang individu diminta untuk meminimalkan atau memaksimalkan suatu fungsi objektif yang bersangkutan tanpa melanggar batasan/*constraints* persoalan.

Algoritma Branch and Bound berbasis algoritma Breadth-First Search (BFS). Perbedaan antara algoritma Branch and Bound dan algoritma BFS terletak pada penentuan simpul yang diekspansi pada langkah selanjutnya. Pada algoritma BFS, ekspansi simpul dilakukan berdasarkan urutan pembangkitan simpul. Pada algoritma Branch and Bound, ekspansi simpul dilakukan pada simpul yang paling memenuhi fungsi objektif terlebih dahulu, yaitu pada simpul dengan *cost* paling kecil untuk kasus minimasi dan pada simpul dengan *cost* paling besar untuk kasus maksimasi.

Algoritma Branch and Bound dapat menyelesaikan berbagai macam masalah. Salah satu masalah yang dapat ia selesaikan adalah Integer Knapsack Problem. Pada masalah tersebut, terdapat suatu himpunan objek yang dapat dipilih. Setiap objek memiliki nilai profit dan bobotnya masing-masing. Solusi dari permasalahan ini adalah subhimpunan objek dengan profit tertinggi yang tidak melanggar bobot total yang dapat ditampung oleh “knapsack” permasalahan tersebut. Dalam kata lain, subhimpunan objek tersebut tidak melanggar batasan/*constraint* yang terdefinisi untuk masalah tersebut.

Salah satu instansiasi dari masalah Integer Knapsack adalah pemilihan barang dagangan dengan sejumlah modal tertentu

untuk memaksimalkan profit. Pada kasus ini, seorang pedagang memiliki sejumlah modal untuk membeli barang yang, kemudian, akan ia jual. Berdasarkan pilihan barang yang tersedia, pedagang menentukan barang apa saja yang ia harus beli dan barang apa yang tidak seharusnya ia beli agar ia dapat memperoleh profit maksimal.

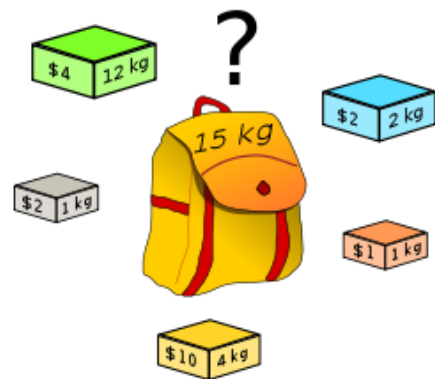
## II. LANDASAN TEORI

### A. Penjualan dan Profit

Penjualan adalah kegiatan menukar barang dengan alat tukar yang sah, seperti uang, yang dilakukan antara penjual dengan pembeli. Pembeli mendapatkan barang yang dijual dan penjual mendapatkan untung, salah satunya dalam bentuk uang, sebagai gantinya.

Profit, berdasarkan Kamus Besar Bahasa Indonesia, adalah keuntungan. Oleh karena itu, profit penjualan adalah keuntungan yang diperoleh dari kegiatan menukar barang dengan alat tukar yang sah.

### B. Integer Knapsack Problem



Bagan 1 Ilustrasi Integer Knapsack Problem

Sumber: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pengantar-Strategi-Algoritma-\(2021\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pengantar-Strategi-Algoritma-(2021).pdf)

Integer Knapsack Problem adalah persoalan kombinatorial. Diberikan  $n$  buah barang dengan berat  $w_1, w_2, \dots, w_n$  dan nilai  $v_1, v_2, \dots, v_n$  dan sebuah *knapsack*/ransel dengan kapasitas  $W$ , carilah subhimpunan barang-barang tersebut dengan nilai total paling besar yang muat untuk ransel tersebut.

Secara matematis, persoalan Integer Knapsack dideskripsikan sebagai berikut.

$$\text{Maksimasi } F = \sum_{i=1}^n p_i x_i \text{ dengan kendala (constraint)}$$

$$\sum_{i=1}^n w_i x_i \leq K, x_i = 0 \text{ atau } x_i = 1, \text{ dan } i = 1, 2, \dots, n.$$

Bagan 2 Formulasi Matematis Integer Knapsack Problem

Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian4.pdf>

Solusi persoalan Integer Knapsack direpresentasikan dalam tuple  $(x_1, x_2, \dots, x_n)$ .

### C. Algoritma Branch and Bound

Algoritma *Branch and Bound* merupakan salah satu strategi/desain algoritma yang dapat digunakan untuk permasalahan optimisasi, yaitu meminimalkan atau memaksimalkan suatu fungsi objektif, dengan suatu batasan. Pada desain algoritma ini, permasalahan direpresentasikan sebagai pohon ruang status. Setiap simpul  $i$  pada pohon tersebut memiliki *cost*  $c(i)$ .  $c(i)$  adalah nilai taksiran lintasan termurah ke simpul status tujuan yang melalui simpul status  $i$  untuk kasus minimasi dan nilai taksiran lintasan termahal ke simpul status tujuan yang melalui simpul status  $i$  untuk kasus maksimasi.

Algoritma Branch and Bound dibangun atas algoritma Breadth-First Search (BFS) sehingga pembangkitan simpul pada algoritma Branch and Bound sama seperti pembangkitan simpul pada algoritma BFS. Yang membedakan algoritma Branch and Bound dari algoritma BFS adalah penentuan ekspansi simpul. Pada algoritma BFS, ekspansi simpul dilakukan berdasarkan urutan pembangkitan. Pada algoritma Branch and Bound, ekspansi simpul dilakukan berdasarkan nilai *cost* simpul-simpul yang telah dibangkitkan. Untuk persoalan minimasi, ekspansi simpul dilakukan terhadap simpul dengan nilai *cost* paling kecil. Pada persoalan maksimasi, ekspansi simpul dilakukan terhadap simpul dengan nilai *cost* paling besar.

Seperti algoritma Backtracking, algoritma Branch and Bound membunuh simpul yang tidak mengarah menuju solusi. Pembunuhan simpul dilakukan ketika nilai simpul tidak lebih baik dari nilai terbaik sejauh ini atau simpul tidak merepresentasikan solusi yang *feasible* karena melanggar suatu batasan.

### III. SOLUSI UMUM PERSOALAN

Dalam suatu persoalan, seorang penjual memiliki modal  $K$ . Terdapat  $n$  buah barang yang dapat ia beli untuk kemudian ia jual. Setiap barang memiliki harga dan profitnya masing-masing.

Tabel 1 Contoh Daftar Barang dengan Harga dan Profitnya

i	Barang ke-i	Harga	Profit
1	$x_1$	$w_1$	$p_1$
2	$x_2$	$w_2$	$p_2$
...	...	...	...
n	$x_n$	$w_n$	$p_n$

Dari  $n$  buah barang yang ada, seseorang harus menentukan barang mana saja yang akan ia beli untuk mendapatkan profit maksimal tanpa melebihi modal  $K$  yang ia miliki.

Untuk menyelesaikan persoalan Integer Knapsack dengan algoritma Branch and Bound, pertama-tama, perlu diingat bahwa penyelesaian persoalan akan direpresentasikan dalam bentuk pohon ruang status yang setiap simpul  $i$ -nya memiliki nilai *cost*  $c(i)$ . Karena persoalan Integer Knapsack adalah persoalan maksimasi, *cost* setiap simpul pada pohon ruang status menyatakan batas atas dari solusi optimum. Simpul yang diekspansi pada setiap langkah adalah simpul hidup yang memiliki *cost* paling besar. Agar pencarian solusi mangkus, barang-barang diurutkan berdasarkan  $p_i/w_i$  yang menurun (dari besar ke kecil). Pohon ruang status yang dihasilkan adalah pohon biner yang cabang kirinya menyatakan barang ke- $i$  dipilih dan cabang kanannya menyatakan barang ke- $i$  tidak dipilih. Tiap simpul pada aras/level  $i$  di dalam pohon biner tersebut menyatakan himpunan bagian dari  $n$  buah barang yang dimasukkan ke dalam *knapsack*. Tiap simpul memiliki nilai total bobot *knapsack* yang sudah terpakai ( $W$ ) dan total keuntungan yang sudah dicapai ( $F$ ). *Cost*  $c(i)$  atau batas atas simpul  $i$  dihitung sebagai keuntungan  $F$  yang sudah diperoleh ditambah dengan perkalian sisa kapasitas *knapsack* ( $K-W$ ) dengan rasio keuntungan per bobot barang yang tersisa berikutnya  $(p_{i+1}/w_{i+1})$ .

$$c(i) = F + (K - W)(p_{i+1}/w_{i+1})$$

Bagan 3 Rumus Cost Simpul ke- $i$  untuk Persoalan Integer Knapsack dengan Algoritma Branch and Bound

Sumber: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/20-2021/Algoritma-Branchand-Bound-2021-Bagian4.pdf>

Langkah-langkah penyelesaian persoalan Integer Knapsack dengan algoritma Branch and Bound adalah sebagai berikut.

1. Hitung  $p_i/w_i$  untuk semua barang
2. Urutkan barang-barang berdasarkan  $p_i/w_i$  yang menurun
3. Bangkitkan simpul akar dengan nilai  $W = 0$  dan  $F = 0$  dan hitung *cost*-nya
4. Bangkitkan simpul anak kiri dan kanan
5. Ekspansi simpul hidup dengan *cost* paling besar
6. Jika simpul ekspansi merupakan simpul pada level  $n$ :
  - a. Jika terdapat simpul hidup dengan  $cost \geq cost$  simpul pada level  $n$  tersebut, ekspansi simpul hidup tersebut
  - b. Jika tidak, simpul pada level  $n$  tersebut adalah simpul solusi dan semua simpul dengan *cost* yang lebih kecil dari *cost* simpul solusi tersebut dimatikan
7. Jika simpul ekspansi bukan simpul pada level  $n$ :
  - a. Jika  $W$  melebihi  $K$ , simpul dimatikan
  - b. Jika  $W$  tidak melebihi  $K$ , kembali ke langkah 4

#### IV. STUDI KASUS

Seorang penjual kaos memiliki modal sebesar Rp1.000.000,00. Terdapat 6 barang yang dapat ia beli untuk kemudian ia jual. Berikut adalah daftar barang serta harga dan profitnya.

Tabel 2 Daftar Barang dengan Harga dan Profitnya

i	Barang ke-i	Harga	Profit
1	Kaos Malioboro	Rp50.000,00	Rp10.000,00
2	Kaos Berlin	Rp200.000,00	Rp50.000,00
3	Kaos bekas artis	Rp500.000,00	Rp200.000,00
4	Kaos polos	Rp100.000,00	Rp20.000,00
5	Kaos tie-dye	Rp250.000,00	Rp50.000,00
6	Kaos turtleneck	Rp200.000,00	Rp100.000,00

Penjual harus menentukan kaos mana saja yang akan ia beli dan kaos mana saja yang tidak akan ia beli dengan tepat sehingga ia dapat memperoleh profit yang maksimal.

Dalam kasus ini,  $K = \text{Rp}1.000.000,00$  dan terdapat 6 kandidat barang yang akan dipilih. Pertama-tama, hitung  $p_i/w_i$  setiap barang.

Tabel 3 Daftar Barang dengan Harga, Profit, serta Rasio-Profit-per-Harga-nya

i	Barang ke-i	Harga	Profit	$p_i/w_i$
1	Kaos Malioboro	Rp50.000,00	Rp10.000,00	0,2
2	Kaos Berlin	Rp200.000,00	Rp50.000,00	0,25
3	Kaos bekas artis	Rp500.000,00	Rp200.000,00	0,4
4	Kaos polos	Rp100.000,00	Rp20.000,00	0,2
5	Kaos tie-dye	Rp250.000,00	Rp50.000,00	0,2
6	Kaos turtleneck	Rp200.000,00	Rp100.000,00	0,5

Kemudian, urutkan barang berdasarkan  $p_i/w_i$  yang lebih besar terlebih dahulu.

Tabel 4 Daftar Barang yang Sudah Diurutkan berdasarkan Rasio-Profit-per-Harga-nya

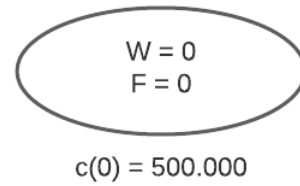
i	Barang ke-i	Harga	Profit	$p_i/w_i$
1	Kaos turtleneck	Rp200.000,00	Rp100.000,00	0,5
2	Kaos bekas artis	Rp500.000,00	Rp200.000,00	0,4
3	Kaos Berlin	Rp200.000,00	Rp50.000,00	0,25
4	Kaos polos	Rp100.000,00	Rp20.000,00	0,2
5	Kaos tie-dye	Rp250.000,00	Rp50.000,00	0,2
6	Kaos Malioboro	Rp50.000,00	Rp10.000,00	0,2

Kemudian, bangkitkan simpul akar dengan nilai  $W = 0$  dan  $F = 0$  dan hitung *cost*-nya.

$$c(0) = F + (K - W) \left( \frac{p_1}{w_1} \right) = 0 + (1.000.000 - 0)(0,5)$$

$$c(0) = 500.000$$

Simpul 0 (Akar)



Bagan 4 Simpul 0 (Akar)

Selanjutnya, bangkitkan simpul anak kiri dan kanan dari simpul akar ini. Simpul kiri menyatakan bahwa barang ke-i diambil dan simpul kanan menyatakan bahwa barang ke-i tidak diambil.

$$c(1) = F + (K - W) \left( \frac{p_2}{w_2} \right)$$

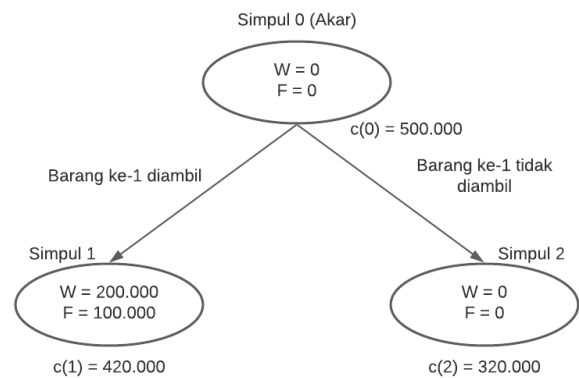
$$c(1) = 100.000 + (1.000.000 - 200.000)(0,4)$$

$$c(1) = 420.000$$

$$c(2) = F + (K - W) \left( \frac{p_2}{w_2} \right)$$

$$c(2) = 0 + (1.000.000 - 200.000)(0,4)$$

$$c(2) = 320.000$$



Bagan 5 Ekspansi Simpul Akar

Karena *cost* simpul 1 lebih besar dari *cost* simpul 2, simpul 1 diekspansi.

$$c(3) = F + (K - W) \left( \frac{p_3}{w_3} \right)$$

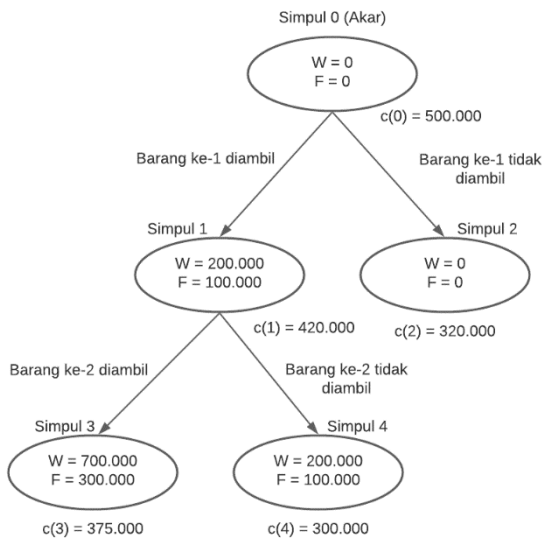
$$c(3) = 300.000 + (1.000.000 - 700.000)(0,25)$$

$$c(3) = 375.000$$

$$c(4) = F + (K - W) \left( \frac{p_3}{w_3} \right)$$

$$c(4) = 100.000 + (1.000.000 - 200.000)(0,25)$$

$$c(4) = 300.000$$



Bagan 6 Ekspansi Simpul 1

Karena *cost* simpul 3 lebih besar dari *cost* simpul 4 dan *cost* simpul 2 dan *W* simpul 3 belum melebihi modal *K*, simpul 3 diekspansi.

$$c(5) = F + (K - W) \left( \frac{p^4}{w^4} \right)$$

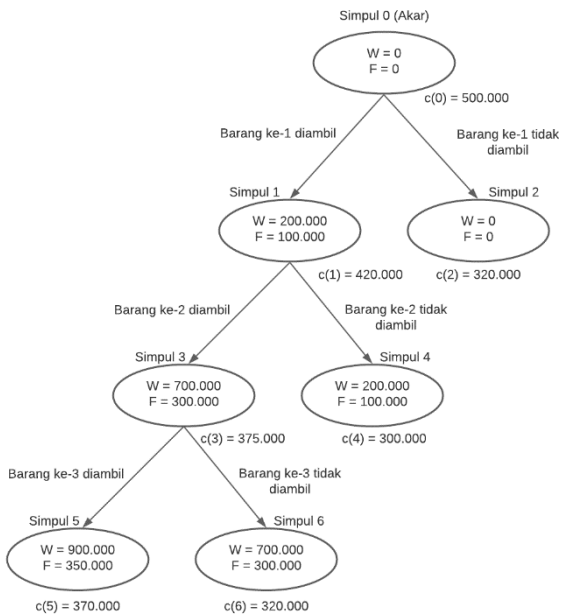
$$c(5) = 350.000 + (1.000.000 - 900.000)(0,2)$$

$$c(5) = 370.000$$

$$c(6) = F + (K - W) \left( \frac{p^4}{w^4} \right)$$

$$c(6) = 300.000 + (1.000.000 - 700.000)(0,2)$$

$$c(6) = 320.000$$



Bagan 7 Ekspansi Simpul 3

Karena Simpul 5 memiliki *cost* yang paling besar dibandingkan dengan Simpul 6, Simpul 4, dan Simpul 2, kita ekspansi Simpul 5.

$$c(7) = F + (K - W) \left( \frac{p^5}{w^5} \right)$$

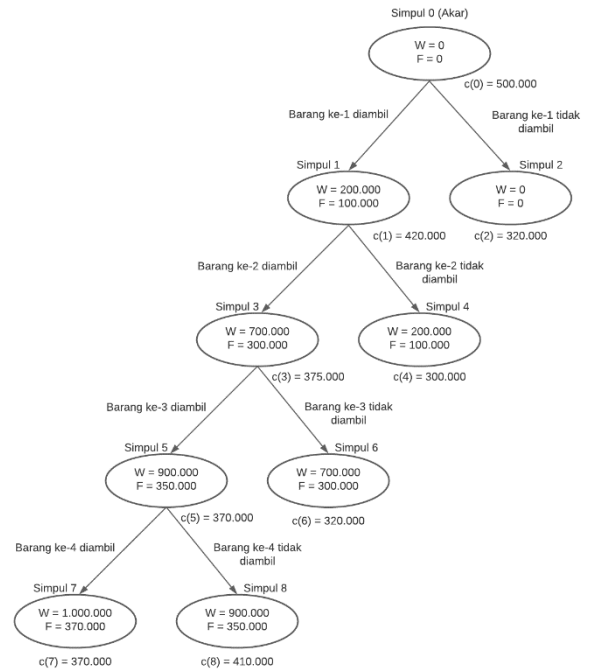
$$c(7) = 370.000 + (1.000.000 - 1.000.000)(0,2)$$

$$c(7) = 370.000$$

$$c(8) = F + (K - W) \left( \frac{p^5}{w^5} \right)$$

$$c(8) = 350.000 + (1.000.000 - 700.000)(0,2)$$

$$c(8) = 410.000$$



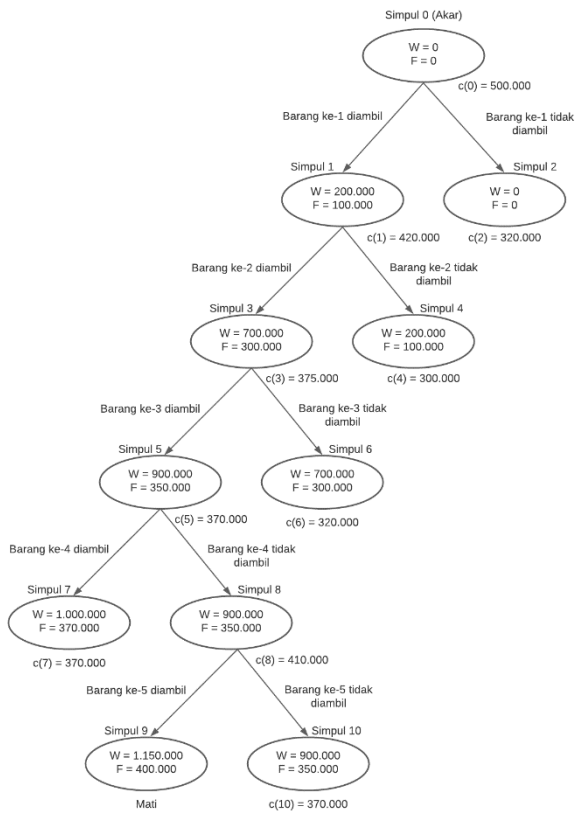
Bagan 8 Ekspansi Simpul 5

Karena Simpul 8 memiliki *cost* yang lebih besar dibandingkan simpul hidup lainnya, Simpul 8 diekspansi. Simpul 9 dimatikan karena ia melanggar batasan, yaitu  $W > K$ .

$$c(10) = F + (K - W) \left( \frac{p^6}{w^6} \right)$$

$$c(10) = 350.000 + (1.000.000 - 900.000)(0,2)$$

$$c(10) = 370.000$$



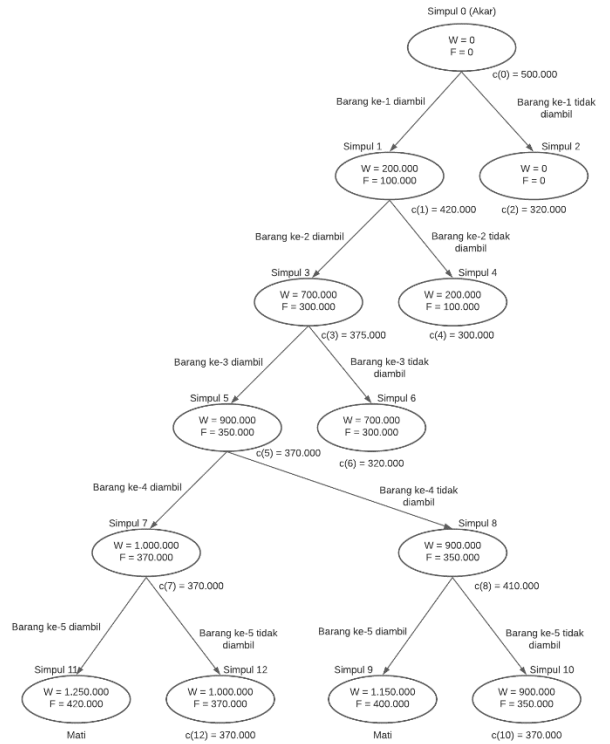
Bagan 9 Ekspansi Simpul 8

Selanjutnya, Simpul 7 diekspansi karena ia memiliki *cost* yang paling besar. Simpul 11 dimatikan karena ia melanggar batasan, yaitu  $W > K$ .

$$c(12) = F + (K - W) \left( \frac{p_6}{w_6} \right)$$

$$c(12) = 370.000 + (1.000.000 - 1.000.000)(0,2)$$

$$c(12) = 370.000$$



Bagan 10 Ekspansi Simpul 7

Selanjutnya, Simpul 10 diekspansi karena ia memiliki *cost* yang paling besar.

$$c(13) = F + (K - W) \left( \frac{p_7}{w_7} \right)$$

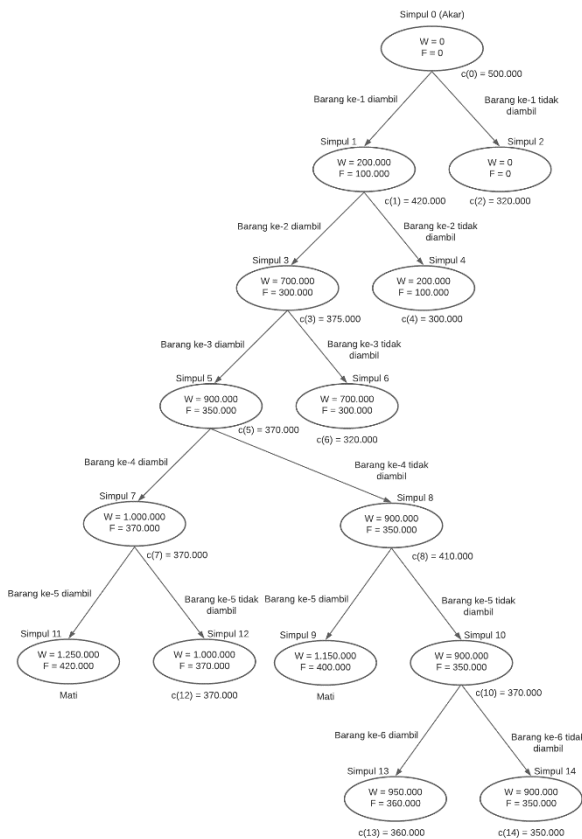
$$c(13) = 360.000 + (1.000.000 - 1.000.000)(0)$$

$$c(13) = 360.000$$

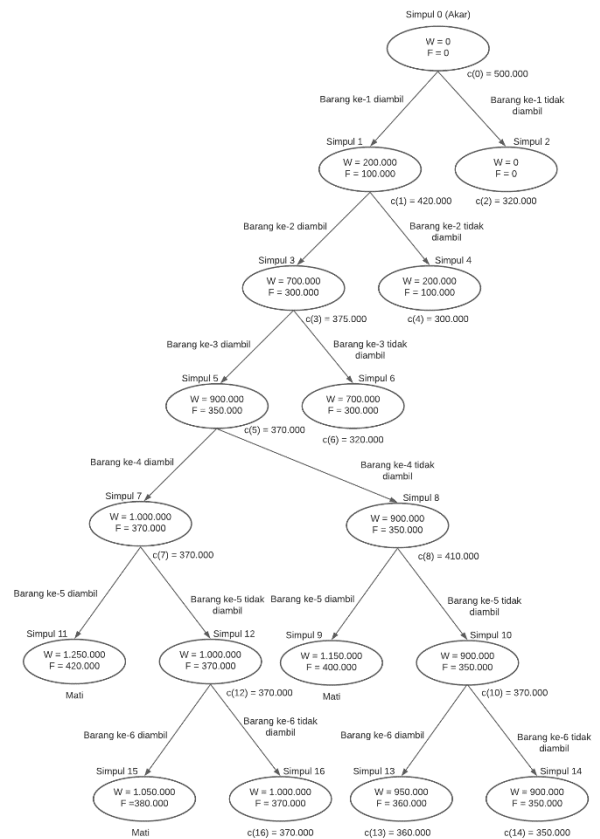
$$c(14) = F + (K - W) \left( \frac{p_7}{w_7} \right)$$

$$c(14) = 350.000 + (1.000.000 - 900.000)(0)$$

$$c(14) = 350.000$$



Bagian 11 Ekspansi Simpul 10



Bagian 12 Ekspansi Simpul 12

Selanjutnya, Simpul 12 diekspansi karena ia memiliki *cost* yang paling besar. Simpul 15 dimatikan karena ia melanggar batasan, yaitu  $W > K$ .

$$c(16) = F + (K - W) \left( \frac{p7}{w7} \right)$$

$$c(16) = 370.000 + (1.000.000 - 900.000)(0)$$

$$c(16) = 370.000$$

Karena Simpul 16 merupakan simpul pada aras/level 6 dan memiliki *cost* paling besar dibandingkan dengan simpul-simpul yang masih hidup, Simpul 16 adalah simpul solusi. Solusi kasus ini dalam tuple adalah  $(x_1, x_2, x_3, x_4, x_5, x_6) = (1, 1, 1, 1, 0, 0)$ . Artinya, kaos yang harus penjual beli pada kasus ini untuk memaksimalkan profit adalah sebagai berikut.

Tabel 5 Daftar Barang yang Harus Dibeli agar Profit Mencapai Maksimum

No	Barang
1	Kaos turtleneck
2	Kaos bekas artis
3	Kaos Berlin
4	Kaos polos

## V. KESIMPULAN

Terdapat beberapa permasalahan aktual yang tergolong ke dalam permasalahan Integer Knapsack. Salah satunya adalah permasalahan optimisasi profit penjualan pada penentuan pembelian barang dagangan.

Ada beberapa strategi/desain algoritma yang dapat digunakan untuk menyelesaikan masalah Integer Knapsack. Salah satunya adalah algoritma Branch and Bound. Sebagaimana telah kita lihat pada studi kasus di atas, algoritma

Branch and Bound menghasilkan solusi yang optimal untuk permasalahan Integer Knapsack.

## VI. PENUTUP

Segala puji bagi Allah swt. yang telah memberikan penulis kelancaran dalam penyusunan makalah ini sehingga makalah ini selesai tepat waktu. Penulis menyampaikan terima kasih kepada Ibu Ulfa Nur Maulidevi, dosen mata kuliah IF2211 Strategi Algoritma untuk kelas K-02, Pak Rinaldi Munir, dosen mata kuliah IF2211, serta seluruh dosen mata kuliah IF2211 Strategi Algoritma yang telah mengajari saya materi-materi berkaitan dengan strategi algoritma sehingga saya dapat menyelesaikan makalah ini. Saya juga menyampaikan terima kasih kepada kedua orang tua serta teman-teman yang memberi dukungan selama proses pengerjaan makalah ini. Penulis berharap makalah ini dapat digunakan dengan sebaik-baiknya dan dikembangkan lebih jauh lagi sehingga dapat memberi manfaat bagi masyarakat.

## PRANALA VIDEO YOUTUBE

<https://www.youtube.com/watch?v=Q-uhRCNYAro>.

## REFERENSI

- [1] A. Levitin, Introduction to The Design and Analysis of Algorithms. Amerika Serikat: Pearson, 2012.
- [2] <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian4.pdf>. Diakses pada 10 Mei 2021
- [3] [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pengantar-Strategi-Algoritma-\(2021\).pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Pengantar-Strategi-Algoritma-(2021).pdf). Diakses pada 11 Mei 2021

## 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, 11 Mei 2021



Rais Vaza Man Tazakka-13519060