AKSIOMA: Jurnal Matematika dan Pendidikan Matematika
Vol. 13, No. 1 April 2022
e-ISSN 2579-7646

Perbandingan Metode *Branch and Bound* dan Enumerasi implisit dalam menyelesaikan masalah *Knapsack*

[1]Fakhry Asad Agusfrianto, [2]Ramya Rachmawati

[1]Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Jakarta
[2]Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Bengkulu
Email: fakhry_asad@yahoo.com

Abstrak

Masalah knapsack merupakan masalah program bilangan bulat yang melibatkan satu kendala saja. Masalah knapsack umumnya diilustrasikan dengan suatu tas dan barang. Masalah yang akan kita selesaikan dalam masalah knapsack adalah memaksimumkan harga barang dengan kapasitas tertentu yang dapat dimuat oleh tas dengan kapasitas tertentu juga. Dalam menyelesaikan masalah knapsack, umumnya dapat dikerjakan secara langsung (penerkaan), menggunakan metode branch and bound, dan enumerasi implisit. Pada paper ini, akan dilakukan perbandingan penyelesaian masalah knapsack dengan metode branch and bound dan enumerasi implisit. Kita juga akan dapat melihat metode mana yang paling efektif untuk menyelesaikan suatu masalah knapsack.

Kata kunci: *masalah knapsack; metode branch and bound; enumerasi implisit*

Abstract

Knapsack problem is an integer programming problem involving only one constrain. Usually, knapsack problem illustrated by a bag and item. A problem which we will solve on knapsack problem is maximizing price of item with particular capacity loadable by bag with particular capacity too. On solve knapsack problem, commonly can be solve directly (guessing), using branch and bound method, and implicit enumeration. In this paper, we will perform comparison solution of knapsack problem using branch and bound method and implicit enumeration. We will also see which method most effective to solve knapsack problem.

Keywords: *knapsack problem; branch and bound method; implicit enumeration*

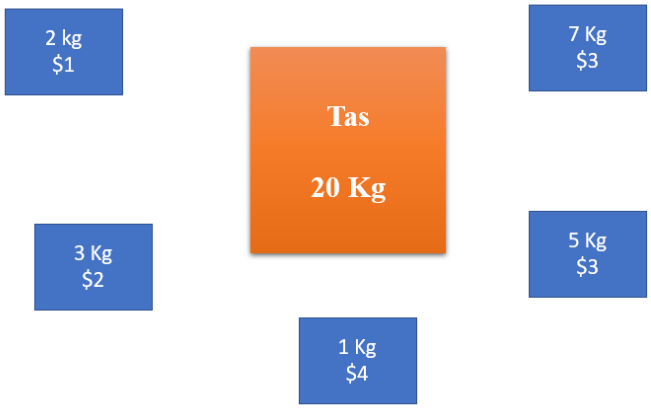
A. Pendahuluan

Masalah program bilangan bulat (*integer programming*) merupakan masalah pemrograman linear yang solusinya harus merupakan bilangan bulat. Penyelesaian program bilangan bulat dapat diselesaikan dengan berbagai cara yaitu dengan metode grafik, metode simpleks, dan metode algoritma bidang potong (Hiller & Lieberman, 2015) dan (Basriati, 2018) . Untuk menyelesaikan program bilangan bulat, kita harus menentukan penyelesaian sebelum nilainya disyaratkan bulat atau umumnya disebut sebagai *LP relaxation* (Winston, 2004). Jika *LP relaxation* sudah bulat, maka *LP relaxation* sekaligus merupakan solusi program bilangan bulat. Jika

belum, maka harus dicari solusi bilangan bulat pada daerah penyelesaian *LP relaxation*. Dengan syarat,

Nilai optimal *LP relaxation* ≥ Nilai optimal program bilangan bulat

Beberapa contoh masalah program bilangan bulat adalah masalah *knapsack*, masalah *lockbox*, dan masalah *set covering* (Winston, 2004). Pada paper ini, akan dibahas penyelesaian masalah *knapsack*. Istilah *knapsack* sendiri berarti karung. Masalah *knapsack* merupakan masalah program bilangan bulat yang hanya melibatkan satu kendala saja (Winston, 2004) dan (Devita & Wibawa, 2020). Ilustrasi masalah dapat dilihat pada Gambar 1 dibawah ini. Misalkan tas berkapasitas 20 Kg dan ada lima barang dengan bobot dan keuntungan yang berbeda-beda. Permasalahannya adalah barang manakah yang harus dimasukkan ke dalam tas supaya keuntungan maksimum dan kapasitas barang yang dimasukkan ke dalam tas tidak melebihi kapasitas tas yang diberikan.



Gambar 1. Ilustrasi masalah *knapsack*

Masalah *knapsack* secara matematis dapat ditulis sebagai berikut.

$$\text{maksimumkan } z = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

$$\text{terhadap kendala : } a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq d$$

$$x_i = 0 \text{ atau } 1 \ (i = 1,2,3, \ldots, n)$$

Sebagai contoh, masalah pada Gambar 1 dapat kita tuliskan sebagai

$$\text{maksimumkan } z = x_1 + 2x_2 + x_3 + 3x_4 + 3x_5$$

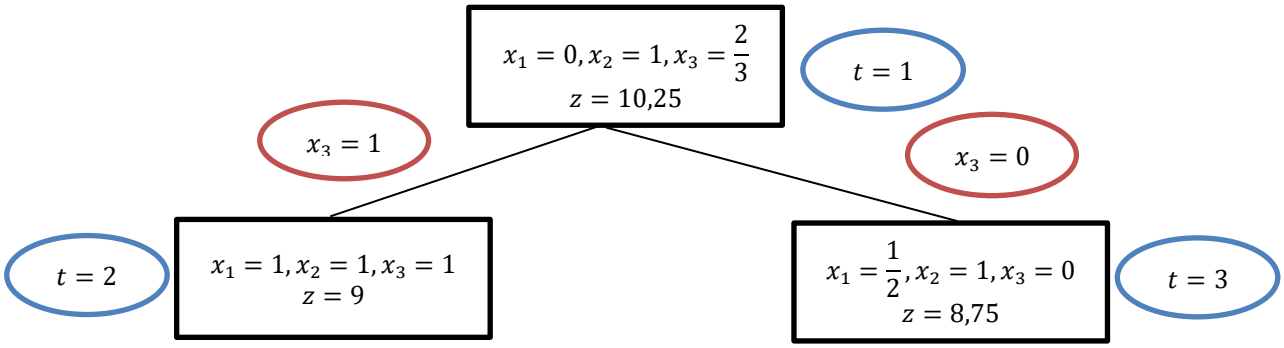
$$\text{terhadap kendala : } 2x_1 + 3x_2 + x_3 + 5x_4 + 7x_3 \leq 20$$

$$x_i = 0 \text{ atau } 1 \ (i = 1,2,3,4,5)$$

Masalah *knapsack* dapat diselesaikan dengan beberapa cara. Kita dapat menerapkan algoritma greedy untuk menyelesaikan masalah *knapsack* (Rachmawati dan Candra, 2013) dan (Sabaruddin, 2016). Selanjutnya, masalah *knapsack* juga dapat diselesaikan dengan metode *branch and bound*. Metode *Branch and Bound* dapat juga digunakan untuk menyelesaikan masalah penjadwalan produksi yang bertujuan untuk meminimalkan waktu produksi (Lesmana, 2016) maupun dalam masalah penjadwalan (Mazda & Kurniawati, 2020) . Selanjutnya, masalah *knapsack* juga dapat diselesaikan dengan algoritma enumerasi implisit. Beberapa penerapan Algoritma enumerasi implisit adalah dapat diterapkan dalam masalah penjadwalan pekerjaan-toko (*job-shop*) (Lageweg et al., 1977) dan masalah desegregasi sekolah (Ligget, 1973). Pada paper ini, akan dilakukan perbandingan penyelesaian masalah *knapsack* dengan menggunakan metode *branch and bound* dan algoritma enumerasi implisit. Kita akan melihat metode mana yang paling efektif untuk menyelesaikan suatu masalah *knapsack*.

B. Metode Penelitian

Penelitian ini merupakan penelitian kajian literatur. Diberikan suatu masalah *knapsack*. Pertama, kita akan mengerjakan dengan metode *branch and bound*. Langkah mengerjakan dengan metode ini adalah kita tentukan peringkat dari item yang tersedia. Item dengan peringkat tertinggi adalah yang kita pilih pertama, item dengan peringkat kedua adalah yang kita pilih kedua, dan seterusnya. Untuk item yang dipilih, $x_i = 1$ dan untuk item yang tidak dipilih, $x_i = 0$. Jika dalam pemilihan item, item melebihi batas kendala yang diberikan, maka bagi sisa dari item tersebut dengan koefisien dari item yang dipilih. Dalam membagi sisa item dengan koefisien item yang dipilih, kita akan peroleh bentuk $x_j = \frac{a}{b}$ dimana ini merupakan bentuk pecahan dan bukan 0 atau 1. Sehingga, nilai inilah yang akan kita cabangkan sedemikian rupa sehingga diperoleh nilai x_j yaitu 0 atau 1. Sebagai ilustrasi pencabangan pada metode *branch and bound,* perhatikan Gambar 2 dibawah ini.



Gambar 2. Ilustrasi pencabangan metode *branch and bound*

Berdasarkan ilustrasi diatas, karena pada $t = 1$ nilai x_3 masih berupa pecahan, maka x_3 harus dicabangkan seperti pada Gambar 2. Ketika $t = 2$, ternyata semua nilai x sudah merupakan bilangan bulat. Sehingga $z = 9$ kita

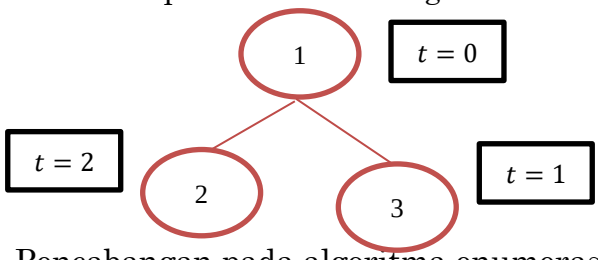
sebut sebagai **solusi kandidat**. Ketika $t = 3$, kita peroleh x_1 masih berupa pecahan. Namun, kita tidak perlu melakukan pencabangan lagi untuk $t = 3$ karena nilai $z = 8{,}75$ tidak lebih baik dari $z = 9$. Hal ini karena kita mencari nilai z yang bulat dan paling maksimum. Sedangkan untuk $z = 8{,}75$, tidak ada lagi bilangan bulat yang dibawah $z = 8{,}75$ yang lebih besar dari $z = 9$. Sehingga, $z = 9$ merupakan solusi dari masalah *knapsack* yang diberikan tersebut.

Selanjutnya, masalah *knapsack* juga dapat diselesaikan dengan algoritma enumerasi implisit. Langkah penyelesaian masalah *knapsack* dengan algoritma enumerasi implisit adalah dengan menentukan terlebih dahulu variabel bebas dan variabel tetapnya. Setelah itu, kita tentukan *best completion* dari masalah *knapsack* tersebut. Setelah itu, nilai dari *best completion* harus ditinjau apakah layak atau tidak. Jika layak, maka itu adalah *best completion* dari masalah *knapsack* tersebut. Jika tidak layak, maka harus ditinjau kelayakan dari masalah *knapsack* tersebut dengan aturan yang tertera pada tabel berikut.

Tipe Kendala	**Tanda dari Koefisien Variabel Bebas dalam Kendala**	**Nilai Variabel Bebas dalam Peninjauan Kelayakan**
$\leq$	+	0
$\leq$	−	1
$\geq$	+	1
$\geq$	−	0

Tabel 1. Aturan pada cek kelayakan

Jika ternyata solusi layak, maka dilakukan pencabangan pada masalah *knapsack* tersebut. Jika tidak layak, maka dapat disimpulkan bahwa masalah *knapsack* tersebut tidak dapat diselesaikan. Ilustrasi pencabangan pada algoritma enumerasi implisit adalah sebagai berikut.



Gambar 3. Pencabangan pada algoritma enumerasi implisit

Pencabangan pada Gambar 3 dilakukan hingga diperoleh *best completion* yang layak untuk masalah *knapsack* yang diberikan.

C. Hasil dan Pembahasan

Dalam paper ini, kita akan menggunakan masalah *knapsack* empat variabel. Misalkan diberikan masalah *knapsack* sebagai berikut

$$\begin{aligned} &\text{maksimumkan } z = 5x_1 + 8x_2 + 3x_3 + 7x_4 \\ &\text{terhadap kendala } 3x_1 + 5x_2 + 2x_3 + 4x_4 \leq 6 \qquad (1) \\ &\text{dengan } x_i = 0 \text{ atau } 1 \ (i = 1,2,3,4) \end{aligned}$$

Pertama, kita akan selesaikan masalah *knapsack* ini menggunakan metode *Branch and Bound*.

Metode *Branch and Bound*

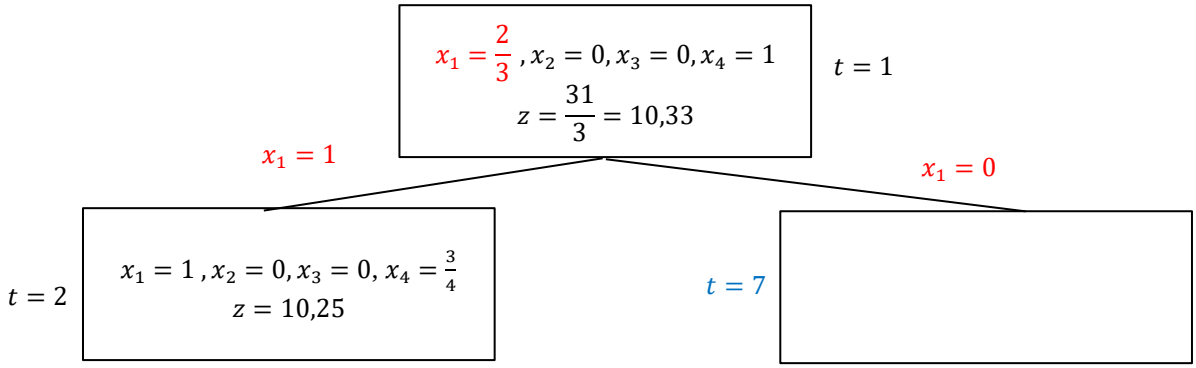
Tahap pertama, kita bentuk tabel sebagai berikut

Item ke-*i*	c_i/a_i	**Peringkat**
1	$\frac{5}{3} = 1{,}67$	2
2	$\frac{8}{5} = 1{,}6$	3
3	$\frac{3}{2} = 1{,}5$	4
4	$\frac{7}{4} = 1{,}75$	1

Kita akan menyelesaikan masalah *knapsack* tersebut dengan langkah sebagai berikut.

Submasalah 1 :

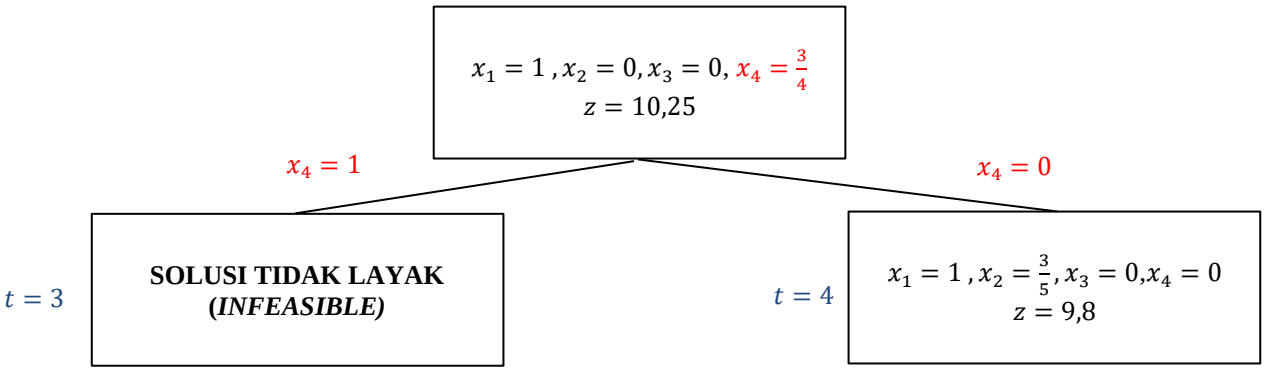
Pilih peringkat tertinggi. Pada kasus ini, x_4 yang dipilih pertama. Karena x_4 dipilih, maka $x_4 = 1$. Sehingga item yang tersisa adalah $6 - 4 = 2$. Selanjutnya, pilih item 1. Perhatikan bahwa, koefisien x_1 pada kendala bernilai 3, sehingga haruslah $x_1 = \frac{sisa}{koefisien\, x_1} = \frac{2}{3}$. Dengan demikian, item yang tersisa adalah 0. Item 2 dan 3 tidak dipilih, sehingga $x_2 = x_3 = 0$. Nilai z untuk submasalah 1 adalah $z = 5\left(\frac{2}{3}\right) + 8(0) + 3(0) + 7(1) = \frac{31}{3}$. Karena x_1 masih bukan berupa bilangan bulat, maka kita akan cabangkan x_1.



Gambar 3. Pencabangan submasalah 1

AKSIOMA: Jurnal Matematika dan Pendidikan Matematika
Vol. 13, No. 1 April 2022
e-ISSN 2579-7646

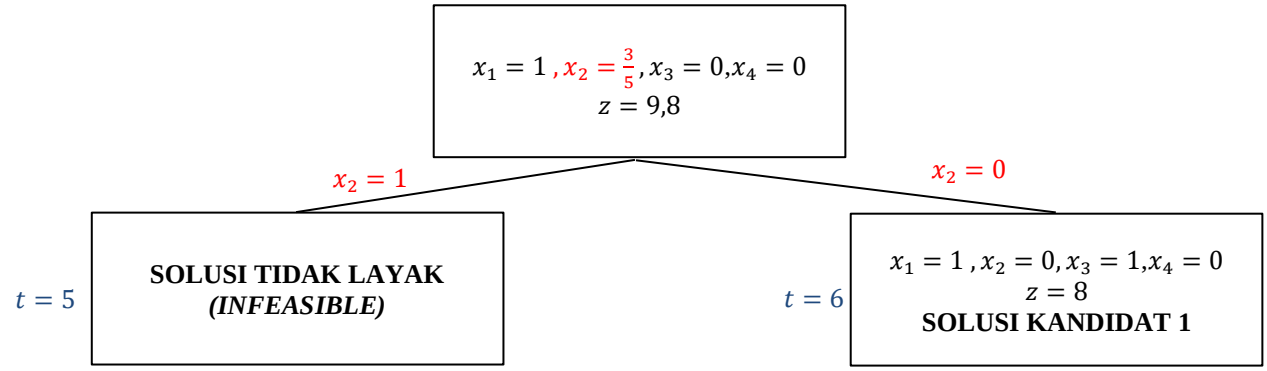
Submasalah 2 : $x_1 = 1$
Karena x_1 tetap, maka $x_1 = 1$ kita pilih terlebih dahulu. Maka item tersisa adalah $6 - 3 = 3$. Selanjutnya, pilih item 4. Karena koefisien x_4 lebih dari 3, maka $x_4 = \frac{3}{4}$. Sehingga diperoleh nilai $z = \frac{41}{4} = 10{,}25$. Karena nilai x_4 masih belum merupakan bilangan bulat, maka kita cabangkan nilai x_4.



Gambar 4. Pencabangan submasalah 2

Submasalah 3 : $x_1 = 1,\ x_4 = 1$
Karena x_1 dan x_4 tetap, maka kita pilih item 4 terlebih dahulu (berdasarkan peringkat). Jika item 4 dipilih, maka $x_4 = 1$ dan tersisa $6 - 4 = 2$ item. Selanjutnya, pilih item 1 maka haruslah $x_1 = 1$. Karena item tersisa 2, sedangkan koefisien kendala dari x_1 adalah 3, maka untuk $t = 2$ diperoleh solusi yang tidak layak.

Submasalah 4 : $x_1 = 1, x_4 = 0$
Karena x_1 dan x_4 tetap, maka kita pilih item 4 terlebih dahulu (berdasarkan peringkat). Tetapi, karena $x_4 = 0$ maka kita katakan "x_4 tidak dipilih". Sehingga masih tersisa 6. Selanjutnya, pilih $x_1 = 1$ maka tersisa $6 - 3 = 3$ item. Selanjutnya, pilih item 2 maka $x_2 = \frac{3}{5}$. Maka diperoleh nilai $z = 9{,}8$. Karena nilai x_2 masih belum merupakan bilangan bulat, maka kita cabangkan x_2.



Gambar 5. Pencabangan submasalah 4

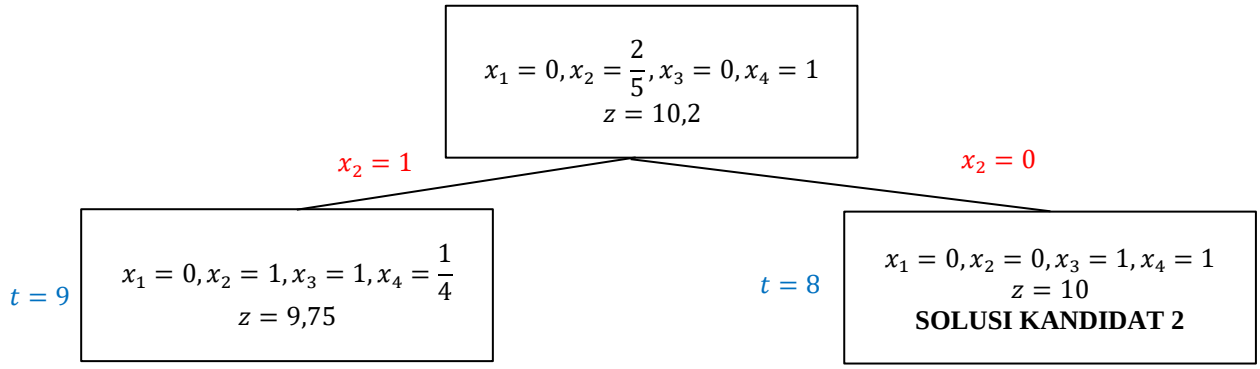
Submasalah 5 : $x_1 = 1, x_2 = 1, x_4 = 0$.
Karena x_1, x_2 dan x_4 tetap, maka terlebih dahulu kita pilih x_4. Karena $x_4 = 0$, maka x_4 tidak dipilih. Sehingga masih tersisa 6 item. Selanjutnya, pilih item 1. Maka $x_1 = 1$ dan tersisa $6 - 3 = 3$ item. Selanjutnya, pilih item 2. Maka haruslah $x_2 = 1$. Namun, karena item tersisa 3 sedangkan koefisien x_2 bernilai 5, maka submasalah 5 memiliki solusi yang tidak layak.

Submasalah 6 : $x_1 = 1, x_2 = 0, x_4 = 0$.
Karena x_1, x_2 dan x_4 tetap, maka terlebih dahulu kita pilih x_4. Karena $x_4 = 0$, maka x_4 tidak dipilih. Sehingga masih tersisa 6 item. Selanjutnya, pilih item 1. Maka $x_1 = 1$ dan tersisa $6 - 3 = 3$ item. Selanjutnya, item 2 tidak dipilih. Maka $x_2 = 0$ dan masih tersisa 3 item. Selanjutnya pilih item 3, maka tersisa 1 item. Maka diperoleh nilai $z = 8$. Karena nilai x_1, x_2, x_3, dan x_4 sudah 0 atau 1, maka $z = 8$ adalah **solusi kandidat 1**.

Sekarang, perhatikan kembali Gambar 3. Kotak yang belum dikerjakan tersebut kita kerjakan untuk melihat apakah ada nilai z yang lebih baik.

$t = 7$ $\quad x_1 = 0, x_2 = \frac{2}{5}, x_3 = 0, x_4 = 1$, $z = 10{,}2$

Submasalah 7 : $x_1 = 0$.
Karena $x_1 = 0$, maka item yang tersisa 6. Selanjutnya, pilih item 4 yaitu $x_4 = 1$ maka tersisa $6 - 4 = 2$ item. Selanjutnya, pilih item 2 yaitu $x_2 = \frac{2}{5}$. Sehingga diperoleh nilai $z = 10{,}2$. Karena nilai $z = 10{,}2$ lebih baik dari solusi kandidat 1, maka kita harus mencabangkan lagi x_2.



Gambar 6. Pencabangan submasalah 7

Submasalah 8 : $x_1 = 0, x_2 = 0$.
Dengan cara yang sama seperti menyelesaikan submasalah sebelumnya, maka diperoleh $x_1 = 0, x_2 = 0, x_3 = 1$, dan $x_4 = 1$ dan nilai $z = 10$. Maka, nilai $z = 10$ merupakan solusi kandidat 2.

Submasalah 9: $x_1 = 0, x_2 = 1$.
Diperoleh $x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 1$ dan nilai $z = 9{,}75$. Karena $z = 9{,}75$ tidak lebih baik dari $z = 10$, maka proses kita hentikan.

Jika dibandingkan solusi kandidat 1 dan solusi kandidat 2, maka nilai optimal pada solusi kandidat 2 lebih baik daripada nilai optimal pada solusi kandidat 1. Hal ini karena pada kasus ini merupakan kasus maksimasi sehingga haruslah dipilih nilai z yang paling besar dan $x_i (i = 1,2,3,4)$ bilangan bulat (Winston, 2004). Dengan demikian, solusi untuk masalah *knapsack* tersebut adalah $x_1 = 0, x_2 = 0, x_3 = 1$ dan $x_4 = 1$ dengan nilai $z = 10$.

Pembaca dapat mencoba dengan cara yang sama yaitu mengerjakan ketika $x_1 = 0$ terlebih dahulu.
Selanjutnya, kita akan selesaikan masalah *knapsack* ini menggunakan algoritma enumerasi implisit.

Algoritma Enumerasi Implisit

Pada bagian ini, kita akan menyelesaikan masalah *knapsack* (1) dengan menggunakan algoritma enumerasi implisit. Langkah pada enumerasi implisit kita beri istilah "node".

Node 1

Variabel bebas : x_1, x_2, x_3, x_4

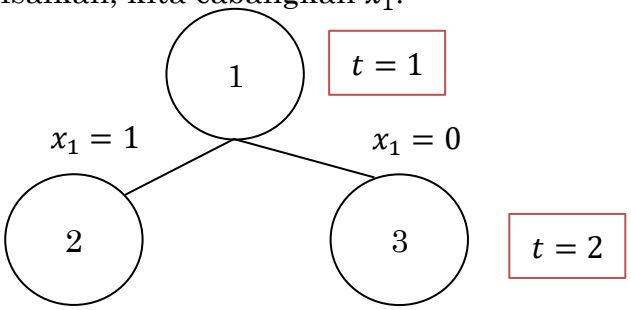
Variabel tetap : -

Best completion

$x_1 = x_2 = x_3 = x_4 = 1$

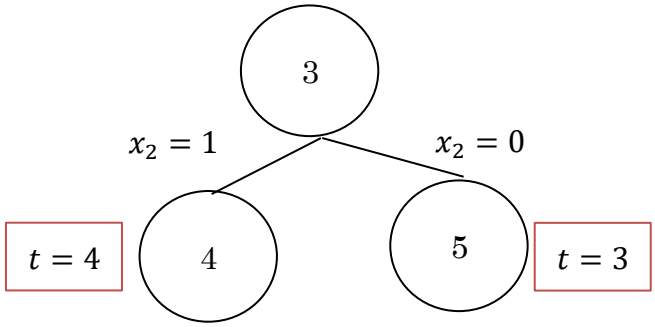
Masukkan nilai diatas pada kendala (1), diperoleh $14 \leq 6$. Sehingga *best completion* tersebut tidak layak karena melanggar kendala yang diberikan.

Sekarang, kita akan cek kelayakan sistem tersebut. Karena tipe kendala $\leq$ dan tanda di tiap koefisien pada kendala (1) seluruhnya positif, maka nilai $x_1 = x_2 = x_3 = x_4 = 0$. Sehingga, jika nilai tersebut dimasukkan pada kendala (1), diperoleh $0 \leq 6$. Maka node 1 punya solusi layak. Maka, cabangkan pada variabel bebasnya. Misalkan, kita cabangkan x_1.



Node 3
Variabel bebas : x_2, x_3, x_4
Variabel tetap : x_1
Best completion
$x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 1.$
Masukkan nilai diatas pada kendala (1), diperoleh $11 \leq 6$. Sehingga *best completion* tersebut tidak layak karena melanggar kendala yang diberikan.

Sekarang, kita akan cek kelayakan sistem tersebut. Karena tipe kendala $\leq$ dan tanda di tiap koefisien pada kendala (1) seluruhnya positif dan x_1 tetap, maka nilai $x_1 = 0,\ x_2 = x_3 = x_4 = 0$. Sehingga, jika nilai tersebut dimasukkan pada kendala (1), diperoleh $0 \leq 6$. Maka node 1 punya solusi layak. Maka, cabangkan pada variabel bebasnya. Misalkan, kita cabangkan x_2.

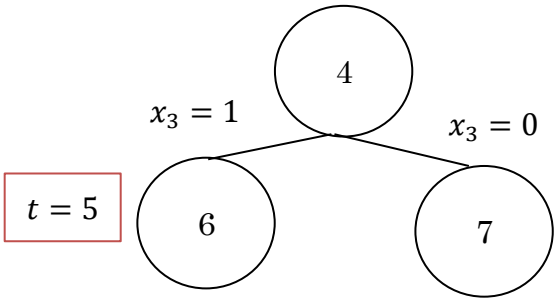


Node 5
Variabel bebas : x_3, x_4
Variabel tetap : x_1, x_2
Best completion
$x_1 = 0, x_2 = 0, x_3 = 1, x_4 = 1.$
Masukkan nilai diatas pada kendala (1), diperoleh $6 \leq 6$. Sehingga *best completion* tersebut **layak**. Diperoleh solusi kandidat 1 $x_1 = 0,\ x_2 = 0, x_3 = 1, x_4 = 1$ dengan nilai $z = 10$.

Walaupun solusi sudah layak, tapi kita masih memiliki beberapa node yang belum dikerjakan. Untuk itu, kita harus mengerjakan node tersebut guna meninjau apakah ada solusi yang lebih baik dari solusi kandidat 1.

Node 4
Variabel bebas : x_3, x_4
Variabel tetap : x_1, x_2
Best completion
$x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 1.$
Masukkan nilai diatas pada kendala (1), diperoleh $11 \leq 6$. Sehingga *best completion* tersebut tidak layak. Sebelum dilanjutkan, nilai z dari node 4 adalah $z = 18$ yang menandakan bahwa node 4 lebih baik dari node 5. Setelah dicek kelayakan, yaitu di titik $x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 0$, maka diperoleh

$5 \leq 6$ yang menandakan bahwa node 4 memiliki solusi layak. Maka kita cabangkan variabel bebas dari node 4. Misalkan x_3.



Node 6
Variabel bebas : x_4
Variabel tetap : x_1, x_2, x_3
Best completion
$x_1 = 0, x_2 = 1, x_3 = 1, x_4 = 1$
Dapat dicek bahwa *best completion* adalah solusi yang tidak layak.
Selanjutnya, setelah dicek kelayakan diperoleh $7 \leq 6$ yang menandakan bahwa node 6 memiliki solusi **tidak layak**.

Node 7
Variabel bebas : x_4
Variabel tetap : x_1, x_2, x_3
Best completion
$x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 1$
Dapat dicek bahwa *best completion* adalah solusi yang tidak layak. Sebelum dilanjut cek kelayakan, nilai z dari node 7 adalah $z = 15$ yang lebih baik dari solusi kandidat pada **Node 5**. Selanjutnya, setelah dicek kelayakan diperoleh $5 \leq 6$ yang menandakan bahwa node 7 memiliki solusi layak.

Begitu pula seterusnya untuk node-node lainnya. Pada akhirnya, diperoleh node 5 merupakan solusi yang paling baik karena kasus ini merupakan kasus maksimasi, sehingga haruslah dipilih nilai z terbesar (Winston, 2004). Nilai z terbesar dengan nilai $x_i (i = 1,2,3,4)$ merupakan bilangan bulat terjadi di node 5. Sehingga, solusi untuk masalah *knapsack* ini adalah $x_1 = 0, x_2 = 0, x_3 = 1$, dan $x_4 = 1$ dengan nilai $z = 10$.

D. Simpulan

Berdasarkan pemaparan diatas, dapat dilihat bahwa kita memperoleh hasil yang sama baik menggunakan metode *branch and bound* maupun menggunakan metode enumerasi implisit. Berdasarkan proses yang kita lakukan, kita dapat melihat bahwa metode *branch and bound* adalah metode

yang lebih efektif daripada metode enumerasi implisit. Pada metode *branch and bound*, kita tidak memerlukan Tabel 1 untuk mencari solusi dari masalah *knapsack*. Sedangkan, untuk metode enumerasi implisit, kita memerlukan Tabel 1 untuk mengecek kelayakannya dan sangat mungkin terjadi kesalahan. Saran untuk penelitian selanjutnya, peneliti dapat menggunakan metode lain seperti algoritma greedy untuk dilakukan perbandingan manakah metode yang paling efektif.

E. Daftar Pustaka

Basriati, S. (2018). Integer Linear Programming Dengan Pendekatan Metode Cutting Plane Dan Branch And Bound Untuk Optimasi Produk Tahu. *Jurnal Sains Matematika Dan Statistika*, ***4***(2), 95–104.

Devita, R. N., & Wibawa, A. P. (2020). Teknik Teknik Optimasi Knapsack Problem. *Sains, Aplikasi, Komputasi Dan Teknologi Informasi*, ***2***(1), 35–40.

Hiller, F. S., & Lieberman, G. J. (2015). *Introduction to Operation Research: Tenth Edition*. Mc Graw Hill Education.

Lageweg, B. J., Lenstra, J. K., & Rinnooy Kan, H. G. (1977). Job-Shop Scheduling by Implicit Enumeration. *Management Science*, ***24***(4), 441–450.

Lesmana, N. I. (2016). Penjadwalan Produksi Untuk Meminimalkan Waktu Produksi Dengan Menggunakan Metode Branch and Bound. *Jurnal Teknik Industri*, ***17***(1), 42–50.

Ligget, R. S. (1973). The Application of An Implicit Enumeration Algorithm To The School Desegregation Problem. *Management Science*, ***20***(2), 159–168.

Mazda, C. N., & Kurniawati, D. A. (2020). Branch and Bound Method to Overcome Delay Delivery Order in Flow Shop Scheduling Problem. *International Conference on Industrial and Manufacturing Engineering*.

Rachmawati, Dian & Candra, Ade. (2013). Implementasi Algoritma Greedy Untuk Menyelesaikan Masalah Knapsack Problem. Jurnal SAINTIKOM, ***12***(3). 185 – 192.

Sabaruddin, Raja. (2016). Solusi Optimum Minmax 0/1 Knapsack Menggunakan Algoritma Greedy. Jurnal Evolusi, ***4***(2). 76 – 82.

Winston, W.L. (2004). Operation Research: Application and Algorithms. Belmont, USA: BROOCKS/COLE.