

Pemanfaatan Algoritma Branch and Bound Untuk Menentukan Menu Diet

IF2211 Strategi Algoritma

Alvin Limassa 13516039

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

alvin.limassa@gmail.com

Abstrak— Obesitas atau yang biasa dikenal dengan kelebihan berat badan adalah kondisi tubuh pada saat lemak telah melebihi batas dan dapat menyebabkan penyakit jantung, diabetes, dan tekanan darah tinggi. Indonesia negara dengan penduduk terbanyak ke-4 di dunia memiliki sekitar 20 persen penduduk yang mengalami obesitas dan sangat rentan terserang penyakit. Dikarenakan jumlah penduduk di Indonesia yang mengalami obesitas, salah satu solusi untuk mengurangi jumlah tersebut adalah melakukan diet agar dapat mengurangi berat badan sehingga dapat terhindar dari berbagai penyakit yang dibawa oleh obesitas. Menu diet dapat ditentukan dengan memanfaatkan algoritma *branch and bound* sehingga dapat memilih menu makanan sebanyak banyaknya dengan jumlah kalori tertentu. Jumlah kalori yang dibatasi dapat membantu dalam proses pengurangan berat badan.

Kata kunci – Diet, obesitas, kalori, dan algoritma *branch and bound*

I. PENDAHULUAN

Indonesia dengan jumlah penduduk terbanyak ke-4 di dunia memiliki 20 persen (tahun 2017) dari total jumlah penduduk yang mengalami obesitas, meningkat jauh dari tahun sebelumnya yaitu sekitar 15 persen.¹ Obesitas merupakan kondisi tubuh pada saat lemak didalam tubuh telah melebihi batas dan dapat menimbulkan penyakit seperti penyakit jantung, diabetes, dan tekanan darah tinggi. Obesitas umumnya diukur dengan perhitungan Body Mass Indeks (BMI). BMI yang tergolong ideal adalah BMI yang berada pada rentang 18-25, yang berada diatas 25 dikatakan memiliki berat badan yang berlebih dan apabila telah menyentuh angka diatas 30 maka sudah dianggap obes/ obesitas. Selain di Indonesia permasalahan obesitas ini juga menimpa banyak negara di dunia dengan total orang di dunia yang mengalami obesitas sekitar 775 juta penduduk dewasa pada juli tahun 2017.²

Berdasarkan informasi yang disampaikan oleh Manajer Riset dan Pengabdian Masyarakat Fakultas Kedokteran Universitas Indonesia (FK-UI), Dr. Budi Wiweko, salah satu

penyebab banyaknya angka obesitas di Indonesia dikarenakan pola konsumsi makanan dan minuman yang mengandung banyak gula dan banyak tersedia di pasaran. Pola ini juga yang terjadi pada masyarakat Amerika pada 20 tahun yang lalu, pada saat itu di Amerika sedang sangat populer minuman bersoda yang mengandung banyak gula.³ Selain kadar gula yang berlebih pada tubuh, obesitas juga dapat disebabkan oleh kadar kalori yang berlebih pada tubuh dan tidak digunakan sehingga menumpuk dan menyebabkan obesitas.

Obesitas juga merupakan salah satu penyebab kematian yang sebenarnya paling mudah untuk dicegah dibandingkan dengan penyakit-penyakit lain, akan tetapi stigma bahwa kegemukan merupakan simbol dari kesuburan yang terdapat di beberapa negara menyebabkan sulitnya untuk mengurangi tingkat obesitas. Selain berdampak pada kesehatan individu, obesitas juga berdampak pada sektor ekonomi, hal ini disebabkan biaya yang dikeluarkan untuk melakukan pengobatan yang memiliki hubungan dengan kegemukan sekitar 190.2 miliar dolar Amerika (20 % dari total biaya kesehatan yang dikeluarkan penduduk Amerika) berdasarkan data pada tahun 2005.⁴ Riset juga menunjukkan bahwa orang yang mengalami kelebihan berat badan akan lebih sulit untuk mendapatkan promosi jabatan dalam pekerjaan serta memiliki gaji yang lebih rendah dibandingkan dengan orang yang tidak mengalami obesitas dan memiliki jabatan yang sama.⁵

Salah satu langkah yang dapat ditempuh untuk mengurangi berat badan adalah memanfaatkan program diet yang dikombinasikan dengan olahraga yang cukup berdasarkan jumlah kalori yang dimakan pada hari tersebut. Penentuan jenis makanan untuk diet berdasarkan kalori sangatlah penting dan krusial untuk melakukan penurunan berat badan. Penentuan tersebut dapat ditentukan dengan bantuan algoritma *branch and bound*. Algoritma ini akan membantu orang untuk memilih sebanyak-banyaknya makanan (agar tetap kenyang) dengan jumlah kalori yang paling minimal sehingga pada saat melakukan diet, orang tersebut tidak akan mengalami

³ <https://www.cnnindonesia.com/gaya-hidup/20170302223030-255-197500/semakin-banyak-orang-indonesia-alami-obesitas>

⁴ Cawley, J. 2012. The medical care costs of obesity: An instrumental variables approach. Journal of Health Economics. Cornell University : United States.

⁵ Puhl R, Brownell KD. 2001. Bias, discrimination, and obesity. Yale University : New Haven.

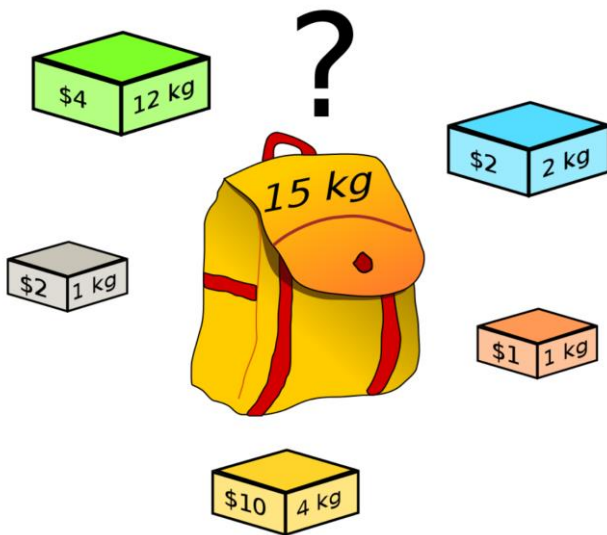
¹ <https://www.cnnindonesia.com/gaya-hidup/20170302223030-255-197500/semakin-banyak-orang-indonesia-alami-obesitas>

² <https://renewbiatrics.com/obesity-rank-by-countries/>

kelaparan yang sangat menyiksa akan tetapi jumlah kalori yang masuk dalam tubuh dapat dikurangi.

II. LANDASAN TEORI

A. Knapsack Problem



Gambar 1. Knapsack problem

(sumber : <https://en.wikipedia.org/wiki/File:Knapsack.svg> diakses pada tanggal 7/5/2018 jam 9.39)

Knapsack problem merupakan salah satu persoalan komputasional klasik berasal dari tahun 1897 yang diperkenalkan oleh seorang matematikawan Tobias Dantzig. Permasalahan ini memiliki tujuan untuk melakukan optimisasi terhadap nilai keuntungan dengan batasan-batasan tertentu. Salah satu persoalan klasik *knapsack problem* adalah untuk memasukkan sebanyak-banyaknya objek yang memiliki nilai jual tinggi dengan syarat bahwa barang yang diambil tidak melebihi kapasitas kantong yang digunakan.

Knapsack problem terbagi atas beberapa permasalahan utama yaitu :

1. 0-1 knapsack problem

0-1 *knapsack problem* adalah permasalahan *knapsack* yang membatasi jumlah objek x_i dengan angka 1 atau 0. Persoalan ini memiliki rumus sebagai berikut :

Jumlah Maksimal :

$$\sum_{i=1}^n v_i x_i$$

Syarat :

$$\sum_{i=1}^n w_i x_i \leq W \text{ and } x_i \in \{0, 1\}.$$

Keterangan :

x_i : objek ke i (ada atau tidak 1/0)
 v_i : nilai objek ke i
 w_i : berat objek ke i

W : berat total yang dapat ditampung *knapsack*

2. Bounded knapsack problem

Bounded knapsack problem adalah permasalahan *knapsack* yang membatasi hanya ada satu objek untuk setiap item akan tetapi boleh memiliki jumlah duplikatnya dengan nilai non negatif yang telah ditentukan. Persoalan ini memiliki rumus sebagai berikut :

Jumlah Maksimal :

$$\sum_{i=1}^n v_i x_i$$

Syarat :

$$\sum_{i=1}^n w_i x_i \leq W \text{ and } 0 \leq x_i \leq c$$

Keterangan :

x_i : jumlah objek ke i
 v_i : nilai objek ke i
 w_i : berat objek ke i
 W : berat total yang dapat ditampung *knapsack*

3. Unbounded knapsack problem

Unbounded knapsack problem adalah permasalahan *knapsack* yang tidak memiliki batasan atas untuk jumlah objek yang dapat dimiliki sehingga nilai dari suatu item dapat dimiliki sebanyak-banyaknya apabila objek tersebut memiliki nilai keuntungan yang terbesar. Persoalan ini memiliki rumus sebagai berikut:

Jumlah Maksimal :

$$\sum_{i=1}^n v_i x_i$$

Syarat :

$$\sum_{i=1}^n w_i x_i \leq W \text{ and } x_i \geq 0$$

Keterangan : variabel yang digunakan pengertiannya sama dengan *Bounded knapsack problem*

Knapsack problem dapat memiliki beberapa variasi tambahan yaitu :

1. Multi-objective knapsack problem

Multi-objective knapsack problem yaitu variasi dari *knapsack problem* yang memiliki lebih dari satu tujuan, tidak hanya memaksimalkan hasil atau value yang diharapkan tetapi memiliki beberapa tujuan sekaligus. Contohnya : kapal nelayan hanya dapat

menampung 10 ton ikan, pemilik kapal harus menentukan jumlah awak kapal yang harus dibawa. permasalahan disini adalah batasan berat awak kapal yang harus di bawa dan jumlah bayaran yang harus dibayar dengan keinginan memaksimalkan jumlah tangkapan dan membawa sebanyak awak kapal yang mungkin untuk mempercepat penangkapan ikan.

2. *Multi-dimension knapsack problem*

Multi-dimension knapsack problem adalah permasalahan objek *knapsack* yang memiliki lebih dari satu dimensi. Contohnya : yang diukur pada saat mengambil barang untuk dimasukkan ke dalam *knapsack* bukanlah berat melainkan ukurannya, ukuran barang memiliki dimensi yaitu panjang, lebar dan tinggi.

3. *Multiple knapsack problem*

Multiple knapsack problem adalah permasalahan *knapsack* yang memiliki beberapa beberapa tempat penampungan untuk menyimpan objek dengan keinginan meminimalkan jumlah tempat penampungan yang digunakan.

4. *Quadratic knapsack problem*

Quadratic knapsack problem adalah persoalan untuk memaksimalkan tujuan/ objektif yang bersifat kuadratik menjadi sebuah permasalahan yang memiliki batasan dalam bentuk biner maupun linear

5. *Subset-sum problem*

Subset-sum problem adalah persoalan *knapsack* yang mirip dengan persoalan 0-1 *knapsack problem* dengan keunikan bahwa semua berat objek akan sama dengan nilai keuntungan/ value dari objek tersebut. Permasalahan ini sangat banyak digunakan dalam dunia kriptografi, salah satu contohnya adalah penerapan *knapsack* dalam permasalahan Karp's 21 NP-complete.

Metode Penyelesaian permasalahan *knapsack* (terutama untuk jenis 0-1 *knapsack problem*) terdiri atas beberapa algoritma yaitu :

1. *Brute force*

Metode ini akan melakukan pencocokkan semua kemungkinan yang terjadi satu persatu dengan kompleksitas $O(2^n)$

2. *Greedy*

Metode ini akan melakukan pemilihan solusi optimum lokal pada setiap kali tahapan sehingga diharapkan pada hasil akhir akan memperoleh solusi optimum global. Akan tetapi penyelesaian 0-1 *knapsack problem* tidak menjamin akan memberikan solusi optimum pada setiap kali percobaan karena keterbatasan yang tidak menjamin bahwa total dari solusi optimum lokal adalah optimum global. Kompleksitas dari algoritma ini adalah $O(n^2)$

3. *Branch and Bound*

Metode ini adalah metode yang melakukan pemilihan pada setiap tahap dengan tambahan nilai taksiran menuju simpul tujuan.

4. *Dynamic programming*

Metode ini adalah metode yang menguraikan solusi menjadi beberapa langkah sehingga solusi tersebut dapat dipandang sebagai serangkaian keputusan yang saling terkait. Solusi akhir merupakan solusi yang dibangun dari solusi sebelumnya.

B. *Branch and Bound*

Branch and Bound merupakan salah satu strategi algoritma yang sering digunakan untuk berbagai keperluan. Metode ini umumnya digunakan untuk menyelesaikan berbagai persoalan yang berhubungan dengan struktur diskrit, kombinatorial, dan persoalan optimisasi. Algoritma ini pertama dikemukakan oleh A. H. Land dan A. G. Doig pada tahun 1960 untuk menyelesaikan berbagai persoalan dalam bidang pemrograman diskrit. Metode ini mengandung enumerasi sistematis solusi kandidat yang dapat digambarkan sebagai suatu pohon yang memiliki himpunan solusi sebagai kumpulan solusi. Metode ini akan melakukan enumerasi terhadap kandidat solusi dengan melakukan pengecekan terhadap batas atas dan batas bawah solusi optimal dan akan langsung membuat solusi yang dianggap tidak memenuhi persyaratan.

Tujuan utama dari algoritma *branch and bound* adalah untuk menemukan nilai x sehingga fungsi $F(x)$ dapat diketahui nilai maksimal atau nilai minimalnya. Tujuan utama tersebut dapat dipenuhi dengan menggunakan dua prinsip utama yaitu melakukan rekursif untuk memecah semesta kedalam bagian-bagian yang lebih kecil yang dikenal dengan istilah percabangan dan yang kedua adalah melakukan pencarian fungsi batasan atau estimasi minimum kemudian melakukan percabangan terhadap simpul tersebut. Tanpa prinsip yang kedua *branch and bound* akan sama saja dengan metode *brute force* atau *exhaustive search*.

Secara umum metode/ algoritma *branch and bound* dapat ditulis tahapan-tahapannya sebagai berikut :

1. Carilah solusi x_0 yang dapat mengoptimalkan persoalan dengan menggunakan metode *heuristic*. Nilai tersebut akan diasumsikan sebagai nilai terbaik yang telah didapatkan sejauh ini (disimpan pada variabel B).
2. Mempersiapkan suatu antrian untuk menyimpan nilai-nilai subbagian solusi yang belum dilakukan pemrosesan.
3. Lakukan pengulangan berikut hingga antrian tersebut kosong :
 - a. Ambil antrian pertama sebagai N dari tabel antrian
 - b. Jika nilai N tersebut merepresentasi kandidat solusi x dan $f(x) < B$, maka x adalah solusi terbaik saat ini kemudian B di isi dengan $f(x)$
 - c. Else, cabang yang menempel pada simpul yang diambil sebelumnya akan dilakukan :

- i. Jika fungsi estimasi $g(N_i) > B$, maka nilai tersebut tidak akan pernah mengarah ke solusi sehingga dibuang dan tidak dilanjutkan pengerjaannya lagi untuk simpul tersebut.
- ii. Else, simpan N_i pada antrian untuk diproses lagi.

III. JENIS MAKANAN YANG COCOK UNTUK DIET

Makanan terdiri atas berbagai jenis mulai dari makanan yang sehat, makanan bergizi, makanan tinggi protein, tinggi lemak, karbohidrat kompleks, dan yang lainnya. Dari berbagai jenis makanan tersebut terdapat beberapa syarat yang harus dipenuhi oleh orang yang ingin melakukan diet kalori/ pengurangan jumlah kalori yang dimakan untuk menurunkan berat badan. Syarat – syarat tersebut adalah :

1. Rendah energi

Makanan yang menghasilkan energi yang rendah sangat diperlukan dalam diet kalori, hal ini dibutuhkan agar energi yang dimakan tidak mencukupi aktivitas sehari – hari sehingga tubuh akan membakar cadangan makanan yang ada dalam tubuh untuk memenuhi kebutuhan energinya. Pembakaran cadangan energi dalam tubuh akan berdampak pada turunnya berat badan.

2. Tinggi protein

Makanan dengan jumlah protein yang tinggi atau lebih banyak dari biasanya dapat membantu Anda bertahan dari rasa kelaparan. Protein merupakan salah satu kandungan zat dalam makanan yang dapat menunda rasa lapar sehingga tidak sering muncul. Protein yang disarankan dalam menu diet kalori adalah protein dari daging tanpa lemak, ayam tanpa kulit, telur, susu, keju, tempe, tahu, kacang kedelai, dan lain-lain. jenis protein yang harus dihindari adalah protein yang berasal dari jenis makanan yang berlemak tinggi, contohnya adalah kulit ayam, jeroan, santan, dan daging kambing.

3. Rendah lemak

Kebutuhan tubuh akan lemak berkisar 20-25% dari total kalori yang dimakan, hal ini menyebabkan seseorang yang ingin melakukan diet kalori harus mengurangi jumlah makanan yang berlemak, karena menurut penelitian makanan berlemak atau berminyak mengandung banyak kalori. Hal ini menyebabkan jumlah kalori yang dimakan banyak tetapi tidak akan merasakan rasa kenyang karena lemak tersebut. Beberapa jenis sumber lemak yang dianjurkan adalah lemak yang mengandung kadar lemak tidak jenuh yang tinggi. Contoh lemak yang harus dihindari adalah lemak yang bersumber dari minyak kelapa, kelapa, dan santan.

4. Karbohidrat kompleks

Karbohidrat merupakan salah satu kebutuhan utama tubuh untuk beraktivitas. jumlah karbohidrat yang harus dikurangi adalah sekitar 55-65% dari total yang umumnya dikonsumsi. Jenis – jenis karbohidrat yang disarankan adalah karbohidrat yang kompleks seperti nasi merah, ubi, singkong, talas, kentang dan sereal. Jenis yang tidak dianjurkan adalah karbohidrat sederhana yang umumnya ditemukan pada gula.

5. Tingkat Kekenyanan yang diberikan suatu makanan

Setiap makanan memiliki suatu tingkat kekenyangan yang akan diberikan kepada orang yang memakan makanan tersebut. Tingkat kekenyangan ini akan membandingkan waktu (rata-rata) umumnya orang akan lapar lagi setelah memakan suatu jenis makanan tertentu dengan jumlah kalori yang sama.

Beberapa jenis makanan yang rendah kalori dan cocok untuk menu diet terdapat dalam tabel berikut :

No	Jenis Makanan	Kalori (100 gram)	Kalori (250 gram)	Tingkat Kekenyangan
1	Pisang	89	223	2.5
2	Anggur	69	173	4
3	Santan	230	575	1.6
4	Nasi Putih	124	310	2.1
5	Nasi Merah	110	275	2.3
6	Daging Sapi	288	720	3.2
7	Daging Ayam	167	418	3.3
8	Ikan	84	210	3.4
9	Susu	122	305	3.4
10	Kentang	118	295	2.5
11	Brokoli	34	85	4.5
12	Kangkung	20	50	4.5
13	Telur	77	193	2.5
14	Wortel	54	135	3.8
15	Rendang	195	488	3.2

Tabel 1 Tabel kalori makanan dan tingkat kekenyangan

IV. PEMILIHAN JENIS MAKANAN UNTUK DIET

Pemilihan jenis makanan yang akan dimakan pada saat diet sangat penting karena menentukan apakah orang tersebut akan merasa kelaparan pada saat menjalankan diet atau tidak. Sehingga diet yang dijalani terutama diet kalori dapat meminimalkan seseorang mengalami kelaparan pada saat diet

dengan memaksimalkan jumlah makanan yang dimakan dengan syarat makanan tersebut dapat membuat orang yang memakan tidak cepat lapar. Untuk menentukan menu apa saja yang dapat dimakan dengan memaksimalkan jumlah yang dimakan dan tingkat kekenyangan yang diperoleh dapat ditentukan dengan memanfaatkan skema/ strategi algoritma yang dapat menyelesaikan persoalan *knapsack*. Persoalan yang dibahas memiliki kemiripan yang tinggi dengan permasalahan *knapsack* karena sama sama memiliki tujuan untuk memaksimalkan jenis makanan yang dapat diambil dan memiliki batasan yaitu kalori yang dapat dikonsumsi.

Permasalahan *knapsack* sendiri dapat diselesaikan dengan beberapa algoritma, pada bab ini akan dibahas perbandingan penyelesaian masing-masing cara untuk persoalan pemilihan jenis makanan secara singkat dan salah satu dari beberapa metode penyelesaian akan dibahas secara rinci, metode yang akan dibahas secara rinci dan spesifik adalah metode *branch and bound*. Berikut adalah beberapa metode yang dapat digunakan untuk menyelesaikan permasalahan pemilihan jenis makanan :

1. Metode *Bruteforce*

Pada metode *bruteforce* program akan melakukan permutasi setiap kemungkinan yang mungkin yaitu memberikan angka 1 dan 0 pada setiap jenis makanan, angka 1 digunakan untuk mengambil jenis makanan tersebut, angka 0 artinya tidak mengambil jenis makanan tersebut. Setelah menentukan kemungkinan yang dipilih akan dilakukan pemeriksaan apakah nilai tersebut masih memenuhi persyaratan bahwa jumlah kalori kurang dari sama dengan syarat yang diinginkan jika benar maka tingkat kekenyangan yang diperoleh dari data tersebut akan dibandingkan dengan nilai maksimum yang telah didapatkan sebelumnya, apabila nilai tersebut lebih besar dari nilai maksimal sebelumnya maka nilai tersebut akan disimpan dan kombinasi jenis makanannya akan disimpan juga. Lakukan pengulangan ini hingga semua kemungkinan diperiksa.

Sebagai contoh pengambilan jenis makanan berdasarkan tabel 1. Jenis makanan yang diambil yaitu (1,1,0,0,0,0,0,0,0,0,0,0,0) yang menandakan bahwa jenis makanan yang diambil adalah pisang dan anggur, kemudian akan dihitung jumlah kalornya yaitu 395 dan tingkat kekenyangannya 6.5. Untuk kasus pertama kali pengujian maka 6.5 menjadi nilai maksimal, kemudian langkah tersebut akan diulang lagi hingga semua nilai dicoba dan didapatkan nilai maksimal serta jenis-jenis makanan yang menghasilkan nilai maksimal tersebut menjadi hasil pemilihan jenis-jenis makanan yang akan dipilih untuk diet. Dengan syarat bahwa nilai maksimal tidak melanggar batasan yang telah diberikan yaitu kalori kurang dari nilai yang ditetapkan pengguna.

2. Metode *Greedy*

Pada metode *Greedy* program akan memilih satu jenis makanan pada setiap tahapnya dengan harapan bahwa pada saat pemilihan nilai optimum lokal akan menghasilkan nilai optimum global. Metode *Greedy* ini tidak menjamin bahwa hasil yang dihasilkan akan memunculkan nilai maksimum global pada setiap persoalan *knapsack* karena pengambilan nilai optimum lokal belum tentu menghasilkan nilai optimum global. Optimum yang digunakan pada permasalahan ini adalah nilai optimum dari tingkat kekenyangan dibagi dengan jumlah kalori.

Sebagai contoh, pada pengambilan jenis makanan pada tabel 1 dengan syarat kalori tidak lebih dari 500 untuk jenis makanan dengan berat 250 gram. Pada tahap pertama program akan mencari nilai kalori minimal sehingga dapat memaksimalkan tingkat kekenyangan yang diperoleh dengan mengambil sebanyak mungkin makanan yang diperbolehkan asalkan tidak melebihi batasan kalori. Pada langkah pertama ini program akan mengambil kangkung dengan kalori 50 dengan tingkat kekenyangan/kalori yaitu 0.09 (yang tertinggi) , kemudian pada tahap berikutnya program akan mengambil brokoli dengan kalori 85 (0.053), wortel dengan kalori 135(0.028), Anggur 172.5(0.023), dan kemudian Ikan 210(0.016). pada setiap tahap pengambilan program akan melakukan pemeriksaan apakah sudah melebihi batas kalori yang diinginkan oleh pengguna, jika belum melebihi maka pemilihan akan dilanjutkan jika sudah melebihi maka program akan menuliskan hasil sebelum pengambilan terakhir menjadi solusi optimumnya. Pada contoh diatas, untuk 500 kalori maka program akan menghasilkan keluaran jenis-jenis makanan berupa kangkung, brokoli, wortel, anggur dengan jumlah kalori 442.5. ikan tidak termasuk jenis makanan yang diambil karena pada saat mengambil telur jumlah kalori semua jenis makanan yang diambil telah melebihi batas yang ditentukan. Solusi yang dihasilkan dari metode ini tidak menjamin bahwa nilai optimum kalori yang diambil merupakan nilai yang paling optimum.

3. Metode *Dynamic Programming*

Pada metode *dynamic programming* program akan membagi suatu permasalahan besar menjadi beberapa tahapan sehingga suatu permasalahan dapat dipandang sebagai serangkaian keputusan yang saling berkaitan. Pada kasus pemilihan jenis makanan yang cocok untuk diet (bergantung pada kalori) akan digunakan tabel pertama dengan data dari nomor 1 hingga 4 dengan jumlah kalori per 100 gram maksimal 200. Tahapan-tahapan pengerjaan akan digambarkan pada tabel berikut :

Keterangan :

Tabel dibulatkan ke 50 satuan kalori untuk mempersingkat penulisan pada laporan. Untuk perhitungan dalam tabel tetap menggunakan

perhitungan detail akan tetapi penulisannya saja yang dibulatkan ke 50 satuan kalori. Jika program akan dibuat maka harus dilakukan secara detail satu per satu tidak dibulatkan. Nilai batasan yang digunakan adalah jumlah kalori pada makanan dan nilai profit yang akan dimaksimalkan adalah tingkat kekenyangan makanan tersebut agar orang yang melakukan diet tidak merasakan lapar dengan cepat.

Tahap 1 :

$$F_1(y) = \max(f_0(y), 2.5 + f_0(y-89))$$

y	$f_0(y)$	$89 + f_0(y-89)$	$f_1(y)$	Solusi
0	0	$-\infty$	0	(0,0,0,0)
50	0	$-\infty$	0	(0,0,0,0)
100	0	2.5	2.5	(1,0,0,0)
150	0	2.5	2.5	(1,0,0,0)
200	0	2.5	2.5	(1,0,0,0)

Tahap 2 :

$$F_2(y) = \max(f_1(y), 4 + f_1(y-69))$$

y	$f_1(y)$	$4 + f_1(y-69)$	$f_2(y)$	Solusi
0	0	$4 + (-\infty) = -\infty$	0	(0,0,0,0)
50	0	$4 + (-\infty) = -\infty$	0	(0,0,0,0)
100	2.5	$4 + 0 = 4$	4	(1,0,0,0)
150	2.5	$4 + 0 = 4$	4	(1,0,0,0)
200	2.5	$4 + 2.5 = 6.5$	6.5	(1,1,0,0)

Tahap 3 :

$$F_3(y) = \max(f_2(y), 1.6 + f_2(y-354))$$

y	$f_2(y)$	$354 + f_2(y-354)$	$f_3(y)$	Solusi
0	0	$1.6 + (-\infty) = -\infty$	0	(0,0,0,0)
50	0	$1.6 + (-\infty) = -\infty$	0	(0,0,0,0)
100	4	$1.6 + (-\infty) = -\infty$	4	(0,1,0,0)
150	4	$1.6 + (-\infty) = -\infty$	4	(0,1,0,0)
200	6.5	$1.6 + (-\infty) = -\infty$	6.5	(1,1,0,0)

Tahap 4 :

$$F_4(y) = \max(f_3(y), 2.1 + f_3(y-124))$$

y	$f_3(y)$	$2.1 + f_3(y-124)$	$f_4(y)$	Solusi
0	0	$2.1 + (-\infty) = -\infty$	0	(0,0,0,0)
50	0	$2.1 + (-\infty) = -\infty$	0	(0,0,0,0)
100	4	$2.1 + (-\infty) = -\infty$	4	(1,0,0,0)
150	4	$2.1 + 0 = 2.1$	4	(0,0,0,1)
200	6.5	$2.1 + 6.5 = 193$	8.6	(0,1,0,1)

Solusi : (0,1,0,1) untuk kalori maksimal 200 dan data yang diambil dari tabel 1 sampai 4. Jenis makanan yang boleh dikonsumsi adalah Anggur dan Nasi Putih dengan total kalori 158.

4. Metode Branch and Bound

Pada metode *Branch and Bound*, program akan melakukan pemilihan pada setiap tahap dengan tambahan nilai taksiran menuju simpul tujuan. Implementasi dilakukan dengan menggunakan bahasa C++.

Pada saat program dijalankan, program akan meminta pengguna untuk memasukkan file data yang akan digunakan dalam penentuan menu diet, data tersebut terdiri atas nama makanan, kalori dalam 250 gram, dan tingkat kekenyangan yang diberikan setiap makanan. Program akan membaca dan mempersiapkan tabel – tabel dan queue yang diperlukan untuk mengeksekusi program. Kemudian pengguna akan diminta untuk memasukkan nilai kalori yang ingin dimakan pada hari tersebut/ dengan kata lain jumlah diet kalori yang akan dilakukan pada hari tersebut.

Setelah mendapat semua data yang diperlukan untuk melakukan eksekusi, program akan menggunakan beban/ *weight* yaitu kalori dengan batasan tertentu dan *profit*/ keuntungan berupa jumlah dari tingkat kekenyangan yang diperoleh dari semua makanan.

Pada tahapan awal program akan membuat suatu nilai dummy/ nilai fiktif untuk memulai pengerjaan dari simpul akar yang paling atas, pada setiap tahap akan diperiksa apakah simpul sudah mencapai titik terakhir ataukah belum dan apakah sudah melanggar dari batasan – batasan yang diberikan. Jika tidak makan akan diperiksa apakah penambahan nilai pada suatu simpul melebihi dari nilai maksimal yang telah didapatkan apabila lebih besar makan tahapan-tahapan simpul akan disimpan sebagai nilai optimum sementara pada variabel global. Program selanjutnya akan menentukan apakah ada simpul-simpul lain yang mengarah ke tujuan dan memiliki nilai maksimal yang lebih besar dari nilai yang ada serang tetapi tidak melanggar batasan-batasan yang ada. Jika ada maka semua simpul-simpul tersebut akan disimpan pada suatu antrian.

Program akan melakukan tahapan sebelumnya yaitu pengecekan berulang kali hingga tabel antrian kosong. Apabila tabel antrian sudah kosong maka nilai maksimum yang terakhir didapatkan merupakan nilai optimum.

Berikut merupakan hasil eksekusi program yang dibuat :

```
D:\Kuliah\Semester 4\Strategi Algoritma\Tugas Laporan>a
Masukkan nama file data makanan = daftarmakanan

Masukkan Jumlah kalori yang anda inginkan = 1500
No 1. Nama Makanan = Kangkung
Kalori = 50
Faktor Kenyang = 45
No 2. Nama Makanan = Brokoli
Kalori = 85
Faktor Kenyang = 45
No 3. Nama Makanan = Wortel
Kalori = 135
Faktor Kenyang = 38
No 4. Nama Makanan = Ikan
Kalori = 210
Faktor Kenyang = 34
No 5. Nama Makanan = Telur
Kalori = 193
Faktor Kenyang = 25
No 6. Nama Makanan = Pisang
Kalori = 223
Faktor Kenyang = 25
No 7. Nama Makanan = Susu
Kalori = 305
Faktor Kenyang = 34
No 8. Nama Makanan = Kentang
Kalori = 295
Faktor Kenyang = 25

Total Kalori = 1496
Jumlah Jenis makanan yang dapat diambil = 8
Maximum derajat kelaparan yang diperoleh dari semua makanan = 27
```

Analisis hasil eksekusi :

Berdasarkan hasil eksekusi yang dilakukan program menghasilkan 8 jenis makan dengan masing-masing jenis makanan seberat 250 gram yang boleh dimakan oleh orang yang melakukan diet kalori dengan jumlah kalori yang dimakan dalam sehari adalah 1500 maksimal. Makanan – makanan yang terpilih adalah kangkung, brokoli, wortel, ikan, telur, pisang, susu, dan kentang dengan total kalori yang dimakan 1496 (masih kurang dari batasan). Total derajat kelaparan yang diperoleh apabila memakan semua makanan tersebut adalah 27.

Pada program digunakan derajat kelaparan sebagai nilai profit atau nilai yang digunakan untuk memaksimalkan permasalahan *knapsack* dikarenakan apabila seseorang cukup kenyang untuk tidak makan maka semakin besar potensi orang tersebut akan sukses dalam diet yang dilakukan, sehingga tidak hanya cukup untuk memilih makanan yang memiliki kalori rendah tetapi juga kalori rendah yang dapat membuat orang tersebut merasa kenyang dalam jangka waktu yang panjang setelah memakan makanan tersebut.

Program ini hanya memaksimalkan tingkat kekenyangan yang diperoleh seseorang pada saat memakan makanan untuk menu diet tersebut. Pada program ini tidak memeriksa apakah menu makan tersebut memiliki kandungan gizi yang cukup, baik untuk kesehatan atau tidak. Program hanya melihat kandungan kalori dan tingkat kekenyangan yang diberikan makanan tersebut.

Program *Branch and bound* untuk menyelesaikan permasalahan ini memiliki kompleksitas $O(2^n)$, yang masih lebih baik jika dibandingkan dengan algoritma *bruteforce*. Hal

ini dikarenakan pada algoritma *Branch and bound* program akan mengeliminasi semua node yang tidak mengarah ke solusi, sedangkan pada *bruteforce* semua simpul tetap dibangkitkan walaupun tidak mengarah ke solusi.

V. UCAPAN TERIMA KASIH

Penulis mengucapkan syukur kepada Tuhan yang Maha Esa, karena dengan rahmat karunia-Nya penulis dapat menyelesaikan makalah ini. Selain itu, ucapan terima kasih juga penulis sampaikan kepada Bapak Dr.Ir. Rinaldi Munir, MT, Dr. Masayu Leylia Khodra, ST., MT, dan Dr. Nur Ulfa Maulidevi, S.T, M.Sc selaku Dosen IF2211 Strategi Algoritma yang telah memberikan ilmu dasar sehingga penulis dapat membuat dan menyelesaikan makalah ini dengan baik.

REFERENCES

- [1] Munir, Rinaldi. Diktat Kuliah IF2211 Strategi Algoritma. Bandung : Program Studi Teknik Informatika ITB
- [2] Cawley, J.2012.The medical care costs of obesity: An instrumental variables approach. Journal of Health Economics. Cornell Universit : United States.
- [3] Puhl R, Brownell KD.2001.Bias, discrimination, and obesity. Yale University : New Haven.
- [4] <https://www.cnnindonesia.com/gaya-hidup/20170302223030-255-197500/semakin-banyak-orang-indonesia-alami-obesitas> diakses pada tanggal 5/5/2017 jam 08.40
- [5] <https://renewbariatrics.com/obesity-rank-by-countries/> diakses pada tanggal 5/5/2017 jam 08.00
- [6] <https://www.cnnindonesia.com/gaya-hidup/20170302223030-255-197500/semakin-banyak-orang-indonesia-alami-obesitas> diakses pada tanggal 12/5/2017 jam 09.00
- [7] <https://helohehat.com/hidup-sehat/nutrisi/diet-rendah-kalori-turun-berat-badan/> diakses pada tanggal 12/5/2017 jam 09.30
- [8] <https://halosehat.com/gizi-nutrisi/panduan-gizi/24-makanan-yang-mengandung-kalori-tinggi> diakses pada tanggal 12/5/2017 jam 09.40
- [9] <https://www.fatsecret.co.id/kalori-gizi/umum/> diakses pada tanggal 12/5/2017 jam 10.00
- [10] <http://nutritiondata.self.com> diakses pada tanggal 13/5/2017 jam 08.00

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, 7 Mei 2018



Alvin Limassa 13516039