

Aplikasi Algoritma *Greedy* dalam Pemenuhan Asupan Kalori Harian

Abda Shaffan Diva 13517021

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

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

13517021@std.stei.itb.ac.id

Abstraksi — Perkembangan keilmuan informatika sangat pesat selama 1 abad terakhir. Salah satu bukti dari bidang strategi algoritma adalah banyaknya algoritma yang dapat digunakan untuk menyelesaikan masalah-masalah yang tidak dapat diselesaikan sebelumnya, atau bisa menyelesaikan suatu masalah dengan lebih baik. Salah satu aplikasi dari strategi algoritma tersebut, yaitu algoritma *greedy* dapat kita gunakan untuk menyelesaikan masalah penentuan asupan kalori sehari-hari berdasarkan usia, berat badan, tinggi badan, dan jenis kelamin.

Kata kunci — algoritma, *greedy*, strategi, kalori

I. PENDAHULUAN

Setiap harinya kita sebagai makhluk hidup pasti memerlukan energi untuk bergerak dan melakukan aktivitas sehari-hari. Sumber energi yang kita gunakan berasal dari makanan dan minuman yang kita konsumsi sehari-hari, karena itu tentunya sangat penting untuk memperhatikan makanan apa saja yang dikonsumsi setiap harinya. Umumnya pada makanan kemasan sudah disajikan kandungan-kandungan yang adalah di dalam makanan tersebut.

ZAT GIZI	JUMLAH
Kalori	188
Lemak (g)	16
Lemak jenuh (g)	3,3
Lemak tak jenuh (g)	12,4
Kolesterol (mg)	0
Sodium (mg)	5
Potasium (mg)	208
Karbohidrat (g)	6
Serat (g)	2
Gula (g)	3
Protein (g)	8
Vitamin E	14%
Niasin	21%
Vitamin B6	30%
Asam Folat	6%
Besi	3%
Kalsium	1%
Magnesium	12%

Contoh kandungan gizi pada kemasan makanan [2]

Salah satu hal yang tidak bisa dipisahkan dari makanan adalah kandungan kalori. Kalori pada makanan dapat memberikan gambaran kepada kita energi yang terkandung di dalam makanan tersebut.

Meskipun tidak semua makanan memiliki detail informasi gizi pada saat dijual, untungnya data tentang kandungan kalori berbagai macam makanan dapat kita akses dengan mudah dengan melakukan pencarian di internet.

Pada makalah ini akan dibahas bagaimana menentukan makanan yang dapat memenuhi kalori harian dengan studi kasus beberapa orang dengan latar belakang yang berbeda-beda

II. TEORI DASAR

1. Definisi Algoritma

Terlepas sadar atau tidak, setiap masalah yang ada di sekitar kita pasti menggunakan algoritma dalam penyelesaian-nya. Dari menjelaskan kegiatan kita dari bangun tidur hingga di posisi kita sekarang, hingga membuat desain bangunan dan jalan.

“Algoritma adalah runtutan langkah-langkah yang digunakan untuk menyelesaikan suatu masalah”.

Dari definisi diatas, maka istilah algoritma sebenarnya kurang tepat jika hanya dikaitkan dengan dunia pemrograman saja, meskipun yang akan dibahas pada artikel ini memang lebih umum ke arah sana ketika membahas teknik perancangan suatu program atau aplikasi. Pada artikel ini, akan dibahas bagaimana suatu algoritma dapat menyelesaikan masalah yang kita hadapi sehari-hari, tentunya dengan terlebih dahulu menerjemahkannya ke dalam kaidah pemrograman yang nantinya dapat diselesaikan dengan membuat sebuah

program yang mengimplementasikan algoritma tersebut.

2. Notasi Algoritma

Saat menerjemahkan sebuah algoritma ke dalam bentuk program, tentunya akan cukup sulit jika langsung menerjemahkan dari bahasa awalnya ke dalam program yang dibuat. Notasi algoritmik adalah sekumpulan notasi yang digunakan untuk memudahkan translasi algoritma ke dalam program, tanpa harus terlalu fokus ke syntax atau *constraint* dari bahasa pemrograman itu sendiri.

Penggunaan notasi algoritmik salah satunya adalah ketika kita belum menentukan bahasa pemrograman yang akan digunakan untuk menyelesaikan suatu algoritma. Selain itu notasi algoritmik dapat memudahkan kita dalam abstraksi permasalahan, dan memecah sebuah masalah yang besar menjadi masalah-masalah yang lebih kecil.

3. Analisis Algoritma

Pada saat kita menyelesaikan suatu masalah, tentunya belum tentu hanya terdapat satu solusi saja ketika kita menyelesaikan suatu masalah. Dari semua solusi yang diajukan untuk menyelesaikan suatu masalah, tentunya terdapat penyelesaian masalah yang baik, dan penyelesaian masalah yang kurang baik. Selain itu metode penyelesaian yang baik untuk suatu masalah tertentu belum baik untuk masalah lainnya.

Untuk menentukan apakah solusi yang diajukan bagus atau tidak pada sebuah program, terdapat dua ukuran yang digunakan, yaitu **kompleksitas waktu**, dan **kompleksitas ruang**, keduanya dinyatakan dalam simbol ***O* besar (*big-Oh*)**. Kompleksitas waktu menunjukkan seberapa lama waktu eksekusi program kita untuk input yang besar sekali, sedangkan kompleksitas ruang, yang menandakan ruang memori yang digunakan pada mesin ketika menjalankan program yang kita gunakan. Baik kompleksitas waktu maupun kompleksitas ruang tidak secara mutlak menentukan kualitas solusi program, tergantung kebutuhan dan *resource* yang dapat digunakan.

4. Strategi Algoritma

Untuk mencapai solusi terbaik yang akan digunakan untuk menyelesaikan suatu masalah, tentunya dibutuhkan strategi untuk mencapainya. Untungnya, sepanjang sejarah keilmuan ilmu komputer (informatika), banyak strategi algoritma yang sudah ditemukan dan dapat digunakan oleh kita untuk menyelesaikan sebuah permasalahan pemrograman, tanpa kita perlu membuat ulang algoritmanya lagi (*reinvent the wheel*). Pada artikel ini, yang akan dibahas hanya dua jenis strategi algoritma, yaitu algoritma *Brute Force*, dan algoritma *Greedy*.

4.1. Algoritma *Brute Force*

Algoritma *brute force* adalah algoritma yang paling sederhana untuk menyelesaikan suatu masalah. Umumnya solusi pertama yang diajukan ketika menyelesaikan suatu masalah adalah solusi yang menggunakan algoritma ini. Kelebihan dari algoritma ini adalah mudah digunakan, serta dapat menyelesaikan semua permasalahan pemrograman. Kelemahan dari algoritma ini adalah umunya boros secara waktu untuk input yang sangat besar.

Contoh penyelesaian masalah menggunakan algoritma *brute force*:

Menghitung a^n ($a > 0$, n adalah bilangan bulat tak-negatif), dengan mengalikan a satu persatu sebanyak n -kali.

4.2. Algoritma *Greedy*

Algoritma *Greedy* adalah jenis strategi algoritma yang mengambil aspek tertentu dari sebuah persoalan dan menjadikan hal tersebut sebagai prioritas dalam memilih solusi yang dianggap terbaik. Meskipun masih termasuk sebagai algoritma *straightforward* (lempeng, sama seperti algoritma *Brute Force*), yang membedakannya adalah algoritma *greedy* tidak perlu mengecek semua kemungkinan solusi pada program.

Skema umum dari algoritma *greedy* adalah sebagai berikut ^[1]:

1. **Himpunan kandidat, C.** Himpunan ini berisi elemen-elemen pembentuk solusi. Contohnya adalah himpunan koin, himpunan pekerjaan yang akan dikerjakan, himpunan simpul di dalam graf, dan lain-lain. Pada setiap langkah, satu buah kandidat diambil dan himpunannya.
2. **Himpunan solusi, S.** Berisi kandidat-kandidat yang terpilih sebagai solusi persoalan. Dengan kata lain, himpunan solusi adalah himpunan bagian dari himpunan kandidat.
3. **Fungsi seleksi** - yaitu fungsi yang pada setiap langkah memilih kandidat yang paling memungkinkan mencapai solusi optimal. Kandidat yang sudah dipilih pada suatu langkah tidak pernah dipertimbangkan lagi pada langkah selanjutnya. Biasanya setiap kandidat, x , di-assign sebuah nilai numerik, dan fungsi seleksi memilih yang

mempunyai nilai terbesar atau memilih yang mempunyai nilai terkecil.

4. **Fungsi kelayakan** - fungsi yang memeriksa apakah suatu kandidat yang telah dipilih dapat memberikan solusi yang layak, yakni kandidat tersebut bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala (constraints) yang ada. Kandidat yang layak dimasukkan ke dalam himpunan solusi, sedangkan kandidat yang tidak layak dibuang dan tidak pernah dipertimbangkan lagi.
5. Fungsi objektif, yaitu fungsi yang memaksimalkan atau meminimumkan nilai solusi (misalnya panjar-rg lintasan, keuntungan, dan lain-lain). Dengan kata lain, persoalan optimasi yang diselesaikan dengan algoritma greedy melibatkan pencarian sebuah himpunan bagian, S, dari himpunan kandidat, C; yang dalam hal ini, S harus memenuhi beberapa kriteria yang ditentukan, yaitu menyatakan suatu solusi dan S dioptimasi oleh fungsi objektif.

5. Diet Seimbang

5.1. Definisi

Menurut Kementerian Kesehatan Republik Indonesia^[2], Diet seimbang adalah pola konsumsi makanan yang mengandung zat gizi dalam jenis dan jumlah yang sesuai dengan kebutuhan tubuh. Zat gizi yang dibutuhkan untuk sehat adalah karbohidrat, protein, lemak, vitamin, dan mineral.

Kebutuhan akan energi dan zat gizi tergantung dari umur, jenis kelamin, berat, dan tinggi badan, iklim serta aktivitas fisik. Angka kecukupan gizi (AKG) adalah konsumsi zat gizi esensial yang harus dipenuhi oleh semua orang sehat. AKG berguna untuk menetapkan pedoman keperluan gizi pada sebuah makanan yang dikemas.

Banyak jenis makanan yang dapat dikombinasikan untuk memenuhi kondisi diet seimbang:

- 5.1.1. Untuk karbohidrat, salah satu sumber yang tersedia sehari-hari adalah gula, yang sederhana seperti gula pasir maupun yang kompleks dan mengandung kelompok gula yang juga

mengandung serat, vitamin, dan mineral, seperti di dalam Nasi, Jagung, Gandum, dan juga Singkong.

- 5.1.2. Garam, mengandung natrium, zat yang dibutuhkan tubuh dalam jumlah kecil untuk mengatur kadar air di dalam tubuh kita.

- 5.1.3. Lemak, dibutuhkan tubuh untuk:

5.1.3.1. Sumber cadangan energi,

5.1.3.2. Media transportasi vitamin A,D,E, dan vitamin K,

5.1.3.3. Membantu menekan rasa lapar dengan memperlambat pengosongan pada lambung,

5.1.3.4. Extra: menambah cita rasa pada makanan yang kita makan.

5.2. Kebutuhan Kalori Makanan

Setiap jenis zat makanan yang disebutkan diatas mengandung kalori yang berbeda-beda. Agar badan kita tetap memiliki energi yang cukup untuk melakukan aktivitas, maka ada angka kebutuhan kalori yang harus dipenuhi oleh kita setiap hari.

Kebutuhan kalori untuk seseorang dapat ditentukan dengan langkah-langkah sebagai berikut^[1]:

1. Ketahui tinggi badan (TB) anda dalam satuan cm (centimeter).
2. Ketahui berat badan (BB) anda dalam satuan kg (kilogram).
3. Hitung berat badan ideal (BBI):

$$BBI = (TB-100) - (0.1*(TB-100))$$

4. Hitung kebutuhan kalori basal(KKB):

Pria: 30kkal x BBI

Wanita: 25kkal x BBI

5. Hitung kebutuhan kalori total (KKT):

$$KKT = KKB + \% \text{Aktivitas Fisik} \times KKB - \% \text{Faktor Koreksi} \times KKB.$$

No	Rentang Umur	Faktor Koreksi
1	kurang dari 40 tahun	0
2	40 - 59 tahun	5%
3	60-69 tahun	10%
4	70 tahun keatas	20%

Tabel faktor koreksi berdasarkan usia ^[2]

No	Aktivitas	Persentase aktivitas fisik	Jenis Aktivitas
1	Membaca	10%	Ringan
2	Menyetir	10%	Ringan
3	Berjalan	20%	Ringan
4	Menyapu	20%	Sedang
5	Jalan Cepat	30%	Sedang
6	Bersepeda	30%	Sedang
7	Aerobik	40%	Berat
8	Mendaki	40%	Berat
9	Jogging	40%	Berat

Tabel persentase energi per kegiatan ^[2]

*catatan: Untuk menyederhanakan perhitungan, pada makalah ini, diasumsikan bahwa kegiatan yang disebutkan pada tabel diatas merupakan kegiatan rutinitas, bukan merupakan kegiatan pada satu waktu saja. Contoh: Aktivitas “berjalan” cukup dihitung satu kali saja.

III. PERHITUNGAN ASUPAN KALORI HARIAN

Misalkan kita akan menghitung data kalori harian selama seminggu yang harus dicukupi dari orang-orang berikut ini:

1. Wati adalah seorang pembantu rumah berusia 42 tahun, dengan tinggi badan 165 cm yang pekerjaan sehari-harinya adalah menyapu pekarangan rumah.

Setiap harinya Wati menghabiskan sejumlah 20000 untuk konsumsi makanan.

2. Ali adalah seorang *tour guide* berusia 32 tahun dengan tinggi badan 181 cm yang kesehariannya adalah menunjukkan rute bagi para pendaki gunung. Setiap harinya Ali menghabiskan sejumlah 50000 untuk konsumsi makanan.
3. Suleman adalah seorang bapak paruh baya berusia 70 tahun dengan tinggi badan 168cm. Dia rutin melakukan olahraga pagi, membaca buku, dan mengantarkan serta menjemput cucunya sepulang sekolah setiap hari. Setiap harinya Suleman menghabiskan sejumlah 35000 untuk konsumsi makanan.

Berikut merupakan data jenis makanan dan kalori yang diambil dari infografis *Kementrian Kesehatan Republik Indonesia* ^[2] :

No	Jenis Makanan (per porsi)	Jumlah makanan pada database
1	Makanan pokok	27
2	Makanan siap saji	21
3	Makanan lauk	39
4	Makanan buah	30
5	Makanan sayur	21

Detail jumlah makanan per tipe yang ada di dalam database penulis, diambil dari ^[2]

*detail semua makanan tidak ditunjukkan agar makalah ini tidak terlalu panjang.

1. Menghitung kebutuhan kalori total dari setiap orang

- 1.1. Wati

TINGGI BADAN : 165cm

Usia : 42 Tahun

Aktivitas : Menyapu

$$\begin{aligned} \text{BBI} &= (165 - 100) - (0.1 * (165 - 100)) \\ &= 65 - 6.5 \\ &= 58.5 \text{ kg} \end{aligned}$$

$$\text{KKB} = 25\text{kcal} \times 58.5\text{kg} = 1462.5$$

$$\begin{aligned} \text{KKT} &= 1462,5 + 0.2 \times 1462.5 - 0.05 \times \\ &1462.5 \\ &= 1681.8 \end{aligned}$$

1.2. Ali

TINGGI BADAN : 181cm
Usia : 32 Tahun
Aktivitas : *Tour Guide* untuk *hiker*

$$\begin{aligned} \text{BBI} &= (181 - 100) - (0.1 * (181 - 100)) \\ &= 81 - 8.1 \\ &= 72.9 \text{ kg} \end{aligned}$$

$$\text{KKB} = 30\text{kcal} \times 72.9\text{kg} = 2187$$

$$\begin{aligned} \text{KKT} &= 2187 + 0.4 \times 2187 - 0 \times 2187 \\ &= 3061.8 \end{aligned}$$

1.3. Suleman

TINGGI BADAN : 168cm
Usia : 70 Tahun
Aktivitas : Membaca, Menyetir, *Jogging*

$$\begin{aligned} \text{BBI} &= (168-100) - (0.1 \times (168-100)) \\ &= 68 - 6.8 \\ &= 61.2\text{kg} \end{aligned}$$

$$\text{KKB} = 30\text{kcal} \times 61.2\text{kg} = 1836$$

$$\begin{aligned} \text{KKT} &= 1836 + (0.4 + 0.1 + 0.1) \times 1836 - 0.2 \\ &\times 1836 \\ &= 2570.4 \end{aligned}$$

MENGGUNAKAN ALGORITMA *BRUTE FORCE*

Jika hanya menghitung berdasarkan jumlah kalori yang harus dipenuhi menggunakan *brute force*, maka untuk setiap orangnya terdapat lebih dari satu juta kemungkinan kombinasi makanan yang bisa dipilih oleh setiap orangnya. Kompleksitas dari pencarian semua kombinasi makanan dari n makanan yang memenuhi kebutuhan kalori dan tidak melebihi jumlah uang yang dimiliki adalah sebesar $O(2^n)$, karena harus dilakukan pengecekan terhadap semua kombinasi makanan yang mungkin.

Kelebihan dari metode *brute force* pada pencarian menu makanan ini adalah pasti mendapatkan kombinasi dengan harga termurah dan kalori yang paling mencukupi untuk aktivitas di hari itu. Tetapi, tentunya akan sangat boros waktu (dan

akhirnya keburu lapar) jika harus terlebih dahulu menghitung semua kombinasi yang mungkin sebelum memilih makanan, karena itu dibutuhkan pendekatan lain dalam memilih makanan dengan waktu yang lebih singkat.

MENGGUNAKAN ALGORITMA *GREEDY*

PERSIAPAN

Sebelum menggunakan algoritma *Greedy*, kita harus mendefinisikan terlebih dahulu skema yang akan digunakan pada algoritma program:

- Himpunan Kandidat, C , adalah himpunan berisi *array* berisi *dictionary* yang berisi harga, jenis, dan nama makanan yang ada di dalam *database* (berjumlah 136 makanan).
- Himpunan solusi, S , adalah himpunan bagian dari C , dan yang akan kita cari menggunakan algoritma program.
- Fungsi Seleksi, F , pada program akan dicoba dengan 3 program berbeda:
 - *Greed by price*, mengambil makanan dengan harga yang paling murah,
 - *Greed by calories*, mengambil makanan dengan kalori yang paling besar, dan
 - *Greedy by calories/price ratio*, mengambil makanan dengan rasio kalori banding harga yang paling besar. Pendekatan yang terakhir ini mencoba menggabungkan kedua solusi yang ada pada poin sebelumnya untuk mencapai solusi terbaik.
- Fungsi Kelayakan, W (*worthy*), merupakan fungsi yang “menyaring” apakah kandidat pada suatu iterasi sudah memenuhi syarat yang ditetapkan. Pada persoalan ini, suatu kandidat makanan baru akan disebut layak jika makanan tersebut ditambah ke dalam himpunan solusi, memenuhi:
 - Kalori total dari himpunan solusi tidak kurang dari kalori kebutuhan orang tersebut (didapat dari formula KKT),
 - Harga total dari himpunan solusi makanan tidak melebihi *budget* makan per hari orang tersebut, dan ,

Hasil yang didapatkan dari setiap algoritma untuk setiap orangnya:

Nama	Tipe	Greed by		
		Price	Kalori	Ratio
Wati	Kalori	910	2078	2219
	Biaya	17734	19890	19579
Ali	Kalori	2881	1959	4036
	Biaya	47004	49796	49796
Suleman	Kalori	2180	1877	3178
	Biaya	34555	34763	34482

Tabel hasil perhitungan kalori yang didapat dan uang yang harus dikeluarkan yang dihasilkan dari setiap tipe greed per orangnya. *Warna merah menunjukkan kalori yang tidak cukup dari batas KKT atau uang yang dikeluarkan terlalu banyak*

Berdasarkan percobaan menggunakan program, maka dapat disimpulkan strategi *greedy* yang paling menguntungkan adalah menggunakan rasio perbandingan kalori dengan harga makanan, pada ketiga orang yang didata yaitu Wati, Ali, dan Suleman, dengan menggunakan strategi *greedy* menggunakan rasio, kebutuhan kalori harian mereka dapat terpenuhi dan menghabiskan jatah uang makan harian mereka.

Berikut merupakan data ketiga menu untuk ketiga orang yang dihasilkan menggunakan strategi rasio:

- Wati

Harga	Jenis	Kalori	Nama
3387	Pokok	642	Spaghetti
4818	Pokok	506	Nasi Uduk
3661	Pokok	296	Bihun Goreng
1178	Buah	92	Sawo
4201	Pook	321	Mie Goreng
1925	Buah	136	Pisang Rebus

- Ali

Harga	Jenis	Kalori	Nama
3387	Pokok	642	Spaghetti
4818	Pokok	506	Nasi Uduk
3661	Pokok	296	Bihun Goreng
1178	Buah	92	Sawo
4201	Pokok	321	Mie Goreng
1925	Buah	136	Pisang Rebus
5211	Pokok	349	Nasi Kentucky
4099	Pokok	263	Soun Goreng
4451	Pokok	260	Tape Singkong
1457	Buah	82	Apel Merah
2258	Buah	126	Pisang Raja
10594	Siap Saji	588	Nasi Tim Ayam
1943	Buah	105	Pir Ijo

- Suleman

Harga	Jenis	Kalori	Nama
3387	Pokok	642	Spaghetti
4818	Pokok	506	Nasi Uduk
3661	Pokok	296	Bihun Goreng
1178	Buah	92	Sawo

4201	Pook	321	Mie Goreng
1925	Buah	136	Pisang Rebus
5211	Pokok	349	Nasi Kentucky
4099	Pokok	263	Soun Goreng
4451	Pokok	260	Tape Singkong
1457	Buah	82	Apel Merah

KOMPLEKSITAS ALGORITMA GREEDY

Karena semua data makanan harus diurutkan terlebih dahulu, maka kompleksitasnya mengikuti kompleksitas pengurutan data-data makanan, yaitu $O(n \log n)$.

KESIMPULAN

Dari percobaan ini dapat disimpulkan bahwa strategi *greedy* tidak selalu dapat mendapatkan solusi optimum untuk setiap persoalan, dibutuhkan fungsi seleksi yang baik yang dapat memaksimalkan hasil makanan yang didapat, meskipun tidak dapat dibuktikan bahwa solusi yang dihasilkan akan selalu optimum, hasil yang dihasilkan sudah sangat baik untuk memenuhi asupan kalori dari Wati, Ali, dan Suleman.

ANALISIS HASIL DAN KEMUNGKINAN PENGEMBANGAN

Jika dilihat lagi baik-baik, sebenarnya sebagian besar dari hasil menu makanan yang dihasilkan dari *greedy by ratio* diatas sebagian besar merupakan makanan pokok, yang tentunya memang salah satu sumber kalori yang baik, namun agar mencapai badan yang benar-benar sehat, dibutuhkan zat-zat lain seperti protein, vitamin, mineral, dll. Penulis tidak membahas kecukupan gizi dari zat-zat diatas karena keterbatasan waktu, dan juga belum menemukan sumber yang memadai untuk membahas penanganan algoritma pemilihan menu makanan menggunakan tambahan pertimbangan zat-zat diatas. Penulis berharap dengan membaca makalah yang sangat sederhana ini, niat pembaca dapat terbangkitkan untuk membuat artikel dengan isi yang lebih lengkap dan kompleks dari yang penulis buat.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tuhan YME, semua dosen pengampu mata kuliah IF2211 Strategi Algoritma, serta semua pihak yang baik secara langsung maupun tidak langsung ikut membantu terselesaikannya makalah yang dibuat penulis, dan juga memohon apabila dalam makalah ini terdapat kesalahan pada penulisan atau data baik disengaja maupun tidak disengaja.


REFERENSI

- [1] Munir, Rinaldi. 2009. Diktat Kuliah IF2211 Strategi Algoritma. Bandung.
 [2] http://promkes.kemkes.go.id/wp-content/uploads/pdf/publikasi_materi_promosi/Informasi%20CERDIK/5.%20Diet%20Seimbang_425x28.5mm.pdf, diakses 23 April 2019

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran atau terjemahan karya orang lain, dan bukan plagiasi.

Bandung, 26 April 2019



Abda Shaffan Diva
13517021