

LAPORAN ALGORITMA BRANCH AND BOUND
MATA KULIAH DESAIN DAN ANALISIS ALGORITMA



Oleh:

- | | |
|--|---------------------|
| 1. Putu Widyantara Artanta Wibawa | (2108561005) |
| 2. Putu Putri Pratiwi | (2108561010) |
| 3. Kenny Belle Lesmana | (2108561015) |
| 4. I Made Ari Madya Santosa | (2108561020) |

Kelas E

PROGRAM STUDI INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS UDAYANA

2022

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, penulis dapat menyusun laporan “*Algoritma Branch-and-Bound*” ini dan menyelesaikannya tepat waktu tanpa adanya halangan yang berarti.

Laporan ini penulis dibuat dalam rangka memenuhi tugas mata kuliah Teori Bahasa dan Otomata yang isinya dibuat dengan jujur sesuai fakta yang penulis dapatkan melalui studi pustaka dan referensi di internet.

Penulis juga menyadari bahwa selama berlangsungnya proses penyusunan sampai pada tahap penyelesaian laporan ini tak lepas dari dukungan dan bantuan dari berbagai pihak. Oleh karena itu, teriring doa dan ucapan terima kasih yang penulis sampaikan kepada:

1. Ibu Dr. Dra. Luh Gede Astuti, M. Kom., selaku dosen pengampu mata kuliah yang telah memberikan bimbingan dan bantuan kepada penulis.
2. Semua pihak yang telah berperan secara maksimal dalam proses penyusunan laporan ini yang tidak dapat penulis sebutkan satu persatu.

Karena keterbatasan ilmu maupun pengalaman, penulis menyadari bahwa laporan ini memiliki banyak kekurangan dan jauh dari kata sempurna. Oleh karena itu, penulis menanti saran maupun kritik yang bersifat membangun dari semua pihak demi kesempurnaan laporan ini.

Akhir kata, penulis berharap agar laporan yang telah dibuat ini mampu memberikan wawasan dan khazanah keilmuan kepada para pembaca khususnya mahasiswa Universitas Udayana di Program Studi Informatika.

Denpasar, 22 Desember 2022

Penulis

DAFTAR ISI

Cover	i
Kata Pengantar	ii
Daftar Isi	iii
BAB I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan	2
1.4 Manfaat	2
BAB II. PEMBAHASAN	3
2.1 Algoritma Branch-and-Bound	3
2.2 Cara Kerja Algoritma	3
2.3 Kompleksitas Algoritma	3
2.4 Desain Aplikasi	5
2.5 Platform Aplikasi	6
2.6 Implementasi	7
BAB III. PENUTUP	19
3.1 Kesimpulan	19
3.2 Saran	19
Daftar Pustaka	

BAB I

PENDAHULUAN

1.1 Latar Belakang

Algoritma *Branch and Bound* pertama kali dikembangkan oleh A. Land dan G. Doig pada tahun 1960, untuk masalah program bilangan bulat baik murni maupun campuran. Lima tahun setelahnya, muncul algoritma pada program integer khusus nol satu pada tahun 1965 yang dikembangkan oleh E. Balas. Algoritma ini juga merupakan metode pencarian di dalam ruang solusi sistematis.

Prinsip yang mendasar pada pendekatan *Branch and Bound* ini adalah total set solusi yang fisibel dan dibagi menjadi sub masalah yang lebih kecil. Penerapan pendekatan metode Branch and Bound digunakan bersama-sama dengan metode simpleks yang normal. Metode ini menggunakan suatu diagram yang terdiri dari node dan cabang sebagai suatu kerangka dalam proses solusi. Solusi program linier dan solusi pembulatan sendiri diisi pada node pertama dalam diagram Branch and Bound. Untuk skema BnB ini, menggunakan skema BFS (*Best-First Search*).

Algoritma *Branch and Bound* biasanya digunakan dalam persoalan optimasi. Persoalan optimasi atau *Optimization Problem* merupakan persoalan yang mengharuskan menemukan suatu solusi terbaik dari beberapa solusi yang ada. Persoalan yang dapat dicontohkan, seperti *Knapsack Problem* dan *N Queen Problem*. *Knapsack Problem* merupakan persoalan memasukkan jumlah barang ke dalam karung dengan kapasitas terbanyak, namun dengan nilai barang sebanyak mungkin. *N Queen Problem* merupakan persoalan meletakkan sejumlah menteri pada papan catur berukuran $n \times n$ sehingga tidak ada menteri yang saling menyerang. Pada laporan ini penulis akan hanya membahas mengenai *Knapsack Problem* menggunakan Algoritma *Branch and Bound*.

1.2 Rumusan Masalah

Dari latar belakang di atas, adapun rumusan masalah yang dapat dirumuskan sebagai berikut:

1. Bagaimana implementasi BnB dengan skema *Best-First Search* permasalahan *Knapsack*?
2. Bagaimana implementasi algoritma BnB dalam kehidupan nyata?
3. Bagaimana kompleksitas algoritma *Branch and Bound* ini?

1.3 Tujuan

Adapun tujuan dari pembuatan laporan ini sebagai berikut

1. Untuk mengetahui cara implementasi BnB dengan skema *Best-First Search* untuk permasalahan *Knapsack*.
2. Untuk mengetahui implementasi algoritma BnB dalam kehidupan nyata.
3. Untuk mengetahui kompleksitas algoritma *Branch and Bound*.

1.4 Manfaat

Adapun manfaat yang diperoleh dari penulisan laporan ini, yaitu

1. Membuat aplikasi yang menerapkan Algoritma *Branch and Bound* untuk memilih makanan dengan protein tertinggi.
2. Mengetahui cara implementasi Algoritma *Branch and Bound* dengan Bahasa Pemrograman *Python*.

BAB II

PEMBAHASAN

2.1 Algoritma Branch and Bound

Algoritma *Branch and Bound* merupakan suatu metode / algoritma yang digunakan untuk menyelesaikan persoalan optimisasi. Algoritma *Branch and Bound* ini menggunakan skema BFS atau *Best-First Search*. Persoalan optimasi atau optimization problem adalah persoalan dimana kita diharuskan menemukan solusi terbaik dari beberapa solusi yang ada. Contoh dari persoalan optimasi adalah *Knapsack Problem*.

2.2 Cara Kerja Algoritma

Algoritma *Branch and Bound* memiliki cara kerja sebagai berikut

1. Masukkan simpul akar ke dalam antrian Q, jika simpul akar adalah simpul solusi (*goal node*), maka solusi telah ditemukan. Proses berakhir.
2. Jika Q kosong, maka tidak ada solusi. Proses berhenti.
3. Jika Q tidak kosong, pilih dari antrian Q simpul i yang mempunyai *bound* yang paling besar. Jika terdapat beberapa simpul i yang memenuhi, pilih satu secara sembarang.
4. Jika simpul i adalah simpul solusi, berarti solusi sudah ditemukan, proses berhenti. Jika simpul i bukan simpul solusi, maka bangkitkan semua anak-anaknya.
5. Untuk setiap anak dari simpul i, hitung *bound* nya dan masukkan semua anak-anaknya ke dalam Q.
6. Kembali ke langkah 2.

2.3 Kompleksitas Algoritma

1. Kompleksitas Waktu

Kompleksitas ruang merupakan kompleksitas yang diukur dari memori yang digunakan oleh struktur data yang terdapat di dalam algoritma sebagai fungsi dari ukuran masukan n. Kompleksitas ruang yang dimiliki oleh algoritma branch and bound bergantung pada jumlah branch dan tinggi maksimum dari pohon yang dibentuk. Secara matematis, kompleksitas waktu dari algoritma ini adalah:

a. Input size = banyaknya node / makanan = n

b. Basic operation = mengecek sebuah node

c. Recurrence relation:

$$X(n) = 2X(n-1) + 1, \text{ untuk } n > 1$$

$$X(n) = 1, \text{ untuk } n = 1$$

$$X(n) = 2X(n-1) + 1$$

$$2[2X(n-2) + 1] + 1 = 2^2X(n-2) + 2 + 1$$

$$2^2X(n-2) + 2 + 1 = 2^2[2X(n-3) + 1] + 2 + 1$$

$$= 2^3X(n-3) + 4 + 2 + 1$$

$$2^3X(n-3) + 4 + 2 + 1 = 2^3[2X(n-4) + 1] + 4 + 2 + 1$$

$$= 2^4X(n-4) + 8 + 4 + 2 + 1$$

d. Substitusi

$$X(n) = 2^iX(n-i) + 2^{i-1} + 2^{i-2} + \dots + 2 + 1$$

$$= 2^iX(n-i) + 2^i - 1$$

e. Substitusi kondisi awal

$$n = 1$$

$$n - i = 1$$

$$i = n - 1$$

$$X(n) = 2^iX(n-i) + 2^i - 1$$

$$X(n) = 2^{(n-1)}X(n - (n-1)) + 2^{(n-1)} - 1$$

$$X(n) = 2^{(n-1)}X(1) + 2^{(n-1)} - 1$$

$$X(n) = 2^{(n-1)}(1) + 2^{(n-1)} - 1$$

$$X(n) = 2^{(n-1)} + 2^{(n-1)} - 1$$

$$X(n) = 2(2^{(n-1)}) - 1$$

$$X(n) = 2^{n-1+1} - 1$$

$$X(n) = 2^n - 1$$

f. Notasi Asimtotik

Dengan menggunakan limit, didapatkan:

$$\lim_{n \rightarrow \infty} \left(\frac{t(n)}{g(n)} \right)$$

$$= \frac{t(2^n - 1)}{g(2^n)} = 1$$

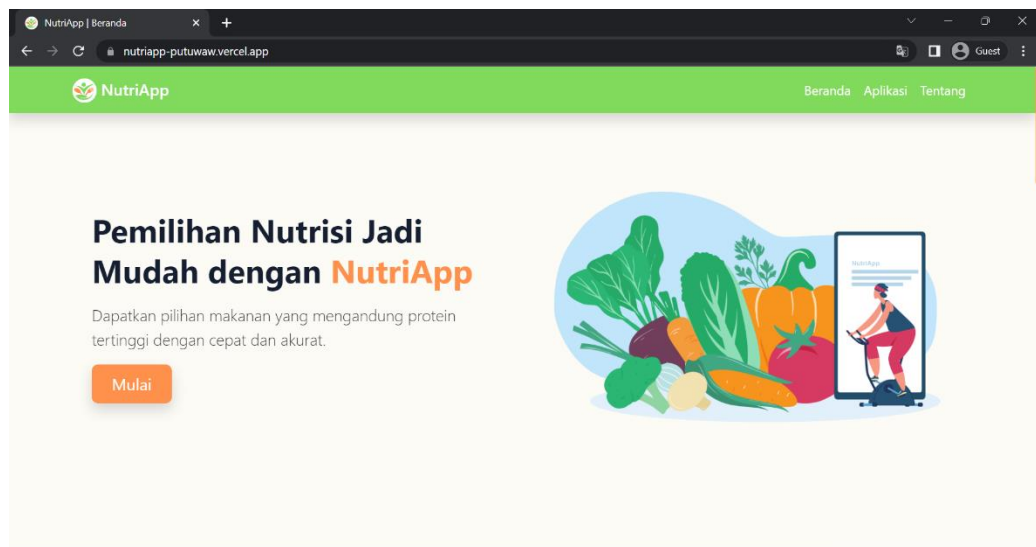
Karena hasilnya adalah sebuah konstanta, maka notasi asimtotiknya adalah theta atau $\Theta(2^N)$.

2. Kompleksitas Ruang

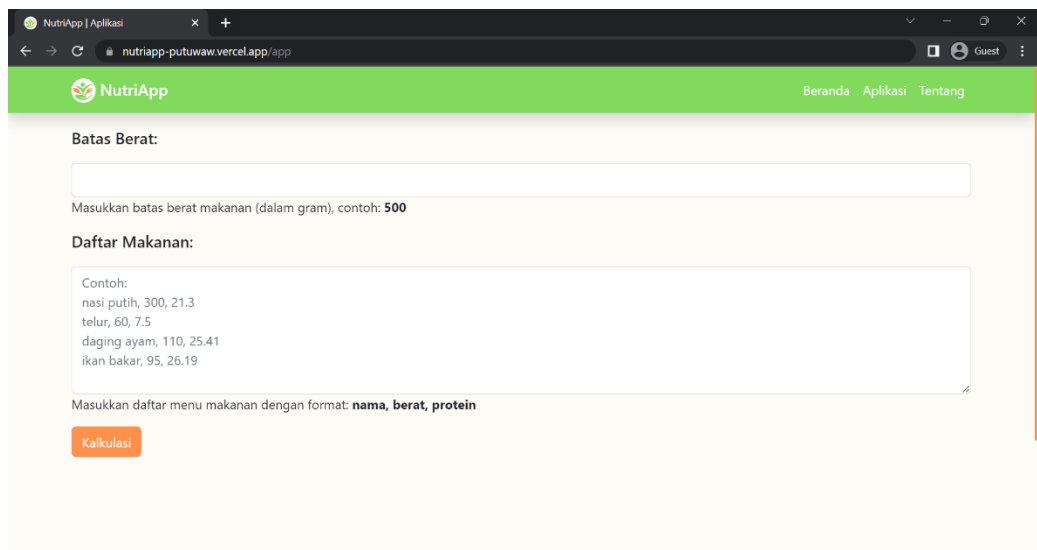
Kompleksitas waktu merupakan kompleksitas yang diukur dari jumlah tahapan komputasi yang dibutuhkan untuk menjalankan algoritma sebagai fungsi dari ukuran masukan n . Secara umum atau dalam average case, algoritma branch and bound mendekati kompleksitas eksponensial atau 2^n .

2.4 Desain Aplikasi

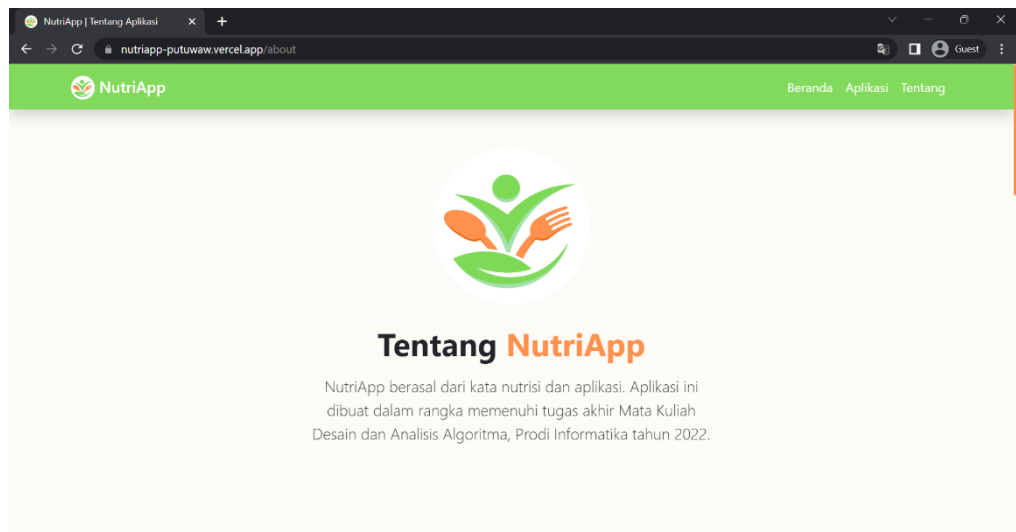
Aplikasi yang kami kembangkan untuk mengimplementasikan algoritma branch-and-bound ini bernama NutriApp. Aplikasi ini berbasis website dan dapat dikunjungi melalui tautan berikut: <https://nutriapp-putuwaw.vercel.app/>. NutriApp terdiri dari 3 halaman, yaitu: beranda, aplikasi, dan tetnang aplikasi. Berikut adalah desain dari aplikasi NutriApp.



Tampilan Beranda Aplikasi



Tampilan Utama Aplikasi



Tampilan Tentang Aplikasi

2.5 Platform Aplikasi

Aplikasi NutriApp dikembangkan menggunakan bahasa pemrograman Python sebagai bahasa pemrograman utama untuk mengimplementasikan algoritma branch-and-bound dibantu dengan beberapa bahasa pemrograman web seperti HTML, CSS, dan JavaScript.

Selain itu, aplikasi yang kami kembangnya juga menggunakan bantuan beberapa platform atau penyedia layanan jasa seperti:

1. Flask

Flask adalah sebuah web framework yang dibuat dengan bahasa pemrograman Python. Flask berfungsi sebagai untuk membuat aplikasi berbasis website.

2. Vercel

Vercel adalah sebuah platform atau penyedia layanan hosting yang dapat digunakan untuk deploy berbagai macam aplikasi frontend. Selain itu, Vercel juga mendukung untuk melakukan deploy aplikasi Flask.

3. Github

Github adalah sebuah platform yang berfungsi untuk menyimpan dan mengelola kode sumber atau source code dari aplikasi yang kami kembangkan. Kode sumber dari aplikasi kami dapat diakses pada tautan berikut:
<https://github.com/putuwaw/nutriapp>

2.6 Implementasi Coding

Berikut adalah implementasi dari penerapan algoritma Branch and Bound dengan pendekatan Best-First dalam bahasa pemrograman Python untuk memilih makanan dengan tingkat protein tertinggi. Seluruh kode sumber dari aplikasi dapat dilihat pada tautan berikut: <https://github.com/putuwaw/nutriapp>

Tabel Implementasi Coding Tampilan Beranda Aplikasi

Nama file: index.html

Tautan: <https://github.com/putuwaw/nutriapp/blob/main/templates/index.html>

No.	Program	Penjelasan
1	<code>{% extends "base.html" %}</code>	
2	<code>{% block title %} NutriApp Beranda {% endblock %}</code>	
3	<code>{% block body %}</code>	
4	<code><section data-aos="fade-up" class="container-lg col-xxl-8 px-4 py-5"></code>	Section untuk hero
5	<code><div class="row flex-lg-row-reverse align-items-center justify-content-center g-5 py-5"></code>	
6	<code><div class="col-10 col-sm-8 col-lg-6"></code>	

7	<code></code>	
8	<code></div></code>	
9	<code><div class="col-lg-6 text-center text-lg-start"></code>	
10	<code><h1 class="display-6 fw-bold lh-sm mb-3 text-slate">Pemilihan Nutrisi Jadi Mudah dengan NutriApp</h1></code>	
11	<code><p class="lead">Dapatkan pilihan makanan yang mengandung protein tertinggi dengan cepat dan akurat.</p></code>	
12	<code><div class="d-grid gap-2 d-md-flex justify-content-md-start"></code>	
13	<code></code>	
14	<code><button type="button" class="btn btn-orange btn-lg px-4 me-md-2 text-white shadow">Mulai</button></code>	
15	<code></code>	
16	<code></div></code>	
17	<code></div></code>	
18	<code></div></code>	
19	<code></section></code>	
20	<code><section id="mulai" class="px-4 pt-5 my-5 text-center border-bottom"></code>	Section untuk apakah itu NutriApp
21	<code><h1 data-aos="fade-up" class="display-6 fw-bold pt-lg-5 pb-2">Apa itu NutriApp?</h1></code>	
22	<code><div data-aos="fade-up" class="col-lg-6 mx-auto"></code>	
23	<code><p class="lead mb-4">NutriApp adalah sebuah aplikasi berbasis website atau web app yang mampu untuk melakukan pemilihan makanan dengan kandungan protein tertinggi, sehingga Anda tidak bingung dalam memilih makanan.</p></code>	
24	<code><div class="d-grid gap-2 d-sm-flex justify-content-sm-center mb-5"></code>	
25	<code></code>	
26	<code><button type="button" class="btn btn-lg btn-orange-reverse px-4 shadow">Selengkapnya</button></code>	
27	<code></code>	
28	<code></div></code>	
29	<code></div></code>	
30	<code><div data-aos="fade-up" class="overflow-hidden" style="max-height: 30vh;"></code>	
31	<code><div class="container px-5"></code>	
32	<code></code>	
33	<code></div></code>	
34	<code></div></code>	

35	</section>	Section untuk mengapa harus NutriApp
36	<section class="container px-4 py-5">	
37	<h1 data-aos="fade-up" class="display-6 fw-bold pt-lg-5 pb-2 text-center mt-lg-5">Mengapa harus NutriApp?</h1>	
38	<div data-aos="fade-up" class="row g-4 py-5 row-cols-1 row-cols-lg-3">	
39	<div class="feature col">	
40	<div class="w-100 d-flex flex-column text-center align-items-center justify-content-center">	
41	<div class="feature-icon w-25 d-inline-flex align-items-center justify-content-center bg-orange-s bg-gradient fs-2 mb-3">	
42		
43	</div>	
44	<h3 class="fs-2 fw-bold">Open Source</h3>	
45	<p>NutriApp bersifat Open Source, artinya semua orang dapat memberikan kontribusi untuk pengembangan dan melihat kode sumber dari aplikasi.</p>	
46	</div>	
47	</div>	
48	<div class="feature col">	
49	<div class="w-100 d-flex flex-column text-center align-items-center justify-content-center">	
50	<div class="feature-icon w-25 d-inline-flex align-items-center justify-content-center bg-orange-s bg-gradient fs-2 mb-3">	
51		
52	</div>	
53	<h3 class="fs-2 fw-bold">Algoritma B&B</h3>	
54	<p>NutriApp menggunakan algoritma Branch-and-Bound dengan pendekatan Best-First, yang biasa digunakan untuk pemecahan masalah optimasi atau optimization problem.</p>	
55	</div>	
56	</div>	
57	<div class="feature col">	
58	<div class="w-100 d-flex flex-column text-center align-items-center justify-content-center">	
59	<div class="feature-icon w-25 d-inline-flex align-items-center justify-content-center bg-orange-s bg-gradient fs-2 mb-3">	
60		
61	</div>	

62	<code><h3 class="fs-2 fw-bold">Berbasis Web</h3></code>	
63	<code><p>NutriApp adalah aplikasi berbasis website, sehingga bisa diakses kapan saja dan dimana saja baik menggunakan perangkat seperti laptop maupun smartphone.</p></code>	
64	<code></div></code>	
65	<code></div></code>	
66	<code></div></code>	
67	<code></section></code>	
68	<code><section data-aos="fade-up" class="container-lg col-xxl-8 px-4 py-5"></code>	Section untuk coba NutriApp
69	<code><div class="row flex-lg-row-reverse align-items-center justify-content-center g-5 py-5"></code>	
70	<code><div class="col-10 col-sm-8 col-lg-6"></code>	
71	<code></code>	
72	<code></div></code>	
73	<code><div class="col-lg-6 text-center text-lg-start"></code>	
74	<code><h1 class="display-6 fw-bold lh-sm mb-3 text-slate">Pilih Makanan Bernutrisi dengan NutriApp!</h1></code>	
75	<code><div class="d-grid gap-2 d-md-flex justify-content-md-start"></code>	
76	<code></code>	
77	<code><button type="button" class="shadow btn btn-orange btn-lg px-4 me-md-2 text-white mt-lg-3">Coba Sekarang</button></code>	
78	<code></code>	
79	<code></div></code>	
80	<code></div></code>	
81	<code></div></code>	
82	<code></section></code>	
83		
84	<code>{% endblock %}</code>	

Tabel Implementasi Coding Tampilan Utama Aplikasi

Nama file: app.html

Tautan: <https://github.com/putuwaw/nutriapp/blob/main/templates/app.html>

No.	Program	Penjelasan
1	<code>{% extends "base.html" %}</code>	
2	<code>{% block title %} NutriApp Aplikasi {% endblock %}</code>	
3	<code>{% block body %}</code>	
4	<code><section data-aos="fade-up" class="mt-3 min-vh-100"></code>	

5	<form action="" method="post">	Section untuk form aplikasi
6	<div class="mb-3">	
7	<p class="lead fw-semibold">Batas Berat:</p>	
8	<input type="number" name="limit" class="form-control p-2" id="limit" aria-describedby="limitHelp" required>	
9	<p class=" " >Masukkan batas berat makanan (dalam gram), contoh: 500</p>	
10	</div>	
11	<div class="mb-3">	
12	<p class="lead fw-semibold">Daftar Makanan:</p>	
13	<textarea class="form-control" name="detail" id="detail" rows="6" required placeholder="Contoh:
nasi putih, 300, 21.3
telur, 60, 7.5
daging ayam, 110, 25.41
ikan bakar, 95, 26.19"></textarea>	
14	<p class="">Masukkan daftar menu makanan dengan format: 	
15	nama, berat, protein	
16		
17	</p>	
18	</div>	
19	<button type="submit" class="btn btn-orange text-white">Kalkulasi</button>	
20	</form>	
21	{% if isSubmit %}	Section untuk hasil perhitungan
22	<section class="mt-5">	
23	<div class="border-orange w-75 h-auto mx-auto text-center rounded rounded-3">	
24	<h3 class="fw-bold my-3">Space State Tree</h3>	
25	<div id="graph" class="text-center w-100 my-3"></div>	
26	</div>	
27	<pre id="resultGraph" class="hiding d-none">	
28	{{graph}}	
29	</pre>	
30	<div class="w-75 mx-auto border-orange rounded rounded-3 my-3 p-3">	
31	<h3 class="fw-bold text-center">Hasil</h3>	
32	<h6 class="lead fw-semibold">Waktu eksekusi program: </h6> <p class="lead">{{time}} detik</p>	
33	<h6 class="lead fw-semibold">Jumlah protein maksimal yang bisa didapat: </h6> <p class="lead">{{maxProfit}} protein.</p>	
34	<h6 class="lead fw-semibold">Makanan yang harus diambil yaitu:</h6>	
35	<ol class="lead">	
36	{%- for i in range(selectedItems length) -%}	
37		

38	{{i+1}}.{{selectedItems[i]}}	
39	{%- endfor -%}	
40		
41	</div>	
42	</section>	
43	{% endif %}	
44	</section>	
45	{% endblock %}	

Tabel Implementasi Coding Tampilan Tentang Aplikasi

Nama file: about.html

Tautan: <https://github.com/putuwaw/nutriapp/blob/main/templates/about.html>

No.	Program	Penjelasan
1	{% extends "base.html" %}	
2	{% block title %} NutriApp Tentang Aplikasi {% endblock %}	
3	{% block body %}	
4	<section data-aos="fade-up" class="container-lg col-xxl-8 px-4 pb-5">	Section untuk penjelasan mengenai aplikasi
5	<div class="px-4 py-5 mb-5 text-center">	
6	<div>	
7		
8	</div>	
9	<h1 class="display-6 fw-bold mb-3">Tentang NutriApp</h1>	
10	<div class="col-lg-6 mx-auto">	
11	<p class="lead mb-4">NutriApp berasal dari kata nutrisi dan aplikasi.	
12	Aplikasi ini dibuat dalam rangka memenuhi tugas akhir Mata Kuliah Desain dan Analisis Algoritma, Prodi Informatika tahun 2022.	
13	</p>	
14	</div>	
15	</div>	
16	</section>	
17	<section class="pb-5 mb-5">	Section tentang pengembang dari aplikasi
18	<div class="row text-center pb-5">	
19	<h1 data-aos="fade-up" class="display-6 fw-bold mb-3 pb-5">Pengembang NutriApp</h1>	
20	<div data-aos="fade-up" class="col-lg-3">	
21		

22	<code><h2 class="fs-4 mt-3 fw-bold">Putu Widyantara Artanta Wibawa</h2></code>	
23	<code><p class="fs-6">NIM 2108561005</p></code>	
24	<code></div></code>	
25	<code><div data-aos="fade-up" class="col-lg-3"></code>	
26	<code></code>	
27	<code><h2 class="fs-4 mt-3 fw-bold">Putu Putri Pratiwi</h2></code>	
28	<code><p>NIM 2108561010</p></code>	
29	<code></div></code>	
30	<code><div data-aos="fade-up" class="col-lg-3"></code>	
31	<code></code>	
32	<code><h2 class="fs-4 mt-3 fw-bold">Kenny Belle Lesmana</h2></code>	
33	<code><p>NIM 2108561015</p></code>	
34	<code></div></code>	
35	<code><div data-aos="fade-up" class="col-lg-3"></code>	
36	<code></code>	
37	<code><h2 class="fs-4 mt-3 fw-bold">I Made Ari Madya Santosa</h2></code>	
38	<code><p>NIM 2108561020</p></code>	
39	<code></div></code>	
40	<code></div></code>	
41	<code></section></code>	
42	<code><section class="pb-5 mb-5"></code>	Section untuk teknologi yang digunakan dalam pengembangan aplikasi
43	<code><div class="row align-items-md-stretch text-center pb-5 mb-3"></code>	
44	<code><h1 data-aos="fade-up" class="display-6 fw-bold mb-3 pb-5">Teknologi NutriApp</h1></code>	
45	<code><div data-aos="fade-up" class="col-md-6 mb-3 mb-lg-0"></code>	
46	<code><div class="h-100 p-4 rounded-3 btn-orange-reverse shadow"></code>	
47	<code><h2 class="fs-3 mt-3 fw-bold">Flask</h2></code>	
48	<code><hr></code>	
49	<code><p class="lead">Flask adalah sebuah web framework yang ditulis dengan bahasa Python dan tergolong sebagai jenis microframework. Flask berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu web. </p></code>	
50	<code></div></code>	
51	<code></div></code>	
52	<code><div data-aos="fade-up" class="col-md-6"></code>	
53	<code><div class="h-100 p-4 rounded-3 btn-orange-reverse shadow"></code>	
54	<code><h2 class="fs-3 mt-3 fw-bold">Vercel</h2></code>	

55	<code><hr></code>	
56	<code><p class="lead">Vercel adalah sebuah platform atau penyedia layanan hosting yang dapat digunakan untuk deploy berbagai macam aplikasi frontend. Selain itu, Vercel juga mendukung untuk melakukan deploy aplikasi Flask.</p></code>	
57	<code></div></code>	
58	<code></div></code>	
59	<code></div></code>	
60	<code></section></code>	
61		
62	<code><section></code>	Section untuk dokumentasi aplikasi
63	<code><div class="row featurette text-center"></code>	
64	<code><h1 data-aos="fade-up" class="display-6 fw-bold mb-3 pb-5">Dokumentasi NutriApp</h1></code>	
65	<code><div data-aos="fade-up" class="col-md-7 text-start"></code>	
66	<code><div class="d-flex flex-column justify-content-center h-100 align-items-start "></code>	
67	<code><p class="lead">Seluruh dokumentasi dari aplikasi NutriApp, baik berupa source code dan laporan tersedia pada halaman GitHub.</p></code>	
68	<code><div></code>	
69	<code></code>	
70	<code><button type="button" class="shadow btn btn-orange btn-lg px-4 me-md-2 text-white">Source Code</button></code>	
71	<code></code>	
72	<code></code>	
73	<code><button type="button" class="shadow btn btn-orange btn-lg px-4 me-md-2 text-white">Laporan</button></code>	
74	<code></code>	
75	<code></div></code>	
76	<code></div></code>	
77	<code></div></code>	
78	<code><div data-aos="fade-up" class="col-md-5 mt-3 mt-lg-0"></code>	
79	<code><div class="shadow-lg"></code>	
80	<code></code>	
81	<code></div></code>	
82	<code></div></code>	
83	<code></div></code>	
84	<code></section></code>	
85	<code>{% endblock %}</code>	

Tabel Implementasi Algoritma Best-First Branch-and-Bound

Nama file: branch_and_bound.py

Tautan: https://github.com/putuwaw/nutriapp/blob/main/modules/branch_and_bound.py

No.	Program	Penjelasan
1	import graphviz	Import graph dan juga time untuk menampilkan tree dan waktu eksekusi program
2	import time	
3		
4	# global variables	Deklarasi global var
5	N = 0	Banyak makanan
6	W = 0	Batas berat
7	w = []	List dari berat makanan
8	p = []	List dari kandungan protein makanan
9	pPerW = []	List dari kandungan protein per gram
10		
11	# class priority queue	Membuat kelas priority queue untuk implementasi best-first branch and bound
12	class PriorityQueue:	
13	def __init__(self):	
14	self.queue = []	
15	self.size = 0	
16	# enqueue with priority (bound)	
17	def enqueue(self, node):	
18	if (self.size == 0):	
19	self.queue.append(node)	
20	else:	
21	saveI = 0	
22	for i in range(self.size):	
23	if node.bound > self.queue[i].bound:	
24	saveI = i	
25	break	
26	self.queue.insert(saveI, node)	
27	self.size += 1	
28	# dequeue like normal queue (front)	
29	def dequeue(self):	
30	result = self.queue.pop(0)	
31	self.size -= 1	
32	return result	
33		
34	# class node	Membuat kelas node atau vertex
35	class Node:	
36	def __init__(self, level, profit, weight, bound=0, number=0, items=[]):	
37	self.level = level	Level dari vertex
38	self.profit = profit	Protein saat ini
39	self.weight = weight	Berat saat ini
40	self.bound = bound	Bound atau batas saat ini

41	<code>self.number = number</code>	Jumlah vertex yang telah dihasilkan
42	<code>self.items = items</code>	
43		
44	<code># function to get bound for node</code>	Fungsi untuk mencair bound atau batas dari suatu vertex
45	<code>def get_bound(node):</code>	
46	<code>global N, W, w, p, pPerW</code>	
47	<code># if weight is already over limit or leaf node</code>	
48	<code>if node.weight >= W or node.level >= N-1:</code>	
49	<code>return 0</code>	
50	<code>else:</code>	
51	<code>result = node.profit</code>	
52	<code>j = node.level + 1</code>	
53	<code>weightLeft = W - node.weight</code>	
54	<code>result += pPerW[j] * weightLeft</code>	
55	<code>return result</code>	
56		
57	<code># main function</code>	Fungsi utama
58	<code>def solve(limit, detail):</code>	
59	<code># start time</code>	Memulai perhitungan waktu
60	<code>startTime = time.time()</code>	
61	<code># access global variables</code>	Mengakses variabel global
62	<code>global N, W, w, p, pPerW, Node, PriorityQueue</code>	
63	<code>N = 0</code>	
64	<code>W = 0</code>	
65	<code>w.clear()</code>	
66	<code>p.clear()</code>	
67	<code>pPerW.clear()</code>	
68	<code># remove newline</code>	Mengolah input dari user
69	<code>detailList = detail.replace("\r\n", ", ").split(", ")</code>	
70	<code># convert to float</code>	
71	<code>for i in range(len(detailList)):</code>	
72	<code>if i % 3 != 0:</code>	
73	<code>detailList[i] = float(detailList[i])</code>	
74	<code># initialize variables</code>	
75	<code>N = len(detailList) // 3</code>	
76	<code>W = float(limit)</code>	
77	<code>detailList = [detailList[i:i+3] for i in range(0, len(detailList), 3)]</code>	
78	<code>dupDetailList = detailList.copy()</code>	
79	<code>tempDict = {}</code>	
80	<code>for i in range(N):</code>	
81	<code>tempDict[i] = detailList[i][2] / detailList[i][1]</code>	

82	sortedDict = sorted(tempDict.items(), key=lambda x: x[1], reverse=True)	
83	detailList.clear()	
84	for i in range(N):	
85	detailList.append(dupDetailList[sortedDict[i][0]])	
86	for i in range(N):	
87	w.append(detailList[i][1])	
88	p.append(detailList[i][2])	
89	pPerW.append(sortedDict[i][1])	
90	# initialize priority queue	Insiasi priority queue
91	pq = PriorityQueue()	
92	# initialize root node	Inisiasi root
93	v = Node(-1, 0, 0)	
94	v.bound = get_bound(v)	
95	# enqueue root node	
96	pq.enqueue(v)	
97	# initialize utils variable	
98	maxProfit = 0	
99	selectedItems = []	
100	nodeNumber = 0	
101	# graph	
102	g = graphviz.Digraph(name="Branch and Bound")	
103	# best-first search	Melakukan pencarian hingga node yang dicari memiliki bound kurang dari profit atau kandungan porotein saat ini
104	while(pq.size > 0):	
105	v = pq.dequeue()	
106	if (v.bound > maxProfit):	
107	# make child node (with)	
108	u = Node(v.level + 1, v.profit + p[v.level+1], v.weight+w[v.level+1])	
109	nodeNumber += 1	
110	u.bound = get_bound(u)	
111	u.number = nodeNumber	
112	u.items = v.items.copy()	
113	u.items.append(u.level)	
114	# calculate max profit	
115	if u.weight <= W and u.profit > maxProfit:	
116	maxProfit = u.profit	
117	selectedItems = u.items.copy()	
118	# make child node (without)	
119	uo = Node(u.level, v.profit, v.weight)	
120	nodeNumber += 1	
121	uo.bound = get_bound(uo)	
122	uo.number = nodeNumber	
123	uo.items = v.items.copy()	
124	# enqueue child nodes	

125	if uo.bound > maxProfit:	
126	pq.enqueue(uo)	
127	if u.bound > maxProfit:	
128	pq.enqueue(u)	
129	# drawing graph	Proses pembuatan tree
130	strU = "ub="+str(u.bound)+ " v="+str(u.profit)+", w="+ str(u.weight) + " (" + str(u.number) + ")"	
131	strU0 = "ub="+str(uo.bound)+ " v="+str(uo.profit)+", w="+ str(uo.weight) + " (" + str(uo.number) + ")"	
132	strV = "ub="+str(v.bound)+ " v="+str(v.profit)+", w="+ str(v.weight) + " (" + str(v.number) + ")"	
133	if (u.level == 0):	
134	g.edge(strV, strU, label="dengan "+detailList[u.level][0])	
135	g.edge(strV, strU0, label="tanpa "+detailList[u.level][0])	
136	else:	
137	g.edge(strV, strU, label="dengan "+detailList[u.level][0])	
138	g.edge(strV, strU0, label="tanpa "+detailList[u.level][0])	
139	# result	Hasil atau return value
140	resultList = []	
141	# 0 -> max profit	
142	resultList.append(maxProfit)	
143	# 1 -> selected items	
144	tempList = []	
145	for i in selectedItems:	
146	tempList.append(detailList[i][0])	
147	resultList.append(tempList)	
148	# 2 -> graph	
149	resultList.append(g)	
150	# 3 -> time	
151	totalTime = time.time() - startTime	
152	resultList.append(totalTime)	
153	# return value	
154	return resultList	

BAB III

PENUTUP

3.1 Kesimpulan

Algoritma Branch and Bound merupakan suatu metode atau algoritma yang digunakan untuk menyelesaikan persoalan optimasi. Algoritma Branch-and-Bound ini menggunakan skema BFS atau Best-First Search. Problem optimasi adalah persoalan dimana kita diharuskan menemukan solusi terbaik dari beberapa solusi yang ada..

Metode ini menggunakan suatu diagram yang terdiri dari node dan cabang sebagai suatu kerangka dalam proses solusi. Solusi program linier dan solusi pembulatan sendiri diisi pada node pertama dalam diagram Branch and Bound.

3.2 Saran

Saran-saran yang bisa diberikan dalam implementasi algoritma Branch-and-Bound ini adalah dalam implementasi algoritma branch-and-bound, terdapat beberapa pendekatan atau skema yang bisa digunakan, sehingga harus dipilih skema yang dapat menghasilkan kompleksitas terendah.

DAFTAR PUSTAKA

- [1] Munir, R., 2021. *Algoritma Branch & Bound*. [Online] Available at: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian4.pdf> [Accessed 22 December 2022].