

Penerapan Dynamic Programming Dalam Optimisasi Pesanan Pada Platform Pemesanan Makanan Online

Rifqi Naufal Abdjul - 13520062
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail : 13520062@std.stei.itb.ac.id

Abstract—Platform pemesanan makanan online merupakan platform dimana pengguna dapat memesan makanan secara online. Dikarenakan perkembangan marketing pada platform tersebut, terdapat perhitungan diskon yang cukup kompleks untuk digunakan oleh pengguna. Pada makalah ini, akan dijelaskan penerapan dynamic programming dalam mengoptimasi pesanan dalam platform tersebut.

Keywords—platform pemesanan makanan online; dynamic programming; diskon; optimisasi

I. PENDAHULUAN

Platform pemesanan makanan online merupakan hal yang marak pada masa kini, dimana seluruh aktivitas lebih baik dilakukan secara daring untuk meningkatkan efektivitas dengan tetap dalam satu tempat (biasanya rumah). Platform pemesanan makanan online ini sangat mendukung perilaku untuk tidak bepergian, yang saat masa pandemi ini sangat dianjurkan demi kesehatan masing masing.

Platform pemesanan makanan secara definisi adalah platform dimana pengguna dapat menggunakan platform tersebut untuk membuat pesanan terhadap suatu merchant yang sudah mendaftarkan tokonya pada platform tersebut. Platform tersebut juga biasanya menyediakan jasa pengantaran makanan agar pengguna dapat hanya menerima dari lokasi masing masing.



Gambar 1.1 Beberapa contoh dari Platform Pemesanan Makanan Online
Sumber: Google Images

Dikarenakan perkembangan marketing dan persaingan yang lumayan ketat antar platform, maka terdapat sistem diskon yang sangat menarik pengguna tetapi sering kurang menguntungkan dalam posisi pengguna. Kurang menguntungkan karena biasanya diskon tersebut dibatasi agar tidak terlalu merugikan merchant maupun pihak platform.

Untuk memanfaatkan diskon dengan cara yang paling efisien, dapat dimanfaatkan dynamic programming untuk melakukan searching seluruh kemungkinan menu yang dapat dipesan, lalu dapat diurutkan berdasarkan beberapa parameter seperti harga per porsi, persentase diskon per harga total, dan lain lain.

II. TEORI DASAR

A. Pendekatan Brute Force

Pendekatan brute force merupakan pendekatan yang paling sederhana dalam penyelesaian masalah. Dimana penyelesaian tersebut berdasarkan dengan topik permasalahan dan konsep yang terdapat pada permasalahan. Pendekatan ini merupakan yang paling jelas dan mudah diimplementasi.

Algoritma yang dibangun dengan pendekatan brute force biasanya tidak efisien dikarenakan biasanya membutuhkan banyak langkah tidak perlu untuk mendapatkan hasil yang diinginkan. Algoritma tersebut juga biasa disebut algoritma naif dikarenakan lebih mengandalkan kekuatan pemrosesan dibandingkan dengan jalur pemrosesan yang efisien.

Pendekatan brute force ini biasanya tidak digunakan untuk menyelesaikan masalah, tetapi biasanya hanya untuk referensi suatu pendekatan lain yang lebih efisien dan cepat dibandingkan pendekatan brute force. Pendekatan brute force hanya dapat digunakan secara realistis dalam permasalahan yang relatif kecil, dikarenakan dalam permasalahan yang besar, pendekatan brute force ini dapat menghasilkan hasil dalam waktu yang sangat lama, bahkan hingga berbulan-bulan atau bertahun-tahun.

Dalam pengimplementasian algoritma dengan pendekatan brute force harus memperhatikan beberapa hal agar sebuah solusi dapat terdefinisi. Hal tersebut adalah sebagai berikut,

- Definiteness
Setiap langkah pada proses harus dinyatakan secara presisi dan tepat

- Effective Compatibility
Setiap langkah dalam proses dapat dilakukan oleh komputer
- Finiteness
Program memiliki batas dan pada akhirnya akan diterminasi

B. Dynamic Programming

Dynamic programming atau program dinamis adalah metode pemecahan masalah dengan cara memecah permasalahan menjadi masalah kecil yang bertahap sehingga dapat dipandang sebagai permasalahan yang berurutan dan berulang. Dynamic programming biasanya berdasar dari suatu algoritma yang mudah dimengerti lalu akan dikembangkan atau di ubah agar dapat dilakukannya *tabulation* atau *memoization*.

Permasalahan dynamic programming mempunyai karakteristik tertentu. Berikut adalah karakteristik dari permasalahan dynamic programming,

- Persoalan dapat dibagi menjadi beberapa tahap yang dibedakan dengan pengambilan keputusan dari tiap langkah
- Masing-masing tahap terdiri dari sejumlah state yang berhubungan dengan tahap tersebut. Definisi state merupakan seluruh kemungkinan dari suatu tahap yang dibedakan dari tiap langkah yang diambil dari awal.

Berdasarkan cara pendekatannya, dynamic programming dibagi menjadi 2 yaitu top-down dan bottom-up. Berikut adalah penjelasan pembagiannya

a. Top-down

Pendekatan dengan top-down berarti memandang permasalahan dari atas (solusi) lalu bergerak kebawah (tahap-tahap) untuk menyelesaikan masalah. Untuk contoh, misal kita ingin sukses dan ingin punya banyak uang, maka kita harus mempunyai pekerjaan yang baik, maka kita harus lulus dari kuliah dengan baik, maka kita harus belajar dengan giat. Teknik yang digunakan dalam top-down biasa disebut dengan *memoization*, yang berarti menyimpan data langkah-langkah sebelumnya yang berkemungkinan dipakai dalam langkah setelahnya.

		i →				
		g	t	c	a	b
j ↓		0	0	0	0	0
	g	0	1	1	1	1
	x	0	1	1	1	1
	t	0	1	2	2	2
	x	0	1	2	2	2
	a	0	1	2	2	3
	b	0	1	2	2	3

Gambar 2.1 Penggambaran matrix DP top down
Sumber: freeCodeCamp.org

Berikut adalah keuntungan dari pendekatan top-down dari dynamic programming ini,

- o Mudah untuk diimplementasi
- o Tidak menyelesaikan permasalahan kecil jika tidak dibutuhkan
- o Debugging lebih mudah

Dan berikut adalah kerugian dari pendekatan top-down dari dynamic programming

- o Diselesaikan dengan cara rekursi, sehingga mengakibatkan overhead dalam program
- o Jika mempunyai tahapan yang panjang, berkemungkinan menghasilkan *stack overflow error*.

b. Bottom-up

Pendekatan dengan bottom-down berarti memandang permasalahan dari permasalahan kecilnya (bottom) ke permasalahan yang lebih besar (up). Untuk contoh, misal kita akan belajar, agar kita bisa mendapat nilai yang bagus sehingga dapat lulus kuliah dengan bagus, sehingga kita bisa mendapatkan pekerjaan yang bagus, pada akhirnya kita akan sukses dan mempunyai banyak uang. Teknik yang digunakan dalam bottom-up biasa disebut dengan *tabulation*, yang berarti menyimpan setiap state dalam suatu tabel agar dapat bisa diakses lagi kedepannya.

Berikut adalah keuntungan dari pendekatan bottom-up dari dynamic programming ini,

- o Lebih cepat dibandingkan top-down dikarenakan hanya menggunakan tabel atau array untuk menyimpan

Berikut adalah kerugian dari pendekatan bottom-up dari dynamic programming ini,

- o Sedikit susah diimplementasi dikarenakan harus merubah perspektif pada permasalahan
- o Harus menyelesaikan seluruh permasalahan kecil bahkan yang tidak dibutuhkan

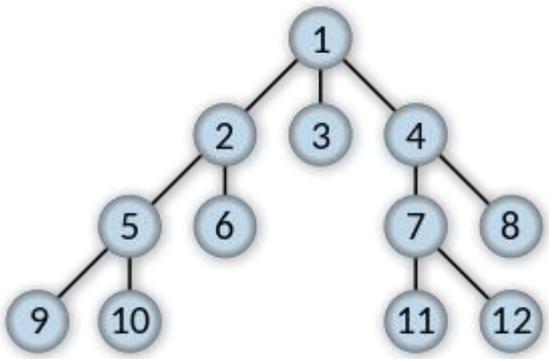
Secara ringkas, langkah-langkah untuk menyelesaikan permasalahan dynamic programming adalah sebagai berikut,

1. Karakteristikan permasalahan apakah top-down atau bottom-up, dan statusnya
2. Definisikan hubungan antar langkahnya
3. Hitung solusi menggunakan hubungan yang telah didefinisikan dan disimpan pada tabel untuk mempermudah
4. Cari solusi dari tabel yang sudah diisi pada langkah sebelumnya

C. Breadth First Search

Breadth First Search, atau disingkat BFS adalah salah satu algoritma penelusuran graf dimana simpul yang mempunyai kedalaman paling rendah mempunyai prioritas paling tinggi

dalam urutan pencarian. Untuk mempermudah pengimplementasian BFS, biasanya digunakan queue untuk mengatur urutan pencarian seluruh simpul.



Gambar 2.2 Penggambaran urutan pencarian BFS
Sumber: Wikipedia

BFS dipastikan dapat menemukan solusi dari pencarian bahkan jika secara teoritis pohon yang dicari berukuran tak hingga tetapi mempunyai solusi. Pada graf yang mempunyai ukuran berat sisi yang sama, pencarian menggunakan BFS dipastikan juga menghasilkan solusi paling optimal.

Dikarenakan aspek BFS yang menguntungkan dibandingkan algoritma penelusuran Depth First Search atau DFS yang lebih digunakan saat kondisi tertentu, algoritma BFS ini jauh lebih sering digunakan secara umum.

D. Mekanisme Diskon pada Platform Pemesanan Makanan Online

Pada platform pemesanan makanan online, terdapat banyak jenis diskon yang ada pada platform yang berbeda. Diskon yang ada pada platform dapat dibedakan menjadi 2, yaitu

- Diskon flat

Diskon ini bersifat konstan nilainya, walau harga menu ditambah. Diskon ini biasanya dibatasi dengan minimal pembelian menu tersebut. Strategi yang tepat untuk mengoptimalkan jenis diskon ini adalah meminimalisir harga menu untuk mendapatkan diskon tersebut.

- Diskon persentase

Diskon ini nilainya bergantung dengan harga menu yang dipesan. Diskon ini biasanya digambarkan dengan persentase yang tinggi tetapi dibatasi dengan nilai maksimal diskon. Strategi yang tepat untuk mengoptimalkan jenis diskon ini adalah memaksimalkan diskon sesuai persentase dan nilai maksimal dari diskon.

Dari banyak diskon dari berbagai platform, seluruh diskon dapat digeneralisir dengan berbagai parameter yang mempengaruhi proses optimisasinya. Berikut adalah parameter yang menggeneralisir diskon yang ada,

- Persentase diskon

Merupakan penentu nilai diskon yang paling berpengaruh. Pada diskon flat nilai ini tidak ada, tetapi

dapat dianggap sebagai 100%, yang akan dibatasi oleh maksimal diskon.

- Minimal pembelian

Merupakan batas bawah dari harga menu, yang bisa juga menjadi batas bawah diskon setelah dikalikan dengan persentase diskon.

- Maksimal diskon

Merupakan batas atas dari diskon, yang bisa juga menjadi batas atas diskon setelah dibagi dengan persentase diskon. Ketika diskon menyentuh nilai maksimal ini, lebih baik untuk tidak menambah harga menu karena tidak akan berpengaruh pada nilai diskon.



Gambar 2.3 Contoh diskon dari salah satu platform
Sumber: ShopeeFood

III. IMPLEMENTASI

A. Pengelompokan Jenis Produk

Terdapat beberapa jenis produk makanan yang dapat dipesan oleh pengguna. Order dari pengguna harus memiliki isi menu yang masuk akal, tidak hanya memenuhi diskon yang optimal. Jenis produk ini juga menentukan jumlah porsi order dari pesanan pengguna. Berikut adalah pengelompokan jenis produk dan beberapa aturannya,

1. Makanan berat

Jumlah makanan berat yang berada pada pesanan menentukan jumlah porsi order tersebut.

2. Minuman

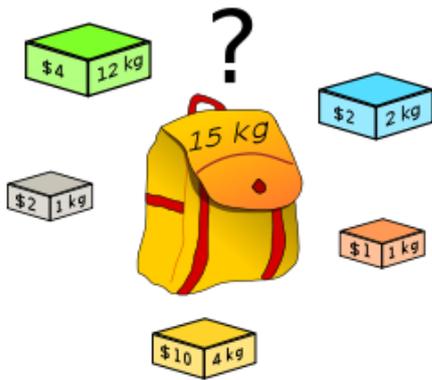
Dengan anggapan pengguna tidak terlalu haus, jumlah minuman pasti lebih sedikit atau sama dengan dibanding makanan berat.

3. Tambahan

Merupakan menu kecil seperti tahu atau topping yang menambahkan harga, tetapi tidak mengubah porsi dari pesanan.

B. Abstraksi permasalahan

Dalam penyelesaian permasalahan kita menggunakan pendekatan dynamic programming yang dikembangkan dari pendekatan brute force lebih tepatnya, exhaustive search. Permasalahan diskon ini mempunyai bentuk yang mirip dengan permasalahan integer knapsack dimana yang dioptimisasi adalah parameter yang diinginkan oleh pengguna seperti nilai diskon, persentase diskon per total harga, atau harga per porsi.



Gambar 3.1 Visualisasi permasalahan Integer Knapsack
Sumber : Wikipedia

Dimana produk dimisalkan sebagai barang yang akan dimasukkan ke order pengguna yang dimisalkan sebagai tas yang mempunyai kapasitas seperti maksimal diskon. Tetapi pada pembatasan, hanya akan dibatasi ketika diskon telah sama dengan nilai maksimal diskon.

Untuk mempermudah abstraksi, dapat digunakan paradigma kelas untuk menggambarkan seluruh prosedur yang terjadi. Berikut adalah kelas yang terlibat dan kegunaannya dalam program ini,

- Diskon
Berfungsi untuk menyimpan nilai diskon agar dapat diakses secara bebas oleh objek order atau pesanan pengguna.
- Menu
Merupakan objek produk yang menyimpan nama produk, jenis produk, dan harga tiap produk.
- Order
Merupakan objek kumpulan menu. Dapat berisi prosedur untuk menghitung jumlah harga, menghitung jumlah porsi, validasi apakah menu valid atau apakah titik maksimal sudah dicapai atau belum.
- Program
Berfungsi untuk menyimpan variabel statik yang digunakan oleh seluruh objek dalam program, seperti diskon yang digunakan, array hasil pencarian, atau graf pencarian.

C. Implementasi Algoritma

Dikarenakan bentukan permasalahan ini adalah seperti *unbounded knapsack* yaitu permasalahan integer knapsack tetapi dengan barang yang mempunyai jumlah banyak dan dapat sama, sehingga dimensi pencarian adalah m^n . Oleh karena itu, untuk melakukan iterasi, dibutuhkan algoritma breadth first search dengan heuristik agar dapat diiterasi dengan terstruktur dan menemukan seluruh perhitungan.

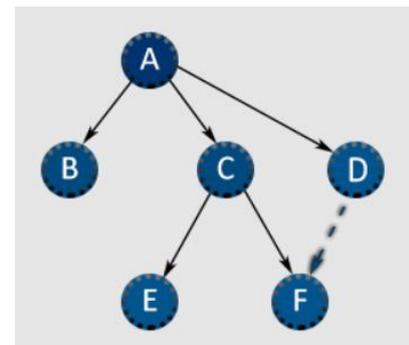
Selain itu, setiap hasil perhitungan akan disimpan hasilnya pada array (atau hash map) yang bertindak sebagai tabel pada proses tabulasi di dynamic programming. Selain untuk

menyimpan hasil, array (atau hash map) ini bersifat untuk memastikan tidak ada perhitungan yang dilakukan 2 kali sesuai agar algoritma berjalan efisien sesuai dengan paradigma dynamic programming.

Dikarenakan kompleksitas algoritma yang berbentuk eksponensial, maka dapat dilakukan pengelompokan produk yang mempunyai harga yang sama untuk mengurangi jumlah produk yang diperhitungkan. Saat setelah pencarian, maka dapat dilakukan ekspansi menu secara kombinasi menu yang tadi dikelompokkan.

Proses algoritma atau tahap tahap yang dapat digunakan untuk menyelesaikan permasalahan ini adalah sebagai berikut,

1. Mengumpulkan seluruh menu untuk menentukan domain pencarian
2. Melakukan iterasi pada graf seperti berikut,
 - a. Memastikan menu adalah menu yang valid memenuhi batasan yang sudah ditentukan.
 - b. Menghitung total harga, apakah sudah memenuhi diskon maksimum atau belum dan menyimpan hasil perhitungan pada array (atau hash map) untuk heuristik dan perhitungan selanjutnya.
 - c. Jika belum, ekspansi simpul untuk menghasilkan simpul anak berdasarkan simpul awal dengan menambahkan menu.
3. Jika proses exhaustive search telah selesai, lakukan pengurutan dengan parameter sesuai dengan masukan pengguna.



Gambar 3.2 Visualisasi graf hasil pencarian program

Sumber: Penggambaran penulis

IV. HASIL

Dikarenakan keterbatasan kekuatan komputasi, maka pengujian akan menggunakan data palsu atau modifikasi dari data menu asli berdasarkan diskon nyata yang berada pada toko tersebut. Dan juga dikarenakan hasil pencarian sangat besar, maka hanya ditampilkan sebagian yang menunjukkan perbedaan.

Pengujian untuk mendapatkan hasil program dilakukan pada mesin penulis dengan program yang ditulis dalam bahasa python,

- Diskon: 60%, min Rp.40.000, max Rp. 25.000
- Menu
 - o Nasi telur, Rp.13.000
 - o Nasi telur sosis, Rp. 14.500
 - o Nasi telur sarden, Rp. 16.500
 - o Magelangan, Rp. 18.000
 - o Teh manis, Rp. 4.000
 - o Teh tarik, Rp. 7.000
 - o Extra sosis Rp. 2.000
 - o Extra omlet Rp. 7.500
- Hasil

Berdasarkan harga per porsi dengan porsi lebih dari 2

Price: 22000 3 - Nasi Telur 1 - Teh Manis 2 - Sosis	Normal Price: 47000	Discount%: 53.19	Price per portion: 7333.33
Price: 22000 3 - Nasi Telur 4 - Sosis	Normal Price: 47000	Discount%: 53.19	Price per portion: 7333.33
Price: 22000 3 - Nasi Telur 2 - Teh Manis	Normal Price: 47000	Discount%: 53.19	Price per portion: 7333.33
Price: 29000 4 - Nasi Telur 1 - Sosis	Normal Price: 54000	Discount%: 46.30	Price per portion: 7250.00
Price: 21500 2 - Nasi Telur 1 - Nasi Telur Sarden 2 - Sosis	Normal Price: 46500	Discount%: 53.76	Price per portion: 7166.67
Price: 21500 2 - Nasi Telur 1 - Nasi Telur Sarden 1 - Teh Manis	Normal Price: 46500	Discount%: 53.76	Price per portion: 7166.67
Price: 21500 2 - Nasi Telur 1 - Nasi Telur Sosis 3 - Sosis	Normal Price: 46500	Discount%: 53.76	Price per portion: 7166.67
Price: 21500 2 - Nasi Telur 1 - Nasi Telur Sosis 1 - Teh Manis 1 - Sosis	Normal Price: 46500	Discount%: 53.76	Price per portion: 7166.67
Price: 21500 3 - Nasi Telur 1 - Omlet	Normal Price: 46500	Discount%: 53.76	Price per portion: 7166.67
Price: 28500 3 - Nasi Telur 1 - Nasi Telur Sosis	Normal Price: 53500	Discount%: 46.73	Price per portion: 7125.00

Berdasarkan harga total dengan porsi lebih dari 2

Price: 33500 2 - Nasi Telur Sosis 1 - Magelangan 1 - Teh Manis 1 - Omlet	Normal Price: 58500	Discount%: 42.74	Price per portion: 11166.67
Price: 33500 1 - Nasi Telur 1 - Nasi Telur Sosis 1 - Magelangan 1 - Teh Tarik 3 - Sosis	Normal Price: 58500	Discount%: 42.74	Price per portion: 11166.67
Price: 33500 1 - Nasi Telur 1 - Nasi Telur Sosis 1 - Magelangan 1 - Teh Manis 1 - Teh Tarik 1 - Sosis	Normal Price: 58500	Discount%: 42.74	Price per portion: 11166.67
Price: 33500 2 - Nasi Telur 1 - Nasi Telur Sosis 1 - Magelangan	Normal Price: 58500	Discount%: 42.74	Price per portion: 8375.00
Price: 33500 2 - Nasi Telur 1 - Magelangan 1 - Teh Tarik 1 - Omlet	Normal Price: 58500	Discount%: 42.74	Price per portion: 11166.67
Price: 34000 1 - Nasi Telur 2 - Magelangan 1 - Teh Manis 3 - Sosis	Normal Price: 59000	Discount%: 42.37	Price per portion: 11333.33
Price: 34000 1 - Nasi Telur 2 - Magelangan 2 - Teh Manis 1 - Sosis	Normal Price: 59000	Discount%: 42.37	Price per portion: 11333.33

Berdasarkan harga total dengan porsi 2

Price: 17000 1 - Nasi Telur Sosis 1 - Nasi Telur Sarden 1 - Teh Tarik 2 - Sosis	Normal Price: 42000	Discount%: 59.52	Price per portion: 8500.00
Price: 17000 2 - Nasi Telur Sarden 1 - Teh Tarik 1 - Sosis	Normal Price: 42000	Discount%: 59.52	Price per portion: 8500.00
Price: 17000 1 - Nasi Telur Sosis 1 - Nasi Telur Sarden 1 - Teh Manis 1 - Teh Tarik	Normal Price: 42000	Discount%: 59.52	Price per portion: 8500.00
Price: 17000 1 - Nasi Telur 1 - Nasi Telur Sosis 1 - Teh Tarik 1 - Omlet	Normal Price: 42000	Discount%: 59.52	Price per portion: 8500.00
Price: 17000 2 - Nasi Telur Sosis 1 - Teh Manis 1 - Teh Tarik 3 - Sosis	Normal Price: 42000	Discount%: 59.52	Price per portion: 8500.00
Price: 17000 2 - Nasi Telur Sosis 1 - Teh Manis 1 - Teh Tarik 1 - Sosis	Normal Price: 42000	Discount%: 59.52	Price per portion: 8500.00
Price: 17000 2 - Nasi Telur 2 - Teh Manis 4 - Sosis	Normal Price: 42000	Discount%: 59.52	Price per portion: 8500.00
Price: 17000 2 - Nasi Telur 2 - Teh Tarik 1 - Sosis	Normal Price: 42000	Discount%: 59.52	Price per portion: 8500.00
Price: 16600.0 1 - Nasi Telur Sosis 1 - Magelangan 1 - Teh Tarik 1 - Sosis	Normal Price: 41500	Discount%: 60.00	Price per portion: 8300.00

V. KESIMPULAN

Dynamic programming merupakan pendekatan yang efektif dalam menyelesaikan masalah, tetapi biasanya harus dibantu dengan algoritma/pendekatan lain seperti brute force atau greedy agar dapat optimal dalam pencarian solusi. Hal ini dikarenakan dynamic programming biasanya hanya berfungsi sebagai peningkatan algoritma yang sudah ada dengan melakukan konsep tabulasi atau memoisasi.

Berdasarkan hasil uji yang dilakukan pada bab sebelumnya, dapat disimpulkan bahwa dapat dicari optimisasi dari sebuah menu dengan penggabungan teori brute force, dynamic programming, dan breadth first search.

Dapat diperhatikan juga terdapat perilaku dimana saat perbedaan harga menu mendekati minimal pembelian diskon, akan ditambahkan menu extra untuk memenuhi minimum pembelian. Perilaku tersebut merupakan salah satu perilaku yang dilakukan saat melakukan pembelian makanan pada platform online ini.

LINK VIDEO YOUTUBE

https://youtu.be/1FFtOrT_A5I

UCAPAN TERIMA KASIH

Penulis ingin berterimakasih kepada seluruh pengajar dari mata kuliah IF2211 yang telah mengajar sepenuh hati dan memberikan ilmu tentang strategi algoritma terutama tentang dynamic programming yang digunakan dalam penulisan makalah ini.

REFERENSI

- [1] Sharma, Ashwin P, <https://www.freecodecamp.org/news/dynamic-programming-made-easy/>. Diakses pada 21 Mei 2022
- [2] Gupta, Shubham, <https://www.geeksforgeeks.org/unbounded-knapsack-repetition-items-allowed/>. Diakses pada 21 Mei 2022
- [3] Munir, Rinaldi, [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-\(2022\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2021-2022/Algoritma-Brute-Force-(2022)-Bag1.pdf). Diakses pada 21 Mei 2022
- [4] Munir, Rinaldi, <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/BFS-DFS-2021-Bag2.pdf>. Diakses pada 21 Mei 2022
- [5] Munir, Rinaldi, <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Program-Dinamis-2020-Bagian1.pdf>. Diakses pada 21 Mei 2022

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, 20 Mei 2022



Rifqi Naufal Abdjul
13520062