

Mencari Rute Silahturami Hari Raya Dalam Memaksimalkan THR dengan Algoritma Branch And Bound

Januar Budi Ghifari - 13520132
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail : 13520132@std.stei.itb.ac.id

Abstract—Di Hari Raya Idul Fitri, Silahturami menjadi hal yang biasa dilakukan oleh umat Islam di Indonesia. Pada kesempatan ini juga biasanya tiap – tiap keluarga akan membagikan THR yang mana sudah menjadi budaya bagi beberapa orang. Untuk memaksimalkan THR yang didapatkan, dapat dilihat sebagai knapsack problem yang bisa diselesaikan dengan algoritma Branch and Bound.

Keywords—branch and bound, optimasi, THR, rute, knapsack

I. PENDAHULUAN

Hari Raya Idul Fitri merupakan waktu yang ditunggu-tunggu oleh seluruh umat di Indonesia. Hari di mana seluruh anggota keluarga berkumpul setelah melakukan puasa selama sebulan penuh. Hari Raya Idul Fitri ini juga erat kaitannya dengan silahturami dengan maksud saling meminta maaf antar umat. Tidak hanya dengan keluarga di rumah, tetapi juga dengan tetangga sekitar.



Gambar 1. Ilustrasi Hari Raya
(ilustrasi Lebaran. foto/istockphoto)

Dalam proses silahturami ini juga, biasanya dilakukan pembagian THR (Tunjangan Hari Raya). THR ini diberikan sebagai rasa syukur serta budaya yang menjadi ciri khas dari hari raya Idul Fitri. Pembagian THR ini juga menjadi salah satu faktor utama yang mendorong kita untuk bersilahturami ke tetangga sekitar. Semakin banyak THR yang didapat, semakin berkesan pula hari raya pada tahun itu.

Dalam *hunting* THR ini, kita memiliki tujuan yaitu meraih nominal sebesar – besarnya sebelum *adzan dzuhur* berkumandang, karena setelah *adzan dzuhur* biasanya orang-orang akan istirahat setelah lelah menyantap oper dan bersalaman. Penyelesaian masalah ini bisa digambarkan sebagai *knapsack problem* dan untuk mengambil solusi optimal digunakan algoritma branch and bound

II. LANDASAN TEORI

A. Algoritma Branch and Bound

Algoritma branch and bound merupakan salah satu algoritma pencarian yang digunakan untuk menyelesaikan persoalan yang melibatkan optimisasi, yaitu untuk meminimalkan atau memaksimalkan suatu fungsi objektif dan tidak melanggar batasan atau *constraint* yang diberikan oleh persoalan tersebut. Algoritma ini merupakan gabungan antara algoritma breadth first search (BFS) dan least cost search. BFS murni dengan algoritma branch and bound ini memiliki perbedaan yang berada pada ekspansi simpulnya. Pada BFS, simpul yang diekspansi akan mengikuti aturan FIFO (First In First Out) sedangkan pada algoritma branch and bound, setiap simpul akan diberi sebuah nilai bobot (cost) terlebih dahulu lalu simpul yang akan diekspansi adalah simpul yang memiliki bobot paling optimal. Persoalan yang melibatkan algoritma branch and bound biasanya direpresentasikan dengan sebuah pohon atau graf. Proses pencarian nilai optimum pada algoritma ini dilakukan dengan menggunakan pohon ruang status dengan setiap simpul memiliki bobot tertentu.

Berdasarkan prinsipnya, algoritma branch and bound terdiri atas 2 hal yaitu :

a. Algoritma akan membangkitkan simpul-simpul anak pada pohon ruang status dari simpul yang sedang diperiksa. Setiap simpul anak akan memiliki variansi tertentu dari simpul orang tuanya dan juga memiliki sebuah bobot yang merepresentasikan biaya yang dibutuhkan untuk mencapai simpul anak dari simpul asal. Proses ini disebut dengan *branching*.

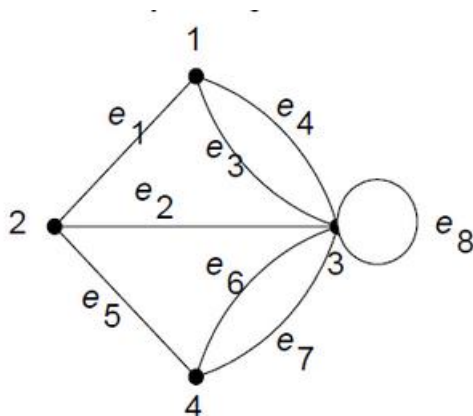
b. Algoritma akan melakukan pemangkasan pada jalur atau simpul yang dianggap tidak lagi mengarah pada solusi yang

diinginkan. Pemangkasan ini dilakukan menggunakan fungsi pembatas. Proses ini disebut sebagai bounding.

Algoritma branch and bound memiliki beberapa fungsi pembatas yang menerapkan pemangkasan pada jalur yang tidak mengarah ke solusi. Terdapat beberapa kriteria pemangkasan yaitu sebagai berikut :

- Nilai simpul tidak lebih baik dari nilai terbaik yang ada
- Simpul tidak merepresentasikan solusi yang layak karena terdapat batasan atau constraint yang dilanggar
- Solusi pada simpul hanya terdiri atas satu titik tidak ada pilihan lain lalu bandingkan nilai fungsi objektif dengan solusi terbaik yang ada kemudian ambil yang terbaik

B. Graf



Gambar 2. contoh graf
(slide perkuliahan)

Graf adalah sebuah representasi diagram yang biasa digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut. Ggraf kerap didefinisikan sebagai kumpulan titik yang dihubungkan oleh garis-garis atau sisi. Sedangkan definisi matematisnya, graf (G) adalah $G = (V,E)$ dimana V = himpunan tidak kosong dari simpul-simpul = $\{v1,v2,v3,...,vn\}$ dan E = himpunan sisi yang menghubungkan sepasang simpul = $\{e1,e2,e3,...,en\}$.

Knapsack

Knapsack Problem merupakan sebuah masalah di mana seseorang berhadapan dengan persoalan optimasi pemilihan atau penempatan item ke dalam suatu tempat atau wadah yang memiliki kapasitas terbatas. Optimasi ini bertujuan untuk mendapatkan solusi agar pemilihan atau penempatan item yang ditempatkan ini dapat menghasilkan keuntungan semaksimal mungkin. Setiap item memiliki berat dan nilai tertentu sehingga total berat dari item atau barang yang ditempatkan tidak melebihi kapasitas atau batasan dari Knapsack dan nilai yang didapatkan dapat maksimal. Terdapat beberapa jenis Knapsack Problem yaitu di antaranya :

- 0/1 Knapsack problem yang dimana tiap item atau barang hanya tersedia sebanyak 1 unit, sehingga menyisakan pilihan untuk diambil atau dilepaskan begitu saja.
- Fracksional knapsack problem yang dimana item atau barang hanya bisa dibawa sebagian. Jenis problem ini bisa masuk akal jika barang yang ada bisa dibagi-bagi seperti tepung, gula, dan lain-lain.
- Bounded Knapsack problem yang dimana masing-masing item atau barang tersedia dalam n unit yang jumlahnya terbatas.
- Unbounded Knapsack problem yang dimana masing-masing item atau barang yang tersedia jumlahnya minimal dua unit atau bahkan tak terbatas.

III. IMPLEMENTASI

Dalam menyelesaikan permasalahan ini, kita dapat memodelkan sampel – sampel yang kita miliki, di konteks ini adalah rumah tetangga yang akan dikunjungi untuk silaturahmi, sebagai knapsack problem. selanjutnya untuk mencari rute yang optimal, digunakan Algoritma branch and bound

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

A. Persiapan dan Sampel data

Menggunakan ilustrasi berdasarkan pengalaman saat hari raya kemarin, Penulis membatasi waktu yang digunakan untuk berburu THR menjadi 2 jam atau perkiraan dari jam 10 sampai waktu adzan dzuhur, karena setelah dzuhur biasanya orang – orang cenderung beristirahat di rumah.

Untuk sampel data rumah yang akan dikunjungi, penulis memperkecil lingkup sampel menjadi 7 rumah (termasuk rumah penulis) untuk mempermudah proses pengolahan data.

Tiap rumah yang dikunjungi akan dibedakan dengan huruf, dan memiliki nominal perkiraan THR yang berbeda – beda :

Rumah	THR (- ribu rupiah)
O	0
A	300
B	250
C	500
D	300
E	100

F	200
----------	-----

Gambar 3. Tabel perkiraan THR

Tiap – tiap rumah yang dikunjungi memiliki estimasi waktu silaturahmi yang berbeda – beda juga (ngobrol + maaf – maafan) :

Rumah	Waktu Silaturahmi (menit)
O	0
A	20
B	15
C	35
D	20
E	10
F	20

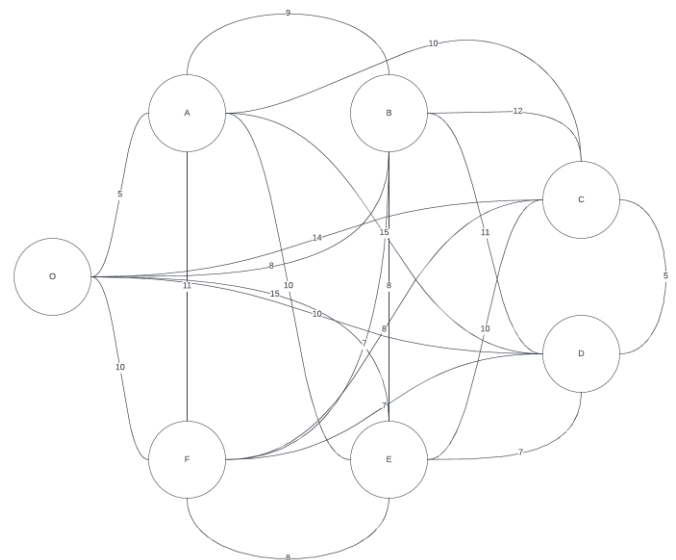
Gambar 4. Tabel waktu silaturahmi

Tiap – Tiap rumah akan dihitung lama tempuhnya (menit) dari setiap rumah :

	O	A	B	C	D	E	F
O	-	5	8	14	10	15	10
A	5	-	9	10	15	10	11
B	8	9	-	12	11	8	7
C	14	10	12	-	5	10	8
D	10	15	11	5	-	7	7
E	15	10	8	10	7	-	8
F	10	11	7	8	7	8	-

Gambar 4. Tabel jarak tiap rumah

Dengan table jarak tiap rumah, kita bisa membuat graf dengan simpul yang melambangkan rumah – rumah :



Gambar 5. graf perjalanan berdasarkan gambar 4

B. Knapsack

Dalam memodelkan permasalahan menggunakan metode knapsack, akan dilakukan pencarian rute paling optimal menggunakan pohon status yang memiliki cost di tiap simpulnya yang menyatakan batas atas. Ekspansi dilakukan di simpul yang memiliki cost terbesar.

Perhitungan cost dari sebuah simpul akan diterapkan dengan rumus :

$$C(i) = F + (K - W) \frac{P_i}{W_{i-1,i}}$$

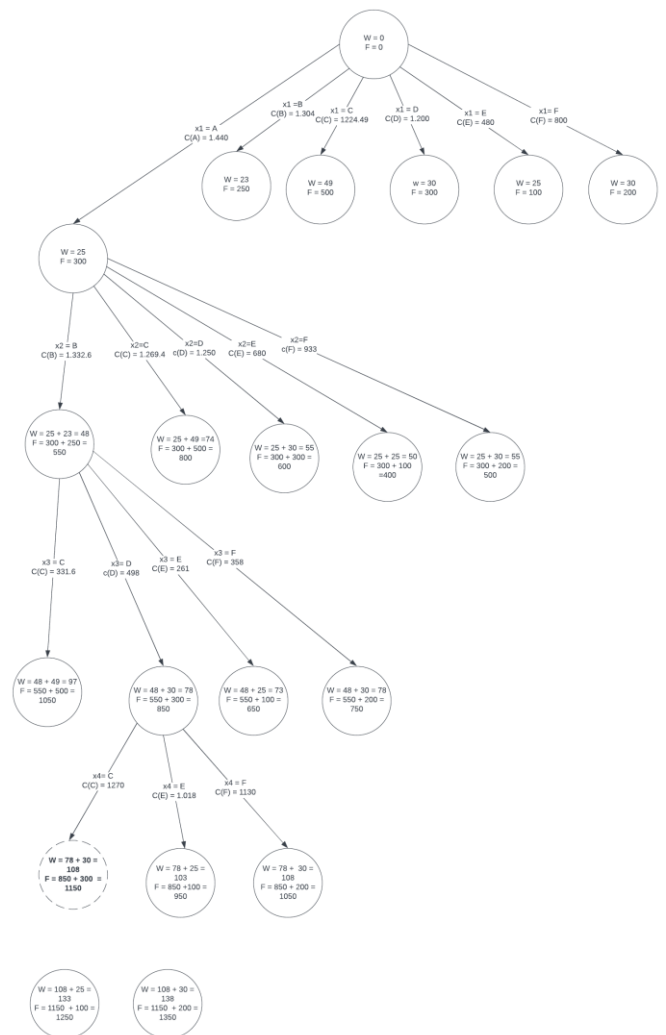
Dengan :

- F = Keuntungan saat ini
- K = Total Waktu Tersedia
- W = Total Waktu Terpakai
- $W_{i-1,i}$ = Waktu yang digunakan di simpul i (Waktu Silaturahmi + Waktu Perjalanan dari simpul sebelumnya)
- P = Keuntungan

Berikut adalah data inisialisasinya :

Rute	P	Jalan	Silahturami	W	P/W
O-A	300.00	5.00	20.00	25.00	12.00
O-B	250.00	8.00	15.00	23.00	10.87
O-C	500.00	14.00	35.00	49.00	10.20
O-D	300.00	10.00	20.00	30.00	10.00
O-E	100.00	15.00	10.00	25.00	4.00
O-F	200.00	10.00	20.00	30.00	6.67
A-B	250.00	8.00	15.00	23.00	10.87
A-C	500.00	14.00	35.00	49.00	10.20
A-D	300.00	10.00	20.00	30.00	10.00
A-E	100.00	15.00	10.00	25.00	4.00
A-F	200.00	10.00	20.00	30.00	6.67
B-C	500.00	14.00	35.00	49.00	10.20
B-D	300.00	10.00	20.00	30.00	10.00
B-E	100.00	15.00	10.00	25.00	4.00
B-F	200.00	10.00	20.00	30.00	6.67
C-D	300.00	10.00	20.00	30.00	10.00
C-E	100.00	15.00	10.00	25.00	4.00
C-F	200.00	10.00	20.00	30.00	6.67
D-E	100.00	15.00	10.00	25.00	4.00
D-F	200.00	10.00	20.00	30.00	6.67
E-F	200.00	10.00	20.00	30.00	6.67

Gambar 6. Tabel data inisiasi



Gambar 7. pohon solusi

C. Fungsi Pembatas

Fungsi pembatas yang digunakan di sini adalah waktu yang tidak melebihi 2 jam. bila ada simpul yang melanggar fungsi pembatas, maka simpulnya akan dimatikan.

Pengecekan dilakukan dengan persamaan:

$$t_0 + t_i \leq T$$

Dimana:

t_0 : waktu yang telah terpakai hingga saat ini

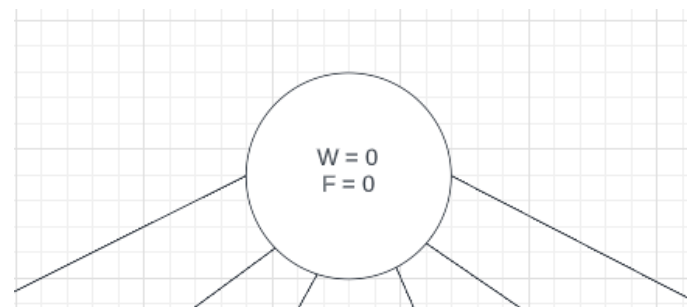
t_i : waktu yang dibutuhkan untuk menjangkau simpul terkait

T : waktu maksimal yang dibolehkan, dimana dalam hal ini adalah 120 menit

D. Implementasi Branch and Bound

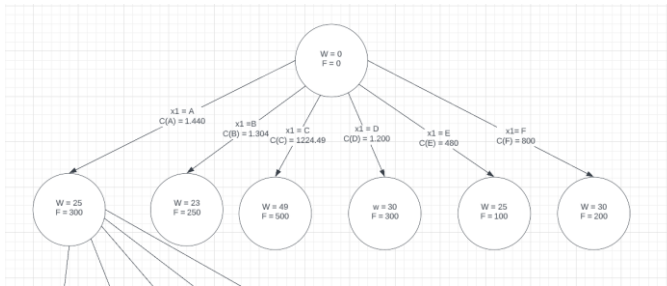
Dengan menggunakan algoritma branch and bound, ditemukan solusi sebagai berikut pada pohon ruang status:

Penjelasan Rinci Sebagai Berikut :



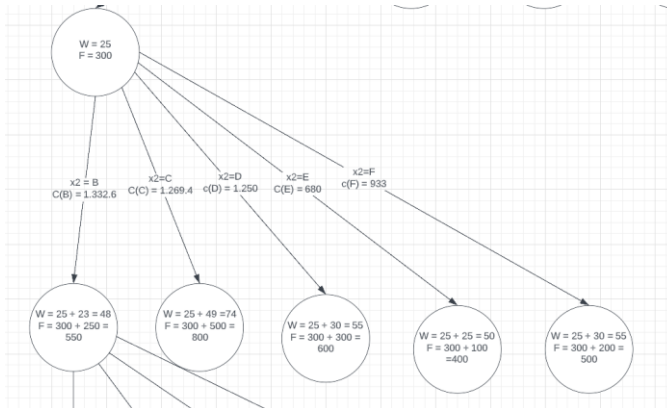
Gambar 8. simpul awal

Awalnya, simpul awal yaitu rumah kita akan membangkitakan semua simpul – simpul yang merepresentasikan seluruh sampel rumah tetangga:



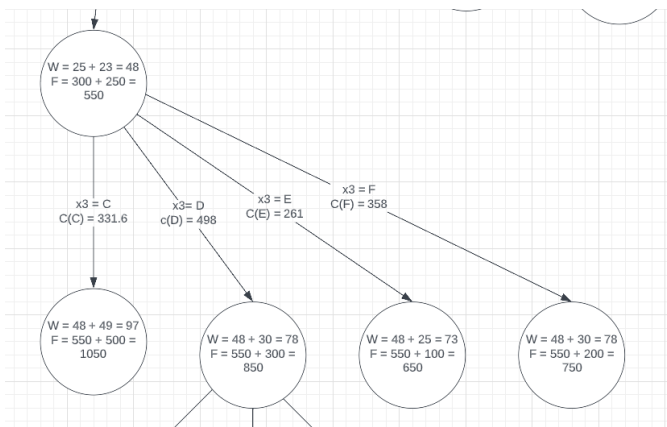
Gambar 9. simpul anak dari O

Didapatkan 6 simpul anak yang merupakan *possible* simpul selanjutnya, setelah itu masing-masing simpul dihitung costnya. akan dipilih $x_1 = A$ sebagai pembangkit simpul selanjutnya karena memiliki cost terbesar diantara x_1 lain.



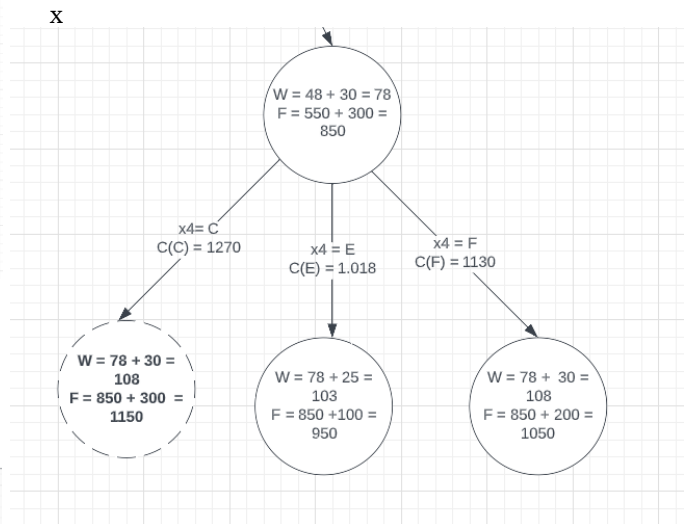
Gambar 10. simpul anak dari $x_1 = A$

simpul ini akan membangkitkan 5 simpul lain. dipilih $x_2 = b$ sebagai pembangkit simpul selanjutnya karena costnya terbesar diantara x_2 lain.



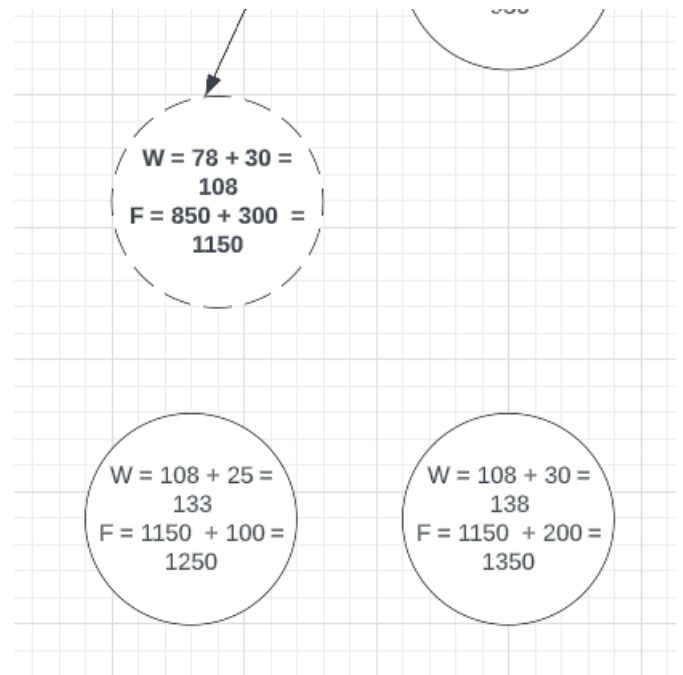
Gambar 11. simpul anak dari $x_2 = B$

$x_2 = b$ membangkitkan 4 simpul anak lagi. dipilih $x_3 = D$ sebagai pembangkit simpul selanjutnya



Gambar 12. simpul anak dari $x_3 = D$

$x_3 = D$ membangkitkan 3 simpul anak. pada titik ini, $x_4 = c$ akan dipilih menjadi simpul solusi selanjutnya akan tetapi tidak akan membuat simpul lanjutan.



Gambar 11. simpul akhir

bisa dilihat, bahwa simpul yang dibangkitkan dari $x_4 = e$ memiliki W total yang melebihi syarat dimana W harus lebih kecil atau sama dengan 120.

Maka, didapat rute yang yang dapat ditempuh dengan memaksimalkan THR yaitu, A-B-D-C dengan total THR = Rp. 1.150.000 dan memakan waktu 108 menit.

KESIMPULAN

Pada dasarnya, Hari Raya adalah hari dimana kita berkumpul Bersama keluarga dan saling bermaaf – maafan. THR hanya lah sebagai media untuk berbagi rezeki di hari kemenangan tersebut. Penulis sebagai mahasiswa yang sangat membutuhkan THR sebagai jalan untuk mempermudah kelangsungan bertahan hidup di ITB, membutuhkan strategi yang matang dan sangat menggambarkan penulis sebagai mahasiswa Teknik informatika. Penentuan rute silahturami ini dengan memperhatikan waktu tempuh, waktu silahturami, dan perkiraan THR untuk menentukan rute dengan THR yang paling optimal dirasa cukup memuaskan. Optimasi Knapsack dengan branch and bound ini juga dapat membantu mengarahkan ke hasil yang optimal dengan lebih sedikit ruang karena mematikan simpul yang tidak sesuai dan melakukan ekspansi dari simpul yang memiliki cost optimal, sehingga mampu menghemat waktu yang diperlukan untuk membuat keputusan.

ACKNOWLEDGMENT

Pertama-tama penulis ingin mengucapkan rasa syukur kepada Allah SWT. karena atas rahmat-Nya, penulis bisa menyelesaikan makalah Strategi Algoritma ini. Penulis juga ingin mengucapkan terimakasih kepada para dosen pembimbing Strategi Algoritma, terutama Bapak Rinaldi Munir selaku dosen K-03 yang telah membimbing kami serta memberikan ilmu - ilmu yang bermanfaat dan juga menjadi acuan dan refrensi dari isi makalah ini.

Terakhir penulis juga ingin mengucapkan terimakasih kepada seluruh rekan Teknik informatika yang telah berjuang Bersama menghadapi lautan tugas dan kuis di semester 4 ini. penulis harap kita semua dapat menjadi lebih baik kedepannya dan menyambut semester 5 dengan semangat sehat sentosa.

REFERENCES

- [1] SPADA KEMENDIKBUD. (2016, October 5). Algoritma Knapsack Problem. DosenIT.com. Diakses pada 23 Mei 2022, dari <https://dosenit.com/kuliah-it/rpl/algoritma-knapsack>
- [2] Sutiono. (2016, October 5). Algoritma Knapsack Problem. DosenIT.com. Diakses pada 23 Mei 2022, dari <https://dosenit.com/kuliah-it/rpl/algoritma-knapsack>
- [3] Pip Tools. (2015, October 24). Algoritma B&B (Branch and Bound). Pip Tools. Diakses pada 23 Mei 2022, dari <https://piptools.net/algoritma-bb-branch-and-bound/>
- [4] R. Munir. 2021. "Algoritma Branch & Bound (Bagian 4)". Sekolah Teknik Elektro dan Informatika ITB. Diakses pada 20 Mei 2022, dari <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branchand-Bound-2021-Bagian4.pdf>

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



Januar Budi Ghifari - 13520132