

Penggunaan Branch and Bound dalam Menentukan Pembelian Permainan pada Steam Sale

Fransiskus Febryan Suryawan (13519124)

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: 13519124@std.stei.itb.ac.id

Abstrak—Permainan video adalah salah satu media hiburan populer saat ini. Namun, keterbatasan finansial menyebabkan sulitnya memiliki permainan video secara legal. Oleh karenanya, dibutuhkan perencanaan keuangan sehingga dapat membeli permainan video secara legal yang memaksimalkan kesenangan yang didapat.

Kata Kunci—*branch and bound, steam, permainan*

I. PENDAHULUAN

Di zaman modern ini, permainan video adalah salah satu media interaktif yang sering digunakan sebagai hiburan. Popularitas permainan video menyebabkan kemunculan banyak permainan-permainan baru, baik yang inovatif maupun yang berupa plagiarisasi. Semua permainan video tersebut memiliki wadah yang menaungi keberadaannya serta memfasilitasi distribusi permainan video. Wadah yang saat ini ada dan cukup populer yaitu Steam, Origin, dan Epic. Wadah-wadah tersebut memfasilitasi pembelian permainan video secara digital, seperti dalam e-commerce.

E-commerce adalah kegiatan membeli atau menjual barang pada internet. Dalam wadah distribusi permainan video, barang yang dijual adalah permainan video itu sendiri. Pengguna dapat membayar sejumlah uang kepada pemilik wadah distribusi, kemudian menerima permainan video yang diinginkan secara digital. Hal ini sedikit berbeda dengan transaksi fisik di masa lalu, dimana pengguna mendatangi toko secara fisik lalu membayar sejumlah uang dan mendapatkan kopi dari permainan video secara fisik.

Wadah distribusi permainan video yang bertajuk Steam seringkali memberikan diskon besar bagi penggunanya. Ada beberapa diskon yang dilakukan tahunan oleh Steam, di antaranya adalah Summer Sale, Spring Sale, serta Winter Sale. Selain itu, Steam juga sering mengadakan diskon saat acara-acara tertentu, seperti saat Golden Week di Jepang tahun ini. Pemberian diskon-diskon ini dapat menarik pemain baru untuk menggunakan wadah Steam sebagai wadah utama untuk pembelian permainan video. Karena seringnya pemberian diskon besar, banyak pengguna yang membuat *countdown* untuk diskon besar selanjutnya, dengan memanfaatkan data tahun-tahun sebelumnya, seperti pada Gambar 1.



Gambar 1. *Countdown* untuk diskon besar selanjutnya, diakses pada 10 Mei 2021.

Pemberian diskon yang cukup sering pada wadah distribusi Steam menghasilkan banyak pengguna yang setia dalam menggunakan wadah tersebut. Pada umumnya, pengguna akan menunggu permainan yang ingin dibelinya diskon sebelum membeli permainan tersebut. Namun, seringkali karena keterbatasan finansial, tidak semua permainan video yang diinginkan dapat dibeli, dan harus membatasi diri dengan anggaran yang ada. Terkadang, pengguna akan memikirkan masalah ini cukup lama, yang menyebabkan pengguna melewatkan kesempatan untuk membeli saat diskon.

Penggunaan algoritma *Branch and Bound* dapat mempermudah penentuan permainan video yang akan dibeli. Dengan menentukan nilai kesenangan yang mungkin didapat untuk setiap permainan video, pemilihan permainan dapat dilakukan secara lebih pintar, dengan harapan permainan video yang dipilih dan dibeli dapat memaksimalkan kesenangan penggunanya.

II. LANDASAN TEORI

A. E-Commerce Pada Steam



Gambar 2. Laman Utama dari Steam, diakses pada tanggal 10 Mei 2021.

E-commerce atau perdagangan elektronik adalah aktivitas jual-beli barang yang terjadi melalui sistem elektronik seperti internet, televisi, dan jaringan komputer lainnya. Wadah distribusi Steam menyediakan layanan e-commerce yang digunakan untuk menjual dan mendistribusikan berbagai produk. Salah satu produk yang merupakan produk utama wadah ini adalah permainan video.

Pembelian produk pada wadah distribusi Steam dapat dilakukan dengan beberapa metode. Metode yang umumnya digunakan adalah dengan membeli kode *gift card* untuk *Steam Wallet*, kemudian menggunakan sejumlah uang yang telah masuk ke akun untuk membeli produk. Selain itu, pengguna juga dapat menggunakan kartu kredit untuk membeli produk secara langsung.

Setelah transaksi dilakukan, pengguna akan mendapat surel pemberitahuan bahwa transaksi berhasil, bersama dengan tanda terima. Pengguna juga akan menerima produk yang dapat langsung diakses melalui aplikasi Steam. Kepemilikan produk yang sudah dibeli bersifat permanen dan terikat pada akun, yang berarti selama pengguna memiliki akses pada akun tertentu maka pengguna tetap memiliki akses untuk setiap produk yang terikat akun tersebut.

Wadah distribusi Steam juga memberikan layanan pengembalian dana apabila pengguna merasa kurang puas dengan produk yang dibeli. Pengguna dapat mengajukan penguangan kembali dengan syarat produk digunakan masih kurang dari dua jam atau pengajuan penguangan kembali dilakukan kurang dari dua minggu sejak produk dibeli.

Selain produk-produk permainan, pengguna juga dapat ikut terlibat dengan pasar komunitas. Pasar ini merupakan bagian dari layanan yang ditawarkan Steam, dengan produk-produk yang beragam dan berasal dari permainan-permainan yang ada. Salah satu produk yang cukup banyak ditemui di pasar ini adalah *trading card*. Produk ini diperjualbelikan oleh pengguna dalam pasar, baik untuk melengkapi koleksi maupun untuk mencari keuntungan.

B. Branch and Bound

Algoritma *Branch and Bound* adalah algoritma yang dapat digunakan untuk mencari solusi optimum dari suatu persoalan. Algoritma ini dapat meminimalkan atau memaksimalkan fungsi objektif dengan memperhatikan batasan-batasan yang ada. Dua prinsip yang mendasari algoritma *Branch and Bound* adalah:

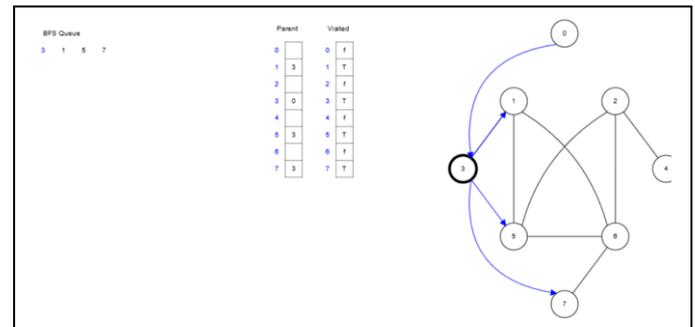
1. Perhitungan dilakukan secara rekursif, dimana masalah dipecah menjadi subpermasalahan yang lebih kecil sambil tetap menghitung nilai fungsi objektif yang terbaik. Proses ini disebut sebagai proses *branching*.
2. Algoritma akan menghitung nilai *bound*, yaitu nilai fungsi objektif terbaik saat ini, untuk memangkas jalur-jalur yang tidak mungkin mencapai nilai terbaik

Nilai *cost* dalam algoritma *Branch and Bound* umumnya merupakan nilai taksiran, yang digambarkan dalam persamaan

$$\hat{c}(i) = \hat{g}(i) + \hat{h}(i)$$

Dengan $\hat{c}(i)$ menyatakan *cost* total untuk simpul i , $\hat{g}(i)$ menyatakan *cost* untuk mencapai simpul i , dan $\hat{h}(i)$ menyatakan taksiran *cost* dari simpul i ke simpul tujuan.

Algoritma *Branch and Bound* merupakan algoritma yang menggabungkan algoritma *Breadth-First Search* (BFS) dan *Least Cost Search*. Algoritma BFS adalah algoritma *graph traversal* yang melakukan pencarian secara melebar, dimulai dari simpul awal dan dilanjutkan ke seluruh simpul-simpul tetangganya. Algoritma BFS menggunakan antrian untuk pembangkitan simpul ekspansi selanjutnya.



Gambar 3. Contoh visualisasi untuk BFS (<https://www.cs.usfca.edu/~galles/visualization/BFS.html>)

Algoritma *Branch and Bound* menggunakan BFS yang dimodifikasi, karena pembangkitan simpul ekspansi selanjutnya menggunakan nilai *cost* yang paling optimum. Pertama, simpul akar diekspansi. Kemudian setiap tetangga dari simpul tersebut dimasukkan dalam sebuah antrian prioritas berdasarkan *cost* yang diberikan, sehingga akan menjamin bahwa simpul yang diekspansi selanjutnya adalah simpul dengan *cost* optimum.

Dalam Algoritma *Branch and Bound*, terdapat fungsi pembatas. Fungsi pembatas ini digunakan untuk memangkas jalur yang tidak mengarah pada solusi optimum maupun jalur yang tidak menghasilkan solusi terbaik. Kriteria pemangkasan yang umum digunakan adalah:

1. Nilai cost tidak lebih baik dari nilai terbaik sejauh ini (*best solution so far*)
2. Simpul tidak merepresentasikan solusi yang mungkin karena ada batasan yang dilanggar.
3. Solusi pada simpul hanya terdiri atas satu titik, sehingga tidak ada pilihan lain. Bandingkan nilai fungsi objektifnya dengan solusi terbaik saat ini, lalu ambil yang terbaik.

III. PEMECAHAN MASALAH DENGAN *BRANCH AND BOUND*

Karena masalah yang diberikan adalah masalah optimasi, maka algoritma *Branch and Bound* dipilih untuk membantu menyelesaikan masalah. Dengan memanfaatkan data review dan harga yang tersedia pada Steam, pembangunan simpul-simpul yang mungkin dapat dilakukan berdasarkan *wishlist* seorang pengguna Steam.

Persoalan pengambilan permainan video ini kemudian dapat diterjemahkan menjadi serupa dengan persoalan *Integer Knapsack*, dimana keuntungan atau *profit* tiap objek menjadi nilai "review" baik suatu permainan, beban atau *weight* merupakan harga dari permainan, dan kapasitas *knapsack* adalah anggaran yang dimiliki pengguna.

Penerapan algoritma *Branch and Bound* akan dilakukan dengan mengatur *cost* simpul-simpul menjadi penjumlahan total uang yang sudah dikeluarkan dengan hasil kali sisa anggaran dan rasio "review" baik per uang yang dikeluarkan untuk permainan selanjutnya. Karena pada umumnya permainan video dijual dalam satuan lebih dari ribuan rupiah, maka dalam implementasinya anggaran dan harga permainan akan dibagi oleh 1.000, untuk menghindari masalah numerik.

Pohon ruang status yang terbentuk dalam penyelesaian ini berbentuk pohon biner, dimana cabang kiri menyatakan objek ke-*i* diambil dan cabang kanan menyatakan objek ke-*i* tidak diambil.

Sebagai contoh, apabila penulis memiliki 8 buah permainan yang ingin dibeli sebagai berikut, dengan data harga diambil pada 10 Mei 2021 pukul 11.37:

TABEL I. PERMAINAN DALAM *WISHLIST*

No	Data Permainan Video		
	Nama	Harga (ratus ribu rupiah)	Review Positif (%)
1.	Risk of Rain 2	120	97
2.	DOOM Eternal	263	90
3.	Ori and the Will of the Wisps	140	96
4.	AI*Shoujo/AI*少女	558	78

Lalu, andaikan penulis memiliki anggaran sebanyak 2 juta rupiah untuk membeli permainan-permainan tersebut. Algoritma *Branch and Bound* dapat diaplikasikan untuk memilih permainan yang akan dibeli.

Langkah-langkah untuk menyelesaikan permasalahan adalah dengan pertama hitung rasio "review" baik per uang yang dikeluarkan.

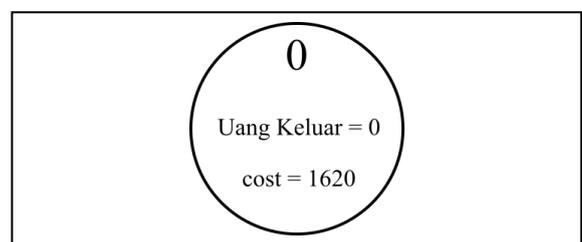
TABEL II. RASIO "REVIEW" BAIK PER UANG YANG DIKELUARKAN

No	Data Permainan Video			
	Nama	Harga (ratus ribu rupiah)	Review Positif (%)	Rasio
1.	Risk of Rain 2	120	97	0,81
2.	DOOM Eternal	263	90	0,34
3.	Ori and the Will of the Wisps	140	96	0,69
4.	AI*Shoujo/AI*少女	558	78	0,14

Setelah diketahui rasio-rasio "review" baik per uang yang dikeluarkan, bangkitkan simpul akar. Karena simpul akar belum ada permainan yang dipilih, maka

$$\hat{c}(0) = 0 + 2.000 \cdot 0,81 = 1620$$

Pohon ruang status yang terbentuk menjadi seperti pada gambar 4.

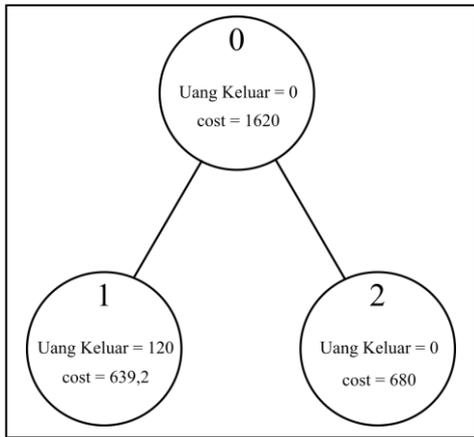


Gambar 4. pohon ruang status untuk simpul akar

Kemudian bangkitkan simpul selanjutnya, yaitu ketika permainan pertama diambil untuk simpul kiri dan permainan pertama tidak diambil untuk simpul kanan.

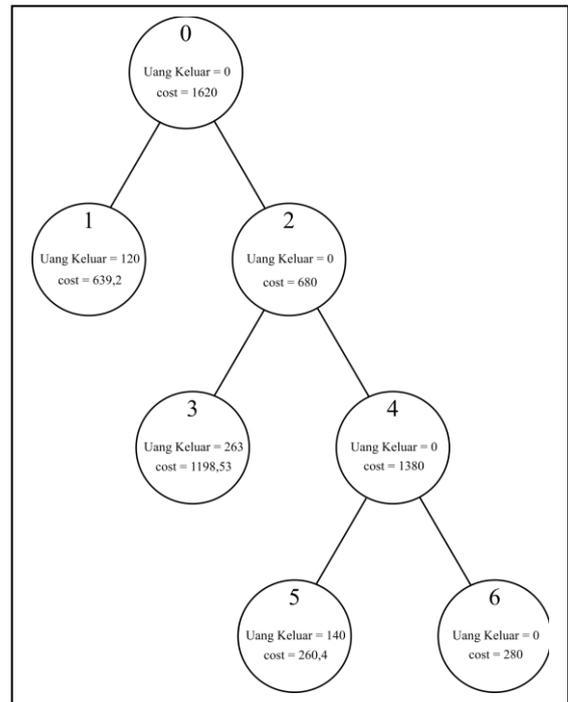
- Ketika permainan pertama tidak diambil:
 $\hat{c}(1) = 120 + 1.880 \cdot 0,34 = 759,2$
- Ketika permainan pertama diambil:
 $\hat{c}(2) = 0 + 2.000 \cdot 0,34 = 639,2$

Pohon ruang status yang telah dibangkitkan menjadi seperti pada gambar

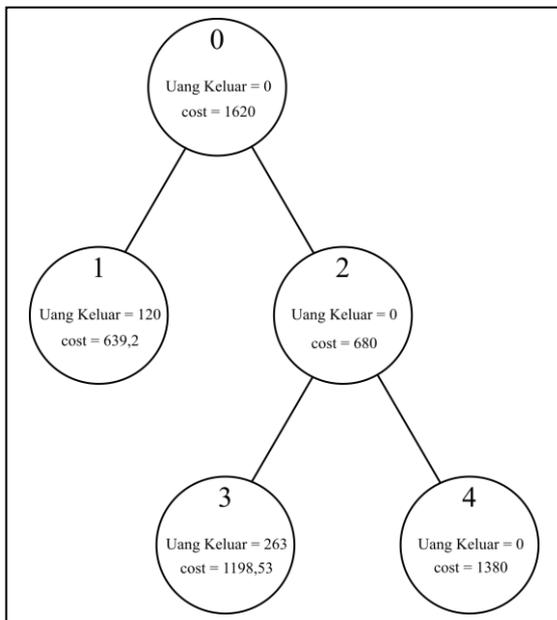


Gambar 5. Pohon ruang status untuk aras 1

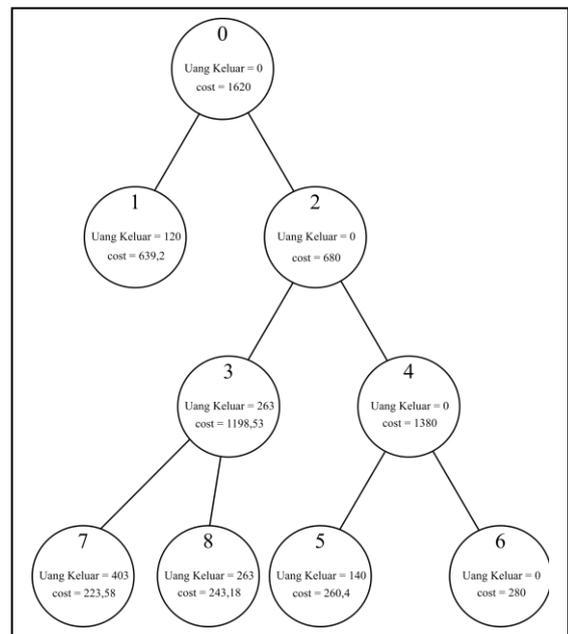
Karena simpul 2 memiliki cost yang lebih besar, maka simpul 2 adalah simpul yang diekspansi selanjutnya. Perhitungan cost dapat dilakukan dengan cara yang serupa dengan langkah-langkah sebelumnya. Langkah-langkah berikutnya akan disajikan dalam bentuk pohon yang sudah terbentuk per langkah.



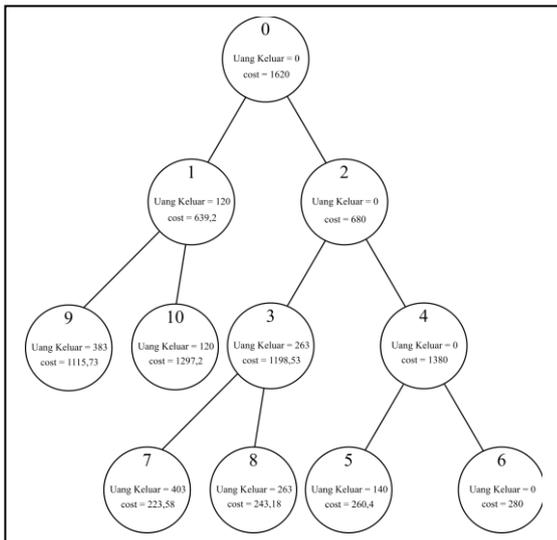
Gambar 7. Pohon ruang status untuk langkah 4



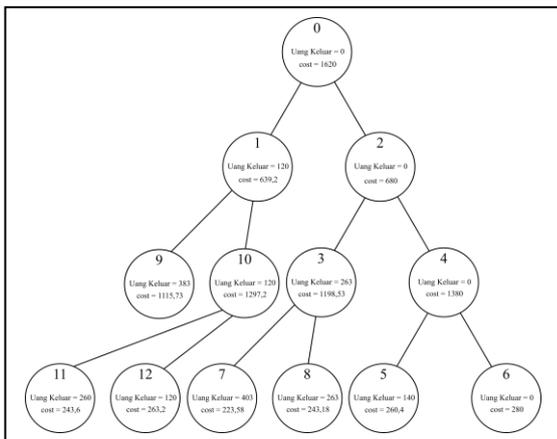
Gambar 6. Pohon ruang status untuk langkah 3



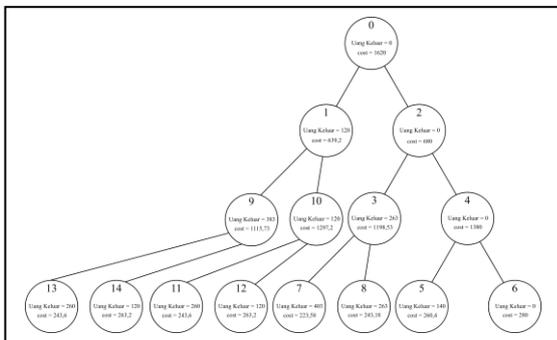
Gambar 8. Pohon ruang status untuk langkah 5



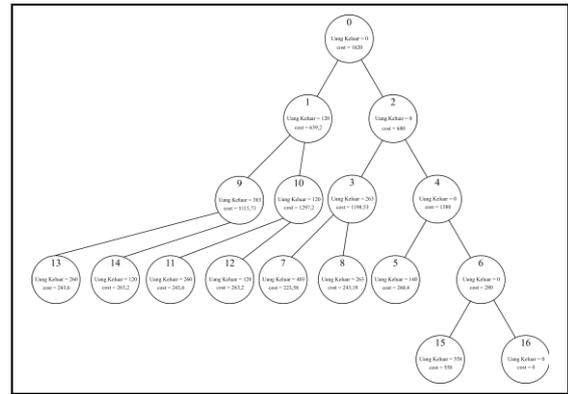
Gambar 9. Pohon ruang status untuk langkah 6



Gambar 10. Pohon ruang status untuk langkah 7



Gambar 11. Pohon ruang status untuk langkah 8



Gambar 12. Pohon ruang status untuk langkah 9

Pada gambar 12, algoritma sudah menemukan solusi yang dianggap optimum, karena *cost* pada simpul 15 lebih besar dari simpul-simpul hidup lainnya. Maka, hasil dari algoritma *Branch and Bound* untuk pemilihan keempat permainan video pada contoh adalah untuk hanya membeli permainan video bertajuk “AI * Shoujo/AI * 少女” saja untuk memaksimalkan kesenangan dan meminimalkan uang yang dikeluarkan.

Apabila dilihat dengan seksama, seharusnya seluruh permainan diambil, karena tidak melebihi anggaran yang ada. Namun, algoritma yang didesain ternyata hanya mengambil permainan terakhir yang muncul pada list. Hal ini mungkin terjadi karena pendefinisian fungsi *cost* untuk tiap simpul yang kurang baik, sehingga algoritma “salah” memilih jalur dan tidak mengambil keuntungan yang bisa diambil seperti kemungkinan implementasi greedy.

IV. KESIMPULAN

Algoritma *Branch and Bound* dapat digunakan untuk membuat keputusan permainan apa yang harus dibeli diberikan anggaran tertentu dan harga-harga permainan yang diinginkan. Namun, agar jawaban yang diberikan memuaskan, maka fungsi *cost* yang ditentukan harus dapat memberikan pendekatan yang baik pula. Dalam makalah ini, penulis menggunakan fungsi *cost* yang merupakan adaptasi dari fungsi *cost* untuk *Integer Knapsack Problem*, dimana fungsi *cost* aslinya adalah $\hat{c}(i) = F + (K - W) \frac{p_{i+1}}{w_{i+1}}$, dimana F adalah harga yang sudah diraih di simpul sebelumnya, K adalah kapasitas *knapsack*, W adalah beban seluruh barang yang sudah diambil, p_i adalah harga barang ke- i , dan w_i adalah beban barang ke- i . Walaupun terlihat serupa, namun ternyata persentase “review” baik bukan ukuran yang baik untuk menyelesaikan masalah ini, sehingga perlu dicari lagi bagaimana fungsi *cost* yang seharusnya.

VIDEO LINK AT YOUTUBE

<https://youtu.be/Toseke8Rcl4>

UCAPAN TERIMA KASIH

Pertama dan terutama, penulis memanjatkan syukur kepada Tuhan Yang Maha Esa, sehingga makalah ini dapat selesai tepat waktu. Tak lupa juga penulis ucapkan terima kasih kepada Bapak Prof. Ir. Dwi Hendratmo Widyantoro, M.Sc., Ph.D yang telah membimbing selama satu semester dalam mata kuliah IF2211 Strategi Algoritma, sehingga penulis dapat memahami seluruh materi yang ada dengan baik. Penulis juga hendak mengucapkan terima kasih kepada teman dan keluarga yang senantiasa membantu dalam kegiatan perkuliahan selama satu semester ini, sehingga penulis dapat mengikuti perkuliahan dengan lancar hingga akhir semester.

REFERENSI

- [1] <https://steamdb.info/sales/history/>, diakses pada 10 Mei 2021.
- [2] <https://store.steampowered.com/>, diakses pada 10 Mei 2021.
- [3] Chaudhury, Abijit & Jean-Pierre KUILBOER (2002), *e-Business and e-Commerce Infrastructure*, McGraw-Hill
- [4] R. Munir, "Branch and Bound (Bagian 1)." Institut Teknologi Bandung, 2021.
- [5] J. Clausen, "Branch and Bound Algorithms – Principles and Examples." 1999

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, 11 Mei 2021



Fransiskus Febryan Suryawan