2nd Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)
21 Maret 2023 – Jakarta, Indonesia
Volume 2, Nomor 1, April 2023 - ISSN 2962-8628 (*online*)

PENERAPAN ALGORITME *GREEDY BEST FIRST SEARCH* (GBFS) TERHADAP SISTEM KASIR

Aaqila Dhiyaanisafa Goenawan[1], Abdurrahman Faqih[2], Mutiara Persada Pulungan[3*]

[1,2,3] Program Studi Ilmu Komputer, Ilmu Komputer, Sekolah Tinggi Ilmu Manajemen dan Ilmu Komputer ESQ, Jakarta, Indonesia

Email: [1]a.dhiyaanisafa.g@students.esqbs.ac.id, [2]a.faqih@students.esqbs.ac.id, [3]m.persada.p@students.esqbs.ac.id*

Abstrak- Aplikasi kasir adalah suatu aplikasi yang sering digunakan untuk melakukan proses bertransaksi pada sebagian besar pusat perbelanjaan. Saat ini masih banyak aplikasi kasir yang dilakukan secara manual, seperti menghitung jumlah uang yang harus dibayar dan uang yang harus dikembalikan sehingga membutuhkan waktu yang banyak agar tidak terjadi kesalahan saat melakukan proses transaksi. Sistem yang akan dibuat berguna untuk memudahkan kasir saat melakukan pekerjaannya dengan cepat dan mudah. Khususnya untuk transaksi pada sistem kasir yang dapat memberikan perubahan dari total belanja yang telah dilakukan sehingga mereka mencari cara untuk menentukan perubahan menjadi pecahan uang yang optimal. Metode yang digunakan untuk menentukan jumlah uang yang kira-kira akan diserahkan oleh pelanggan adalah algoritme Algoritme Greedy Best First Search. Diharapkan dengan diimplementasikannya algoritme Algoritme Greedy Best First Search pada aplikasi kasir dapat mempermudah dan mempercepat proses transaksi pembayaran pada kasir. Pada penelitian ini dibangun sistem untuk memberikan perubahan dari total pembelian yang dibeli. Perhitungan Algoritme Greedy ini akan dibantu dengan pendekatan metode Heuristic Analysis agar hasil yang dicapai lebih optimal.

Kata Kunci: Teknologi, Algoritme Greedy Best First Search, Sistem Kasir, Heuristic

IMPLEMENTATION OF THE GREEDY BEST FIRST SEARCH (GBFS) ALGORITHM TO THE CASHIER SYSTEM

Abstract- *The cashier application is an application that is often used to process transactions in most shopping centers. Currently there are still many cashier applications that are done manually, such as calculating the amount of money to be paid and money to be returned so that it takes a lot of time to avoid errors when processing transactions. The system that will be made is useful to make it easier for cashiers to do their work quickly and easily. Especially for transactions on the cashier system that can provide changes from the total spending that has been done so they are looking for ways to determine the change into optimal denominations of money. The method used to determine the amount of money that will be handed over by customers is the Greedy Best First Search Algorithm. It is hoped that the implementation of the Greedy Best First Search Algorithm in the cashier application can simplify and speed up the payment transaction process at the cashier. In this study a system was built to provide changes from the total purchases purchased. The Greedy Algorithm calculation will be assisted by the Heuristic Analysis method approach so that the results achieved are more optimal.*

Keywords: *Technology, Greedy Best First Search Algorithm, Cashier System, Heuristic*

1. PENDAHULUAN

Dalam kehidupan sehari-hari, banyak terdapat persoalan yang menuntut pencarian solusi optimum. Persoalan tersebut dinamakan persoalan optimasi (*Optimization Problems*). Persoalan optimasi adalah persoalan yang tidak hanya mencari sekedar solusi, tetapi mencari solusi terbaik. Solusi terbaik adalah solusi yang memiliki nilai minimum atau maksimum dari sekumpulan alternatif solusi yang mungkin. *Algoritme Greedy* adalah salah satu algoritme yang dapat digunakan untuk mendapatkan solusi terbaik dan merupakan algoritme yang paling populer saat ini. *Algoritme Greedy* merupakan metode yang paling populer dalam memecahkan persoalan optimasi. Hanya terdapat dua macam masalah optimasi, yaitu maksimasi serta minimasi[1]. Dalam bahasa Indonesia *greedy* berarti rakus, tamak, atau serakah. *Algoritme Greedy* menghasilkan solusi langkah per langkah sehingga pada tiap langkah harus dirancang keputusan terbaik untuk mengambil langkah selanjutnya. Keputusan yang sudah dirancang tidak bisa diubah pada langkah-langkah selanjutnya [2].

Algoritme Greedy dapat memilih jalur mana yang akan diambil terlebih dahulu atau bisa disebut dengan jalur optimum lokal sehingga sampai semua jalur diambil di akhir perjalanan dan membentuk rute perjalanan terpendek atau disebut dengan optimum global [3]. Penelitian ini membahas masalah sistem transaksi kembalian uang belanja pada sebuah toko ataupun minimarket, yaitu mencari jumlah minimum uang kembalian ketika berbelanja yang dihasilkan dari total belanjaan dengan uang yang dibayarkan oleh pembeli. Oleh karena itu, penelitian ini akan membuat suatu aplikasi yang dapat mencari solusi dari masalah tersebut, yaitu dengan menggunakkan *Algoritme Greedy Best First Search*.

Pembuatan sistem ini didasari pemikiran untuk memudahkan kasir dan menambah fungsi dari mesin kasir. Dengan sistem ini, mesin akan secara otomatis memberikan pecahan uang untuk kembalian uang yang dibutuhkan. Maka dari itu, fungsi dari sistem ini adalah untuk membantu mengurangi *human error* dalam menghitung kembalian uang pecahan dan menghindari kecurangan karyawan.
Adapun batasan-batasan masalahnya yang kami ambil yaitu:

a. Penentuan pecahan kembalian uang pada mesin kasir
b. Penentuan pecahan yang mendekati solusi optimum global
c. Total nilai pecahan yang dipilih harus sesuai jumlahnya dengan nilai uang yang harus dikembalikan.
d. Memeriksa apakah nilai total dari himpunan pecahan yang dipilih tidak melebihi jumlah uang yang harus dikembalikan.
e. Penyelesaian masalah menggunakan algoritme greedy best first search
f. Output yang dihasilkan merupakan jumlah pecahan uang yang dikeluarkan dengan nominalnya
g. User melakukan input total pembelian, total uang yang diterima, dan berapa kombinasi yang ingin dicoba untuk menghasilkan total uang yang harus dikembalikan
h. Nilai heuristiknya merupakan pecahan kembalian, semakin minimum pecahannya semakin baik.

2. METODE PENELITIAN

2.1 Tinjauan Pustaka

2.1.1 Algoritme

Algoritme merupakan suatu cara atau prosedur yang jelas untuk penyusunan langkah dalam menyelesaikan suatu masalah yang berbentuk kalimat dengan jumlah kata terbatas namun tersusun secara logis dan sistematis (cite). Algoritme bisa disebut dengan urutan logis langkah-langkah penyelesaian masalah yang disusun secara sistematis. Dapat disimpulkan definisi algoritme adalah ilmu yang membahas cara cara penyelesaian sebuah persoalan dengan deretan langkah tertentu kemudian tersusun dengan sistematis dan memakai bahasa logis serta memiliki sebuah tujuan [4].

2.1.2 Algoritme Greedy

Secara harfiah greedy berarti rakus atau tamak, hal ini sesuai dengan cara kerjanya yang mirip menggunakan salah satu sifat buruk manusia yaitu rakus. Algoritme Greedy adalah grup prosedur pemecahan yang selalu mengambil penyelesaian sementara/lokal yang terbaik pada setiap langkahnya untuk menyelesaikan suatu konflik. Pilihan terbaik akan diambil di setiap langkah tanpa perlu memikirkan bagaimana pengaruhnya terhadap penyelesaian secara keseluruhan. Algoritme greedy merupakan salah satu metode yang paling populer dalam menyelesaikan persoalan meningkatkan secara optimal (Optimization masalah). Persoalan optimasi ialah masalah yang menuntut pencarian solusi optimum (terbaik) [5].

Algoritme ini merupakan prosedur pemecahan yang sederhana serta fleksibel sehingga bisa digunakan di berbagai kasus persoalan menggunakan hasil yang relatif memuaskan. Algoritme Greedy dapat menentukan jalur mana yang akan diambil terlebih dahulu atau bisa disebut dengan jalur optimum lokal sehingga sampai seluruh jalur diambil di akhir perjalanan serta membentuk rute perjalanan terpendek atau disebut dengan optimum global. Algoritme greedy adalah algoritme yang memecahkan masalah langkah demi langkah, pada setiap langkah: Mengambil pilihan yang terbaik yang dapat diperoleh saat itu, berharap bahwa dengan memilih optimum lokal pada setiap langkah akan mencapai optimum global. Algoritme greedy mengasumsikan bahwa optimum lokal merupakan bagian dari optimum global.

Algoritme Greedy memiliki pendekatan untuk menciptakan solusi secara bertahap melalui urutan yang terus berkembang hingga solusi dari masalah sudah tercapai. Greedy memberikan cara lain optimal lokal dengan harapan setiap cara lain lokal membentuk alternatif global yang optimal secara keseluruhan. algoritme Greedy dapat menyelesaikan dengan menghitung nilai lokal optimal dan mendapatkan nilai optimasi global pada akhirnya [5].

2.1.3 Algoritme *Best First Search* (BFS)

Algoritme BFS menggunakan nilai heuristik di setiap simpul-simpulnya. Fungsi heuristik digunakan untuk mencari biaya perkiraan serta dikatakan terbaik jika nilai tersebut mendekati nilai sebenarnya. Greedy best –first search mencoba untuk memperluas node yang paling dekat dengan tujuan, menggunakan alasan bahwa ini adalah cenderung mengarah ke solusi cepat. Dengan demikian, mengevaluasi node hanya dengan menggunakan fungsi heuristik. Metode Best First Search merupakan sebuah bentuk metode dengan proses kerja membangkitkan simpul dari simpul yang sebelumnya. Metode BFS adalah proses menentukan simpul yang baru dengan

biaya yang paling terkecil dari seluruh simpul yang mempunyai nilai terdalam dan pernah juga simpul tersebut dibangkitkan [6].

Fungsi penilaian f(n) merupakan sebuah fungsi evaluasi best first search dengan penentuan simpul terbaik. Fungsi ini bisa diperkirakan menjadi simpul yg mempunyai simpul perkiraan untuk mencapai nilai akhir, atau dianggap juga sebagai penggabungan biaya perkiraan atau biaya yang sebenarnya.pada proses pencarian terbaik pertama,kita wajib menyeleksi node node yang memakai fungsi heuristik yang telah memadai di setiap node atau simpul dengan menggunakan aturan aturan untuk memperoleh hasil pengganti. Pada Fungsi Heuristik bagian dari suatu strategi dalam melakukan proses pencarian ruang secara selektif, menggunakan konsep kemungkinan benar paling besar [7]. Beberapa terminologi dalam algoritme best first search adalah sebagai berikut.

a. Fungsi evaluasi f(n), adalah fungsi yang digunakan untuk membangkitkan simpul dari simpul sebelumnya.
b. Nilai heuristik h(n), adalah nilai perkiraan yang menjadi dasar dipilihnya simpul saat ini menjadi simpul terbaik.
c. Nilai sebenarnya g(n), adalah nilai "jarak" antara simpul akar dengan simpul saat ini.
d. Simpul awal disebut juga simpul akar, adalah simpul pertama dalam pohon yang akan dibentuk.
e. Simpul sekarang adalah simpul yang sedang dievaluasi dengan fungsi evaluasi untuk ditentukan apakah layak atau tidak menjadi solusi.
f. Kandidat (suksesor) adalah simpul selanjutnya yang hendak diperiksa.
g. Open list, adalah daftar simpul yang mungkin diakses dari simpul awal atau simpul yang sedang dijalankan.
h. Close list, adalah daftar simpul yang saat ini menjadi solusi sementara, yaitu solusi terbaik saat ini.
i. Simpul tujuan, adalah simpul yang hendak menjadi tujuan akhir.

2.1.4 Algoritme *Greedy Best First Search*

Algorithm Greedy Best-First search adalah jenis algoritme yang menggunakan pendekatan penyelesaian masalah dengan mencari nilai maksimum sementara di setiap langkahnya. Tujuan dari algoritme greedy adalah memilih rute terpendek berdasarkan biaya perkiraan. *Greedy best –first search* mencoba untuk memperluas node yang paling dekat dengan tujuan, dengan alasan bahwa ini adalah cenderung mengarah ke solusi cepat[8]. Dengan demikian, mengevaluasi node dengan hanya menggunakan fungsi heuristik pada persamaan (1).

$$\mathbf{F(n) = h(n)} \quad (1)$$

Dimana:
f(n) = Fungsi Evaluasi
h(n) = Fungsi Heuristik
Adapun mekanisme algoritme Greedy Best-First Search adalah [9]:

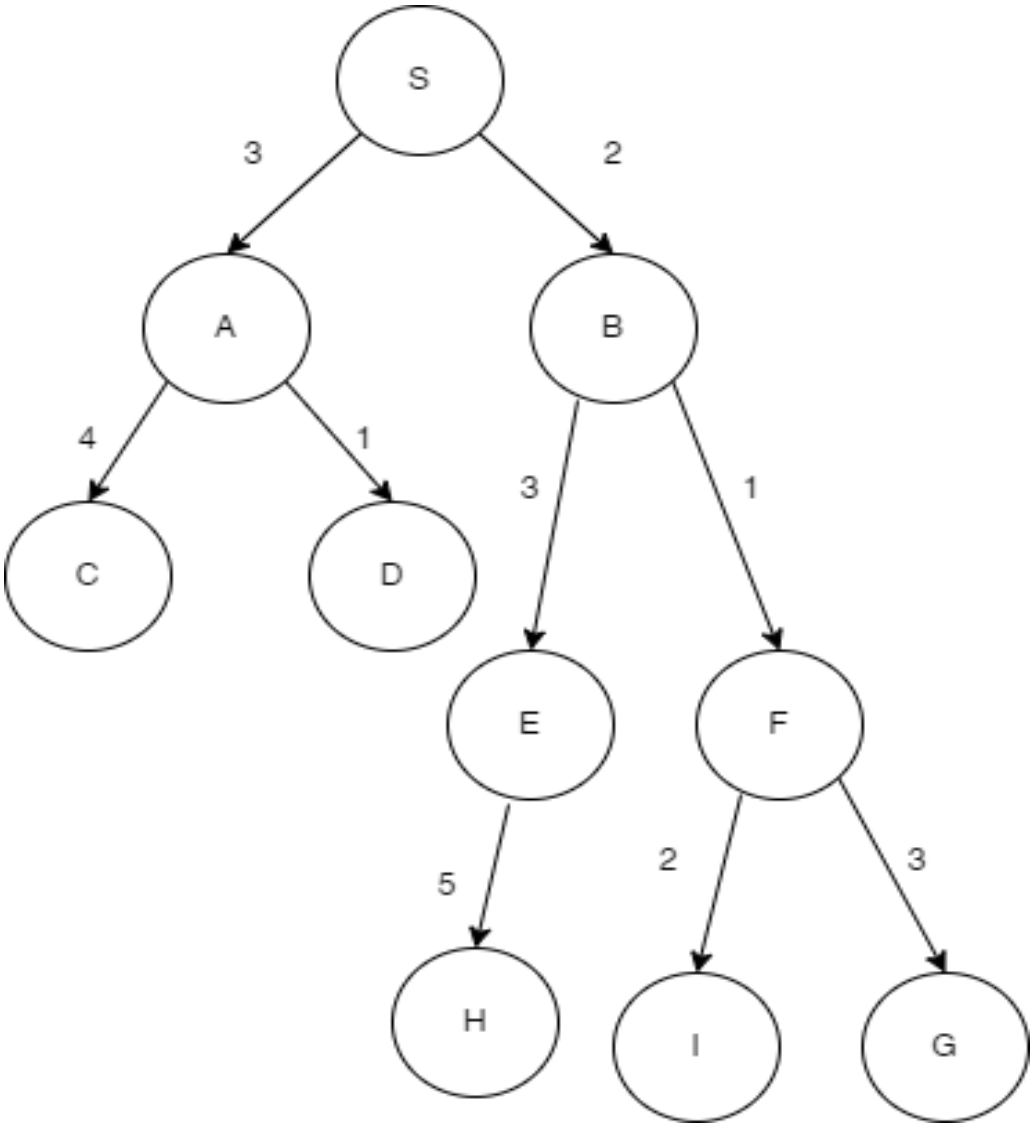
1. Tempatkan Node awal di antrian open.
2. Kerjakan langkah-langkah berikut sampai tujuan ditemukan atau sampai antrian open telah kosong:
3. Ambil node terbaik dari open.
4. Bangkitkan seluruh successornya.
5. Untuk tiap-tiap successornya kerjakan.
6. Jika node tersebut belum pernah dibangkitkan, evaluasi node tadi dan masukkan ke open.

Jika node tersebut sudah pernah dibangkitkan sebelumnya, ubah parent Jika lintasan baru lebih menjanjikan. Hapus node tersebut dari antrian open. Contoh penerapan algoritme Greedy Best-First Search untuk pencarian pada pencarian pada Gambar 1.
Pada setiap iterasi, setiap node diperluas menggunakan fungsi evaluasi f(n)=h(n) , yang diberikan pada Tabel 1.

Tabel 1. Data pencarian

Node	H(n)
A	12
B	4
C	7
D	3
E	8
F	2
H	4
I	9
S	13
G	0

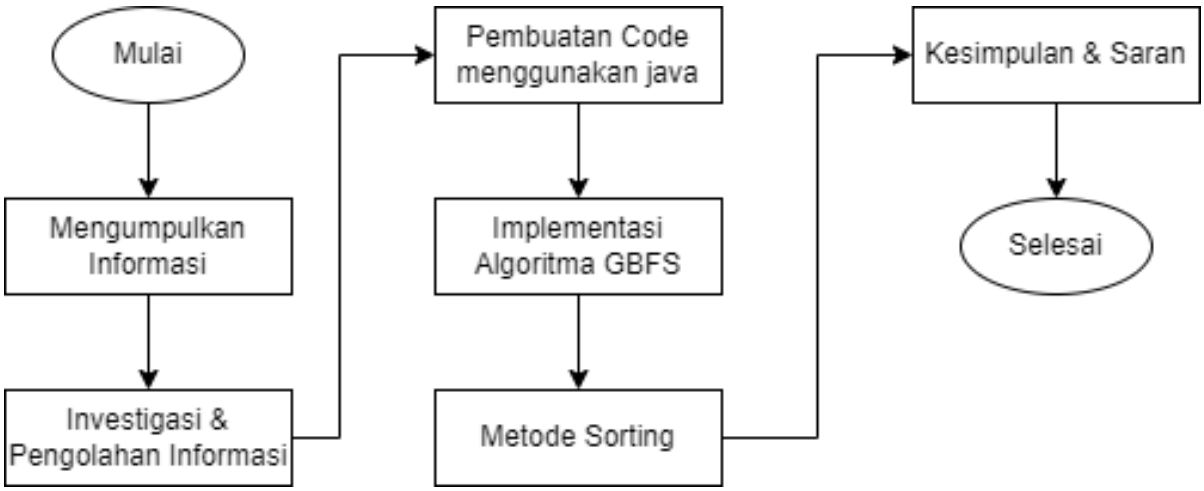
Dalam contoh pencarian ini, digunakan dua list yaitu OPEN dan CLOSED. Berikut ini adalah iterasi untuk melintasi contoh di atas.
Expand nodes S dan letakkan di list CLOSED
Initialisasi: Open [A, B], Closed [S]
Iterasi 1: Open [A], Closed [S, B]
Iterasi 2: Open [E, F, A], Closed [S, B]
Open [E, A], Closed [S, B, F]
Iterasi 3: Open [I, G, E, A], Closed [S, B, F]
Open [I, E, A], Closed [S, B, F, G]
Sehingga solusi final adalah S----> B----->F----> G



Gambar 1. Masalah Pencarian

2.2 Metodologi Penelitian

Metode penelitian dilakukan untuk mengetahui langkah yang dimiliki seperti mengumpulkan informasi, investigasi dan pengolahan pada informasi yang didapat, pembuatan code, pengimplementasian algoritme, hingga memberikan kesimpulan dari segala informasi yang kami dapat.



Gambar 2. Alur Penelitian

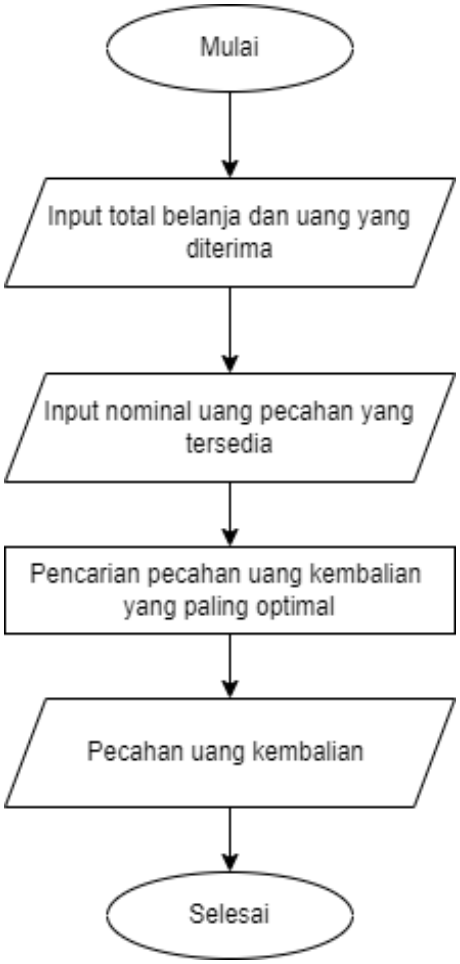
Adapun tahapan penelitian pada gambar 2 adalah sebagai berikut:

1. Langkah pertama yang dilakukan dalam penelitian ini adalah mengumpulkan informasi yang dibutuhkan untuk membangun algoritme. Informasi yang dibutuhkan antara lain cara kerja algoritme, pembuatan node, cara pembangkitan successor, dsb. Selain itu juga digali masalah utama penelitian yang melatarbelakangi pemilihan algoritme.
2. Setelah informasi dikumpulkan kemudian dilakukan investigasi dan pengolahan informasi. Pada tahap ini dibuat juga skenario untuk melakukan pengujian untuk melakukan evaluasi terhadap algoritme.
3. Tahap berikutnya adalah melakukan tahap pembuatan code menggunakan bahasa pemrograman Java. Bahasa pemrograman Java dipilih karena cukup sederhana dan mendukung struktur data yang diperlukan untuk pengolahan data
4. Dalam tahap pembuatan code, juga dilakukan implementasi algoritme GBFS sesuai dengan alur teoritisnya. Algoritme GBFS diterapkan untuk memilih jalur terpendek dalam menghasilkan beberapa pecahan uang yang totalnya memenuhi total uang kembalian yang dihitung dari total uang yang diterima dikurangi dengan total belanjanya
5. Setelah implementasi algoritme GBFS, tahap selanjutnya dilakukan sorting terhadap hasil yang diperoleh
6. Tahap akhir adalah penarikan kesimpulan dan saran terhadap hasil implementasi algoritme yang telah diujicoba.

Dalam penelitian ini, kami menerapkan sebuah algoritme yang disebut *Algoritme Greedy Best-First Search*, atau yang dikenal dengan *Algoritme Greedy*. Algoritme sendiri merupakan kumpulan perintah untuk menyelesaikan suatu masalah dimana masalah itu diselesaikan secara sistematis, terstruktur dan logis. Dalam sistem ini, kami menggunakan *Algoritme Greedy* yang pengambilan keputusan terbaiknya pada saat terjadi masalah adalah yang paling pertama ditemukan dan diharapkan keputusan tersebut menjadi solusi terbaik. Nilai fungsi evaluasi pada *Greedy Best First Search* bergantung pada nilai fungsi heuristik h(n) itu sendiri. Fungsi heuristik h(n) akan memberikan estimasi arah yang benar, sehingga pencarian jalur terpendek dapat sangat cepat. Metode ini cukup populer untuk memecahkan masalah optimal. *Algoritme Greedy* menggunakan fungsi evaluasi dengan meniadakan perkiraan biaya g(n), dimana f(n) = h(n) [4].

Langkah yang kami lakukan adalah membuat code program menggunakan bahasa Java untuk menjalankan algoritme Greedynya yang digunakan untuk menghasilkan beberapa pecahan uang yang totalnya memenuhi total uang kembalian yang dihitung dari total uang yang diterima dikurangi dengan total belanjanya. Metode Greedy dibuat untuk menemukan hasil mendekati optimal, yaitu hasil yang pertama kali ditemukan. Setelah itu, kami membuat metode sorting untuk mendukung metode yang mencari solusi Global, atau metode untuk mencari solusi maksimasi dan minimasinya.

2.3 *Flowchart*



Gambar 3. Flowchart

Gambar 3 menunjukkan alur kerja sistem yang dibangun. Tahap awal pada flowchart adalah menginputkan total uang belanja dan total uang yang diterima oleh kasir. Selanjutnya sistem akan otomatis menghitung kembalian uang yang harus diberikan. Setelah itu, algoritme greedy best first search akan menentukan uang pecahan uang kembalian dengan pilihan yang paling optimal (sedikit). Tahapan akhir dari sistem yang akan dibangun adalah dengan menghitung berapa jumlah pecahan uang yang paling sedikit(minimasi) dan paling banyak (maksimasi).

3. HASIL DAN PEMBAHASAN

Sistem yang dirancang merupakan sistem kasir yang penerapannya menggunakan algoritme Greedy Best First Search untuk memilih pecahan uang kembalian dari uang yang diterima. Sistem ini akan memilih kemungkinan pecahan uang kembalian yang sesuai dengan jumlah yang dibutuhkan. Pada penelitian ini, user perlu untuk memasukkan total uang belanjaan, uang yang diterima, nominal pecahan yang tersedia pada kasir, serta jumlah kombinasi penjumlahan uang pecahan untuk mencapai kembalian. Selanjutnya, sistem akan menghitung nominal uang yang harus dikembalikan dan memberikan jumlah nominal pecahan yang mencapai total kembalian yang tepat.

Uji coba dilakukan dengan menggunakan metode *White Box Testing*. *White Box Testing* adalah salah satu metode pengujian program yang bertujuan untuk memeriksa komponen program apakah berjalan semestinya dengan melihat internal kode program tersebut [10]. Pada penelitian ini, kami melakukan uji coba dengan contoh kasus dimana seseorang ingin membayarkan belanjaannya sebanyak Rp40.000, dengan memberikan uang sebesar Rp100.000. Uang pecahan yang tersedia pada sistem kasir saat ini adalah Rp5.000, Rp10.000 dan Rp20.000.

Dengan contoh kasus itu, sistem kasir yang diteliti ini dapat memberikan solusi optimal untuk pecahan uang kembalian yang dibutuhkan untuk memenuhi total kembaliannya.

a. Masukan total belanja (Rp) : 40000
b. Masukan uang yang diterima (Rp) : 100000
c. Masukkan jumlah kemungkinan pecahan uang kembalian untuk uang Rp60000
d. Pecahan ke 1 yaitu 10000 + 20000 + 10000 +50000 = 90000 tidak memenuhi 60000
e. Pecahan ke 2 yaitu 20000 + 20000 + 10000 + 10000 = 60000, dengan 4 Pecahan uang
f. Pecahan ke 3 yaitu 20000 + 10000 + 20000 + 10000 = 60000, dengan 4 Pecahan uang
g. Pecahan ke 4 yaitu 10000 + 50000, = 60000, dengan 2 Pecahan uang
h. Pecahan ke 5 yaitu 20000 + 20000 + 50000 = 90000 tidak memenuhi 60000
i. Paling Sedikit 2 Pecahan (minimasi)
j. Paling Banyak adalah 4 Pecahan (maksimasi)

Dengan Algoritme Greedy, karena yang paling pertama ditemukan adalah pecahan ke 2 yaitu 2 lembar Rp20000 dan 2 lembar Rp10000 maka itulah solusi terbaik yang ditemukan. Sedangkan untuk solusi paling optimalnya adalah pecahan ke-4 yaitu pecahan Rp10000 dan pecahan Rp50000. Pada Algoritme Greedy kita juga harus memecahkan persoalan optimasinya, yaitu mencari maksimasi dan minimasi dari hasil-hasil yang ditemukan. Untuk contoh kasus ini, nilai maksimasi yang ditemukan adalah 4 pecahan uang, yang juga merupakan solusi terbaik yang ditemukan dengan Algoritme Greedy.

Untuk hasil minimasi yang ditemukan di contoh kasus ini adalah 2 pecahan uang, dimana 2 pecahan uang merupakan solusi optimal dari kasus ini. Namun, pada Algoritme Greedy sendiri, solusi yang ditemukan tidak selalu hasil yang optimal, tetapi yang mendekati optimal. Sehingga, hasil minimasi yang ditemukan pada contoh kasus ini bukan merupakan solusi yang dihasilkan oleh Algoritme Greedy.

```
Masukan total belanja (Rp) :40000
Masukan uang yang diterima (Rp) :100000
Nominal uang yang harus dikembalikan adalah sebesar Rp60000
Nominal uang yang tersedia :
```

Gambar 4. Tampilan Input Total Belanja dan Uang Yang Diterima

Gambar 4 merupakan tampilan uji coba program dengan memasukkan total belanja dan uang yang diterima. Sistem akan secara otomatis menghitung berapa nominal uang yang harus dikembalikan. Setelah itu kasir akan memasukkan nominal uang yang tersedia pada mesin kasir tersebut.

Gambar 5 Uji coba program dengan memasukkan jumlah nominal uang yang tersedia pada mesin kasir tersebut. Serta jumlah pecahan uang yang tersedia. Program secara otomatis menghitung jumlah uang yang harus dikembalikan.

```
Jumlah uang yang tersedia :3
Uang 1 dengan nominal (Rp) : 10000
Uang 2 dengan nominal (Rp) : 20000
Uang 3 dengan nominal (Rp) : 50000
Masukkan jumlah kemungkinan pecahan uang kembalian untuk uang Rp60000  :5
Pecahan ke 1 yaitu 10000, 20000, 10000,  = 90000 tidak memenuhi 60000
Pecahan ke 2 yaitu 20000, 20000, 10000, 10000,  = 60000, dengan 4 Pecahan uang
Pecahan ke 3 yaitu 20000, 10000, 20000, 10000,  = 60000, dengan 4 Pecahan uang
Pecahan ke 4 yaitu 10000, 50000,  = 60000, dengan 2 Pecahan uang
Pecahan ke 5 yaitu 20000, 20000,  = 90000 tidak memenuhi 60000
Paling Sedikit 2 Pecahan
Paling Banyak adalah 4 Pecahan
BUILD SUCCESSFUL (total time: 40 seconds)
```

Gambar 5. Tampilan Input Nominal Uang Yang Tersedia

4. KESIMPULAN

Berdasarkan penelitian yang dilakukan Penerapan Algoritme Greedy Untuk Penukaran Uang Rupiah, maka dapat ditarik kesimpulan bahwa Perancangan Aplikasi Penerapan Algoritme Greedy Untuk Kembalian Uang Rupiah dapat mempermudah melakukan proses pencarian solusi optimal untuk menentukan pecahan yang paling kecil. Penerapan Algoritme Greedy Untuk kembalian uang rupiah ini dapat dipakai di toko atau supermarket dimana fungsinya untuk mempermudah kasir dalam memberikan pecahan kembalian uang.

UCAPAN TERIMA KASIH

Selama menyelesaikan program ini para penulis banyak dibantu oleh dosen pendamping Ibu Ahlijati Nur Aminah dan Bapak Edo Surya Utama.

DAFTAR PUSTAKA

[1] E. S. Laela, W. Gata, and J. J. Purnama, "Optimalisasi Algoritme Greedy dalam Penyusunan Jadwal Pelajaran pada SMK Nurul Islam Cianjur," vol. 12, no. 7, 2022, doi: 10.36418/syntax.

[2] M. Fikri Naufal, "Penerapan Algoritme Greedy dalam Menghitung Bandwith pada Jaringan Telepon," 2021. [Online]. Available: https://www.pengertianilmu.com/

[3] Y. Darnita and R. Toyib, "Penerapan Algoritme Greedy Dalam Pencarian Jalur Terpendek… Penerapan Algoritme Greedy Dalam Pencarian Jalur Terpendek Pada Instansi-Instasi Penting Di Kota Argamakmur Kabupaten Bengkulu Utara," 2019.

[4] P. B. Sitepu and A. P. Harianja, "Implementasi Penukaran Uang Rupiah Dengan Menggunakan Algortima Greedy," 2018.

[5] Y. Afero *et al.*, "Algoritme Best First Search Menentukan Lintasan Jalur Terpendek Pada Kota Wisata Bukittinggi," *JOISIE Journal Of Information System And Informatics Engineering*, vol. 5, no. Desember, pp. 138–145, 2021.

[6] N. Sugianti, A. Mardhiyah, and N. Romihim Fadlilah, "Komparasi Kinerja Algoritme BFS, Dijkstra, Greedy BFS, dan A* dalam Melakukan Pathfinding," 2020.

[7] A. Yulianto, P. Satya Darma, and D. D. Praseno, "Aplikasi Android Untuk Deteksi Penyakit Kucing Dengan Metode Forward Chaining," 2023.

[8] G. Omonkhodion, "A Comparative Study of A* and Greedy Best-First Search Algorithms in Solving 8-Puzzle Game ☆ A Comparative Study of A* and Greedy Best-First Search Algorithms in Solving 8-Puzzle Game," 2023.

[9] A. D. Sabilla and A. Taufiq, "Journal of Information System and Computer Penerapan Algoritme A* Pada Webgis Pencarian Rute Terpendek," 2022. [Online]. Available: https://journal.unisnu.ac.id/JISTER/

[10] dan Joko Susilo, "Aplikasi Pengujian White-Box Ibii Online Judge," 2019. [Online]. Available: http://www.kwikkiangie.ac.id