Visualization Software Design Greedy's Algorithm

Wasit Ginting[1], Alex Rikki[2], Hermansyah Sembiring[3]
[1,2]Faculty of Computer Science, Universitas Katolik Santo Thomas, Medan, Indonesia,
[3]Informatics Management Study Program, STMIK Kaputama, Binjai, Indonesia.

Article Info

Article history:

Received, Jul 31, 2021
Revised, Oct 18, 2021
Accepted, Nov 11, 2021

Keywords:

Design,
visualization,
algorithm,
greedy.

ABSTRACT

Greedy Algorithm Visualization Software Design is a depiction or planning of software with engineering in making images, diagrams or animations to display information in the form of a logical and systematic arrangement to solve a problem by finding the optimum solution. The goal is to create a visualization software design program that can optimize the preparation of goods in containers and the steps to solve them so as to facilitate problem solving, especially in real life. Greedy algorithm solves the problem step by step or step by step, which at each step will take the best option that can be obtained at that time or get the solution quickly on the same day. Greedy Algorithm will be used to find the optimum solution from the arrangement of goods in containers that will get optimal results, especially to reduce empty space. Items are arranged according to height, width, and length. By making visualizations developed using software is an effective way that helps users to better understand and can do the learning process independently.

Corresponding Author:

Wasit Ginting
Faculty of Computer Science, Universitas Katolik Santo Thomas, Medan, Indonesia,
Jl. Setia Budi No.479-F Medan Sumatera Utara.
Email: wasitginting74@gmail.com

1. INTRODUCTION

The arrangement of goods must be considered carefully so that the placement of goods is arranged optimally with a minimum number of shipping loads. Many things must be considered in the preparation of goods, because field officers will usually try to arrange the location and position of the goods so that they are optimal, so that it takes a long time, drains a lot of energy and cost savings are also difficult to achieve and will even cost more.

Arrangement of goods in containers is a problem because it often occurs that the arrangement of goods is not optimal and there is still empty space left that should be able to be placed with more goods so that the cost is doubled. The rest of the goods left behind can be transported in one container only if the arrangement of the layout of the goods in it is optimized.

In the preparation of an item, the length, width, and height of the item must be considered, which are separate thoughts in the process of placing goods in a container. Because the length, width, and height of an item indicate at the level of the layout of the cargo into the container. To overcome these problems, a solution can be found, one of which is to use the Greedy algorithm. This algorithm is a popular method for solving an optimization problem. Optimization problem which means finding the optimum solution. Greedy's algorithm is an algorithm that solves the problem step by step, at each step taking the best option that can be obtained at that time without regard to future consequences (the principle of "take what you can get now!"), hoping that by choosing a local optimum at each the step will end with a global optimum" (UH and Wijanarto, 2011:1).

2. RESEARCH METHOD

As for how the greedy algorithm works, it will be discussed based on examples that have been studied previously, namely:

1. Greedy in knapsack problem

In solving the Knapsack Problem there are 3 types of Greedy algorithms that can be used, namely:

A. Greedy By Weight

At each step select the object that has the lightest weight. Try to maximize profit by inserting as many objects into the knapsack as possible. The first thing to do is to sort the program in ascending order of objects based on their weight. Then the objects that can be accommodated by the knapsack are taken one by one until the knapsack is full or there are no more objects that can be inserted.

B. Greedy by profit

At each step, select the object that has the greatest advantage. Try to maximize profit by selecting the most beneficial object first. The first thing to do is to sort the program in descending order of objects based on their profit. Then the objects that can be accommodated by the knapsack are taken one by one until the knapsack is full or there are no more objects that can be inserted. C. Greedy By Density

At each step the knapsack is filled with the object that has the largest pi/wi, where p is the profit and w is the weight of the item. Trying to maximize profit by selecting the object that has the greatest density per unit weight. The first thing that is done is the program looks for the value of profit per weight of each unit (density) of each object. Then the objects are sorted according to their density. Then the objects that can be accommodated by the knapsack are taken one by one until the knapsack is full or there are no more objects that can be inserted (Rachwati and Candra, 2013: 188).

2). Greedy Algorithm on Vending Machines

This vending machine program will apply the Greedy algorithm to determine what fractions will appear by minimizing the appearance of coins during the refund process. The vending machine will work to remove the goods if you insert paper money or coins first. The vending machine program is designed in such a way that can give change in the form of coins. By applying the Greedy algorithm to the vending machine program, the amount of change received by consumers will be minimized (Alamsyah and Putri, 2014: 202).

3). Shortest route search using Greedy Algorithm

In determining the most optimal route, the steps taken are to describe the routes (trajectories) traversed with a network diagram using graph theory and then look for the most optimal route (trajectory) using a greedy algorithm. The greedy algorithm to find the shortest path can be formulated as follows:

a. Examine all edges directly adjacent to the vertex . Choose the side with the smallest weight. This side becomes the first shortest path, call it L(1).
b. Determine the second shortest path in the following way:
   i.calculate: d(i) = length L(1) + side weight from end vertex L(1) to other vertex i
   ii. choose the smallest d(i)

Compare d(i) with the side weights (a, i).

If the side weight (a, i) is less than d(i), then L(2) = L(1) U (side from end vertex L(i) to vertex i)

c. In the same way, repeat step 2 to determine the next shortest path (Hayati and Yohanes, 2014: 394).

General Schematic of Greedy Algorithm

According to (Rachmawati and Candra, 2013: 188). Algorithm general scheme greedy is as follows:

a. Initialize S with blank.
b. Select a candidate C with the selection function.
c. Subtract C from the candidate selected from step (b) above.

d. Check whether the selected candidate is together with the set solutions form a feasible solution (with a feasibility function).
e. Check if the solution set has provided a complete solution and optimal (with objective function).
f. The general scheme of the Greedy algorithm can be seen in Figure 1 (Alamsyah and Putri, 2014: 204).

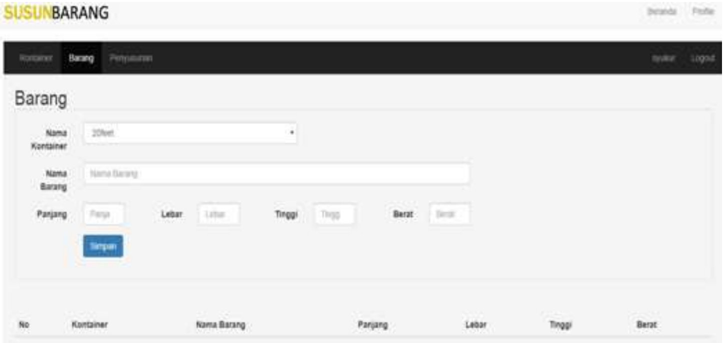
3. RESULTS AND DISCUSSION

The Greedy algorithm works in the preparation of goods into containers, namely:

1. Sort the item data based on the length, height, and width of the item.
2. Put items that have length, height, and width first into the container by choosing items that are longer than the other items.
3. Check the length, width, and height of the item to be loaded. If the goods that have been entered into the container are < the goods to be included, then enter the goods above the previous goods.
4. Then calculate the remaining height of the container against the goods in the container.
5. If the remaining height of the container is < the item to be loaded, then raise the item to the top of the item in the previous container.
6. If the remaining height of the container > the items to be loaded then check the remaining width of the container against the items in the container.
7. If the remaining width of the container for existing goods > the width of the goods to be included, the goods are placed next to the goods already in the container.
8. If the remaining width of the container for the existing goods is < the goods to be included, then calculate the remaining length of the container for the existing goods.
9. If the remaining length of the container for existing goods is < than the length of the goods to be included, the goods will not be included.
10. If the remaining length of the container for existing goods is > from the length of the goods to be inserted, then check the remaining width and height of the container against the height and width of the goods to be inserted.
    a) If it is smaller then enter the item in front of the existing item.
    b) If it is larger then the item is not included.

a. In this application there are 5 items, namely:
   1. Item A length = 50, width = 40, height = 30.
   2. Item B length = 60, width = 50, height = 40.
   3. Item C length = 40, width = 20, height = 30.
   4. Item D length = 70, width = 50, height = 20.
   5. Item E length = 30, width = 20, height = 50.
b. The size of the container length = 575, width = 235, height = 238.
c. Sort the items first from the largest length, width, and height.
d. Compare the size of the container with items that have a larger length first. Item D is longer than item D. Then item D is compared to the size of the container.
e. If the length, width, and height of the goods are < the length, width, and height of the container, then item D is included in the container.
f. Next, calculate the remaining length, width, and height of the empty container.
g. tem B is compared with the remaining length, width, and height of the container. If item B<large length, width, and height of the container, then item B goes into the container.
h. Then return to part f to g.
i. If the length, width, and height of the goods are > the length, width, and height of the container, the goods will not be included in the container.
j. For the layout of goods D, B, A, C, E.

The item input page display is used by members to enter, change, and delete item data. Goods can be filled if the container is pre-made. The display of the goods data input page can be seen in Figure 1.



Barang

Nama Kontainer: 20feet

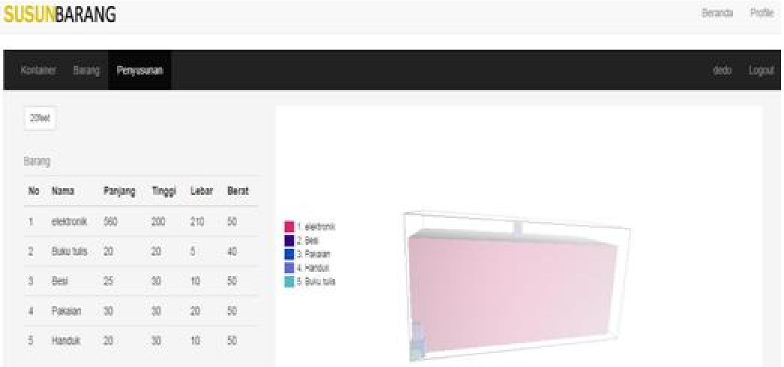
Nama Barang: Nama Barang

Panjang: Panja Lebar: Lebar Tinggi: Tinggi Berat: Berat

No	Kontainer	Nama Barang	Panjang	Lebar	Tinggi	Berat

Figure 1. Display of the Item Data Input Page

The input page display for the preparation process is used by members to process the preparation of goods into containers. The drafting process can proceed if the containers and goods have been pre-filled. The page display of the preparation process can be seen as shown in Figure 2.

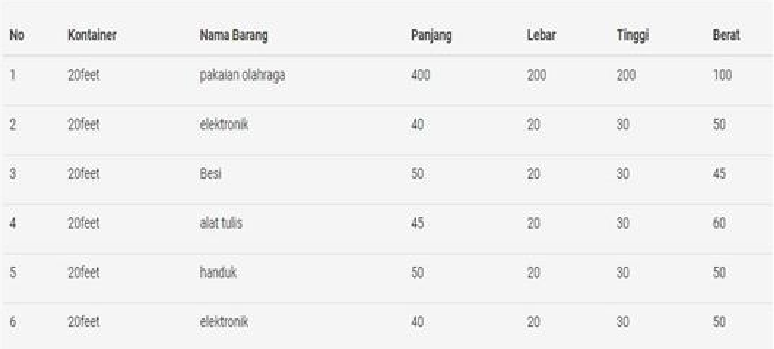


Barang

No	Nama	Panjang	Tinggi	Lebar	Berat
1	elektronik	560	200	210	50
2	Buku tulis	20	20	5	40
3	Besi	25	30	10	50
4	Pakaian	30	30	20	50
5	Handuk	20	30	10	50

Figure 2. Display of the Compilation Process

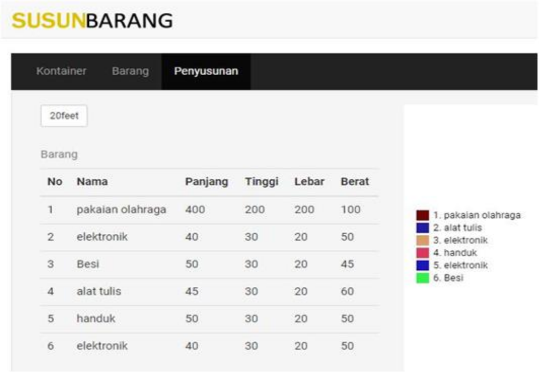
The list of items will display information on all member items listed in the database. The results of the list of goods can be seen as shown in Figure 3.



No	Kontainer	Nama Barang	Panjang	Lebar	Tinggi	Berat
1	20feet	pakaian olahraga	400	200	200	100
2	20feet	elektronik	40	20	30	50
3	20feet	Besi	50	20	30	45
4	20feet	alat tulis	45	20	30	60
5	20feet	handuk	50	20	30	50
6	20feet	elektronik	40	20	30	50

Figure 3. Display of Item List Results

The list of goods arrangement will display information on the list of goods in the container. The results of the list of items can be seen as shown in Figure 4.



Barang

No	Nama	Panjang	Tinggi	Lebar	Berat
1	pakaian olahraga	400	200	200	100
2	elektronik	40	30	20	50
3	Besi	50	30	20	45
4	alat tulis	45	30	20	60
5	handuk	50	30	20	50
6	elektronik	40	30	20	50

Figure 4. Display of the results of the List of Items

4. CONCLUSION

Based on the results of data collection that has been obtained or obtained, several conclusions are obtained from the Greedy Algorithm Visualization Software Design, especially in the preparation of goods into containers that will be built to meet user needs, namely The system is able to store data on goods that enter the container, The system can display per-container goods, Can be viewed online for those who have access rights, The system provides facilities for making reports on goods, The greedy algorithm is one of the algorithms that can solve problems and provide solutions to the process of arranging goods into containers where the optimization of goods is sorted by width, length, and height.

REFERENCES

[1] Alamsyah, dan Putri, I. T. 2014. Penerapan Algoritma Greedy pada Mesin Penjual Otomatis (Vending Machine). Scientific Journal of Informatics. Vol. 1 No. 2:201-209.

[2] Afyenni, R. 2014. Perancangan Data Flow Diagram untuk Sistem Informasi Sekolah (Studi Kasus pada SMA Pembangunan Laboratorium UNP). Jurnal TEKNOIF. Vol. 2. No. 1: 35-39.

[3] Cahyaningtyas, R. dan Iriani, S. 2015. Perancangan Sistem Informasi Perpustakaan pada Smp Negeri 3 Tulakan, Kecamatan Tulakan Kabupaten Pacitan. Indonesian Journal on Networking and Security. Vol. 4. No. 2: 15-20

[4] Februariyanti, H. dan Zuliarso, E. 2012. Rancang Bangun Sistem Perpustakaan untuk Jurnal Elektronik. Jurnal Teknologi Informasi Dinamik. Vol. 17. No. 2: 124-132.

[5] Ghozali, A. E. Setiawan, B. D. dan Furqon, M. T. 2017. Aplikasi Perencanaan Wisata di Malang Raya dengan Algoritma Greedy. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer. Vol. 1. No. 12: 1459-1467.

[6] Harison, dan Syarif, A. 2016. Sistem Informasi Geografis Sarana pada Kabupaten Pasaman Barat. Jurnal Teknoif. Vol. 4. No. 2: 40-50.

[7] Hayadi, B. H. 2017. Visualisasi Konsep Umum Sistem Pakar Berbasis Multimedia. Riau Journal of Computer Science. Vol. 3. No. 1: 17-22.

[8] Hayati, E. N. dan Yohanes, A. 2014. Pencarian Rute Terpendek menggunakan Algoritma Greedy. Semarang: Universitas Stikubank.