Jurnal JURTIE, Vol 3 No 1 Januari 2021 hal 26-31
e-ISSN: 2809-7742; p-ISSN : 2809-770X
Received 14 November 2020/ Revised 3 Desember 2020 / Accepted 24 Januari 2021

Jurnal Teknik Informatika dan Elektro
https://jurnal.ugp.ac.id/index.php/jutei

Searching The Shortest Route Of Delivery Using The A* Algorithm Case Study Jne Office In Jakarta Selatan

Abdurrahman Faqih[1*], Khonsa Mutmainnah[2], Muhammad Bakhara Alief R[3]

[1,2,3]*Program Studi Ilmu Komputer STIMIK ESQ Jakarta, Kampus Menara 165, Lantai 18 & 19, Jl. Tb. Simatupang Kav. 1, Jakarta, Indonesia*

Abstract. Couriers deliver goods to many places every day. Therefore, the couriers need to find the shortest path between their intended locations. This is to save time, energy, and costs. The A* algorithm is one of the methods commonly used in finding the shortest path. By using this method, it is hoped that couriers will find it easier to find the shortest path when sending goods

Keywords: A* Algorithm; Delivery courier; JNE; Path search

1. Introduction

In the current era of globalization, the need for shipping services has increased quite rapidly. Delivery services are part of consumer needs. Consumers who use this freight forwarding service tend to want something fast, easy, safe, and practical in terms of shipping goods. The domestic shipping service business in recent years has grown quite rapidly, this can be seen from its growth which reached 14.7% in 2014 and has increased by more than Rp. 100 trillion per year in the last 3 years. One of the companies engaged in shipping services is JNE, which is a freight forwarding and logistics company that serves delivery in the form of packages, documents, etc.

In the process of delivering goods, JNE couriers need to find the fastest route so that the goods sent arrive quickly to consumers, also saving costs and time spent. In the search for the shortest path, the A* (A-Star) algorithm is used. The A* (A-Star) algorithm is one of the guided search technique algorithms (heuristic techniques). This algorithm is studied to solve problems that use graphs or images to expand the state space. In other words, used to solve problems that can be represented by graphs, there are many algorithms used in finding the shortest path, one of which is A*

2. Methods

2.1 Implementation of A* (A-*Star*)

In this case, we used the JNE freight forwarding office in South Jakarta as the object of research to determine the shortest distance traveled by couriers to other JNE offices using the A* algorithm. In this case, we implement it in a java programming language that can output the shortest mileage according to the starting point and destination point entered by the user.

*Corresponding author's email: a.faqih@students.esqbs.ac.id

The following is a map of the JNE office located in the South Jakarta area which was taken and used as an object of research by the author based on search results using Google Maps:

Figure 1. Maps of JNE Office
Source:https://www.google.com/maps/search/kantor+jne+jakarta+selatan/@6.2670789,106.7384905,12z/data=!3m1!4b1, 2021

From these results, then we visualize it in the form of a graph accompanied by the paths at each JNE office in the South Jakarta area. The following is a visualization image using a graph of the JNE office in the South Jakarta area:

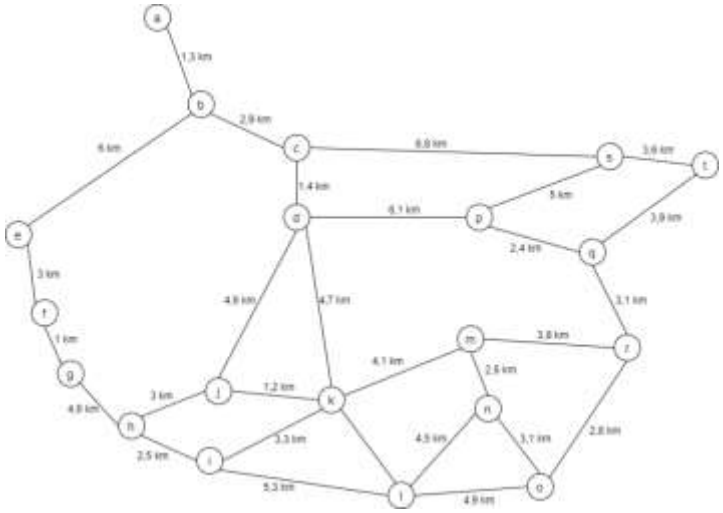
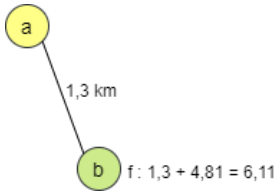


Figure 2. Office of JNE in South Jakarta Area.

After searching the distance between JNE offices, we try to calculate the approximate cost of the route. We use Google to calculate the straight-line distance from the initial JNE office to the destination JNE office. Then, we try to calculate the shortest distance using the Algorithm formula A*.

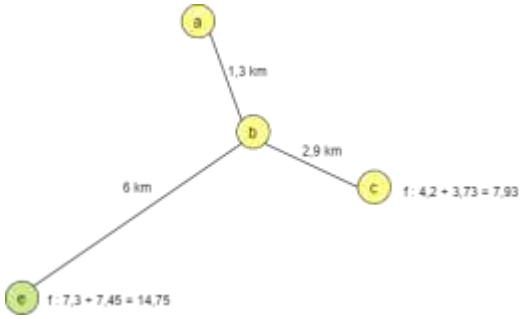
Step 1



Open = [b]

Close = [a]

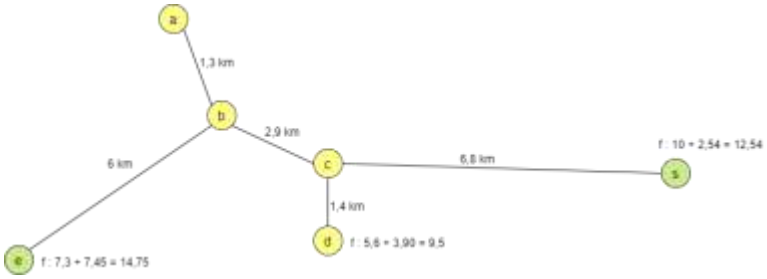
Step 2



Open = [e]

Close = [a b c]

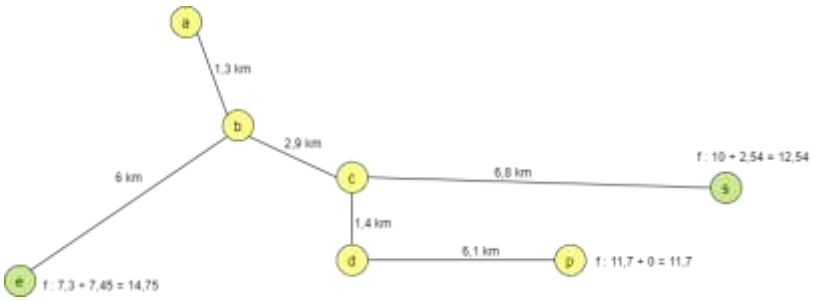
Step 3



Open = [e s]

Close = [a b c d]

Step 4



Open = [e s]

Close = [a b c d p]

Because p (JNE Mampang representative office) is the destination point, and also the shortest distance between the nodes in the open list. This means that a solution has been found.

The route and the total cost can be traced using this a star algorithm method. The search results generated from the graph above are the routes to JNE Kebayoran Lama (a), JNE Kebayoran Center (b), JNE Agen Ban (c), JNE Radio Dalam (d), and JNE Mampang Representative Office (p) with a total distance equal to 11.7 km, this route is the route with the shortest distance generated from this algorithm

2.2 Implementation Of Java Programming Language

In the program, we use 4 classes consisting of Main, AStarSearch, Node, and Edge. We use java language to run our A Star algorithm where it uses its Classes to find the fastest path.

```
public class Node {
    private String nama;
    private double gn;
    private double fn;
    private double hn;
    private List<Edge> childlist;
    private Node parentnode;

    @Override
    public String toString(){
        return this.nama+"("+this.fn+")";
    }

    public Node(String nama, double hn){
        this.nama=nama;
        this.hn=hn;
        this.childlist=new ArrayList<Edge>();
    }
```

Figure 3. *screenshot program java kelas Node.java*

The Node class is a class that represents every point on the map. each place is practically a node used by a java program. But the node can't know what the open node next to it is without having what we created which is the Edge Class.

```
public class Edge {
    private double cost;
    private Node targetnode;

    public Edge(Node target, double cost) {
        targetnode=target;
        this.cost=cost;
    }
```

```
node1.connect(new Edge(node2, 1.3));

node2.connect(new Edge(node3, 2.9));
node2.connect(new Edge(node5, 6));

node3.connect(new Edge(node10, 6.5));
node3.connect(new Edge(node4, 1.4));

node4.connect(new Edge(node16, 6.1));
node4.connect(new Edge(node10, 1.6));
node4.connect(new Edge(node11, 4.7));

node5.connect(new Edge(node6, 3));

node6.connect(new Edge(node7, 2));

node7.connect(new Edge(node8, 4.6));
```

Figure 4. *screenshot program java kelas Edge.java*

Edge class is used to determine the other nodes that are next to it and also determine the cost or distance between places on the map. by using the connect method that connects one node to another node.

```
while(open.isEmpty() == false){
    Node best;
    if (open.contains(goal)){
        int position = open.indexOf(goal);
        best = open.get(position);
    } else {
        //selection sort
        for (int i=0; i< open.size()-1; i++ ){
            int min=i;
            for(int j = i+1; j<open.size(); j++) {
                if(open.get(j).getFn() < open.get(min).getFn()) {
                    min=j;
                }
            }
            Node temp = open.get(min);
            open.set(min, open.get(i));
            open.set(i, temp);
        }
        best = open.get(0);
```

```
for(Edge e:best.getChildlist()){
    Node child = e.getTargetnode();
    double tempcost = e.getCost();
    double tempGn = best.getGn()+ tempcost;
    double tempFn = tempGn + child.getHn();

    if (closed.contains(child)){
        continue;
    } else if (!open.contains(child)) {
        child.setFn(tempFn);
        child.setGn(tempGn);
        child.setParentnode(best);
        if (open.contains(child)) {
            open.remove(child);
        }
        open.add(child);
    }
}
```

Figure 5. *screenshot program java kelas Edge.java*

3. Results and Discussion

The time complexity of the A* algorithm itself is determined by the heuristic value. In the worst case, the number of nodes being expanded can be exponential in the shortest path solution. However, the complexity can also be a polynomial if the heuristic function h meets the following conditions:

$$|hx - hx| = \mathrm{O} \log_h x$$

Where h is the optimal heuristic, or the exact cost to get to the goal of x. in other words, the errors of h must not grow faster than the "perfect heuristic" algorithm h which returns the actual distance from x to the destination.

In this section, we show the previously created java program when run:

```
run:
1: Kebayoran Lama
2: Kebayoran Center
3: Agen Ban
4: Radio Dalam
5: Agen Bintaro Permai
6: Kesehatan
7: Fiyora
8: Agen MRT Lebak Bulus
9: Lebak Bulus Agent
10: Refin
11: Juang Amirain
12: Agen Ripqi Hidayat
13: Kemang Selatan
14: Jl Margasatwa
15: TB Simatupang
16: Kantor Perwakilan Mampang
17: Berkah Jaya
18: Kantor Perwakilan Pasar Minggu
19: Agen Pacorant Tebet
20: Agen Assembaris

Pilih Tempat Awal Anda dengan mengetik nomor yang sesuai
1

Pilih Tujuan Anda dengan mengetik nomor yang sesuai
5
```

Figure 6. *Program Homepage*

This form is used as the main menu display, which can be seen as above. The user will choose the starting place for the JNE courier trip by typing the number according to the name of the location in the list of places after that type the destination.

Then the program will immediately execute looking for the fastest route between the two JNE offices, the results issued by the program will look like:

```
     OPEN : [1:Kebayoran Lama(0.0)]
     CLOSED : []
***BEST NODE : 1:Kebayoran Lama***
     OPEN : []
     CLOSED : [1:Kebayoran Lama(0.0)]


     OPEN : [2:Kebayoran Center(1.8)]
     CLOSED : [1:Kebayoran Lama(0.0)]
***BEST NODE : 2:Kebayoran Center***
     OPEN : []
     CLOSED : [1:Kebayoran Lama(0.0), 2:Kebayoran Center(1.8)]


     OPEN : [3:Agen Ban(9.2), 5:Agen Bintaro Permai(11.8)]
     CLOSED : [1:Kebayoran Lama(0.0), 2:Kebayoran Center(1.8)]
***BEST NODE : 5:Agen Bintaro Permai***
     OPEN : [3:Agen Ban(9.2)]
     CLOSED : [1:Kebayoran Lama(0.0), 2:Kebayoran Center(1.8), 5:Agen Bintaro Permai(11.8)]


Rutenya : 1:Kebayoran Lama -> 2:Kebayoran Center -> 5:Agen Bintaro Permai -> done

Costnya : 7.3
```

Figure 7. *Program result*

As an example, we use the JNE Kebayoran Lama office to the Bintaro Permai Agent Office. It can be seen that to go to the Bintaro Permai Agent Office, the courier skipped Kebayoran quickly. Below that, we can also see the cost, which is the distance from the starting and destination points, in this example, the distance is 7.3 km.

Above it, we can see the A* algorithm and how it finds the fastest route. The first one we have the old Kebayoran Node being the first open node, and because this is the first step, it becomes the best node. In the next step, the old kebayoran has become a closed node, and the open node is the kebayoran center. He became the best node, the next one is also the same as before and the algorithm chooses the best node between the Ban Agent Office and the Bintaro Permai Agent, he chooses the Bintaro Permai Agent Office because that is the final destination of the courier.

4. Conclusions

The A* algorithm by calculating the smallest heuristic value is proven to be very efficient in finding the shortest path because the A* algorithm compares the initial value to the destination by choosing the best solution, to minimize wasted time and costs.

References

Nugraeni, Rosita Ayu, Mulyono, Mulyono & Rochmad, Rochmad. "Penerapan Algoritma A* Dalam Penyelesaian Rute Terpendek Pendistribusian Barang". *UNNES Journal of Mathematics*, Vol 4 No.1, 2015

Nuryoso Y H."Implementasi Algoritma A-Star Untuk Mencari Rute Terpendek Angkutan Umum Kota (Studi Kasus Pada Rute Angkutan Umum Kota di Kota Sukabumi)". *Jurnal Sarjana Teknik Informatika*,Vol.8 No.1, 1 Mei 2021.

Wahyudinur, A. "Pencarian Jalur Terpendek Pengiriman Barang Menggunakan Algoritma A* Studi Kasus Kantor Pos Besar Medan". *Jurnal Riset Komputer*, Vol. 3, No.1, 2016