

ISSN 2085-692X

B.0/B.9



INFORMATICS DEPARTMENT  
FACULTY OF INDUSTRIAL TECHNOLOGY  
UPN VETERAN JATIM

The 1<sup>st</sup> International Seminar



# PROCEEDINGS ITIS 2009

[www.itis-upn.net](http://www.itis-upn.net)







INFORMATICS DEPARTMENT  
FACULTY OF INDUSTRIAL TECHNOLOGY  
UPN VETERAN JATIM

The 1<sup>st</sup> International Seminar



# PROCEEDINGS ITIS 2009

Information and Communication Technology International Seminar

[www.itis-upn.net](http://www.itis-upn.net)

Proceeding of The First Annual International Conference  
**Information and Communication Technology  
International Seminar (ITIS)**  
Volume 1, Number 1, July 2009

**Executive Board**

Rector of UPN "Veteran" Jawa Timur  
Dean of Industrial Engineering Faculty

**Technical Program Committee**

**Dr. Nicolas Spyrtos**  
University of Paris-South, France

**Dr. Choong Yeow Wei**  
HELP University College, Malaysia

**Emanuel S. Grant, Ph.D**  
University of North Dakota, USA

**Dr. Endra Julianto**  
ITB, Indonesia

**Prof. Ir.Drs.Ec. Ryanarto Sarno, MSc. PhD**  
ITS, Indonesia

**Drs. Jazi Eko, MSc, PhD**  
UGM, Indonesia

**Ir. Era Purwanto, M.Eng**  
ITS, Indonesia

**Prof. Dr. Ir. Akhmad Fauzi, MMT**  
UPN "Veteran" Jatim, Indonesia



**Organizing Committee**

**Organizing Committee Chairman**  
Dra. Wulan Retno W., M.Pd

**Secretariat and Registration**  
Intan Yuniar P, S.Kom  
Harianto M., SE

**Website and Publication**  
Rizky Parluka, S.Kom  
Hj. Asti Dwi I., SKom, Mkom

**Proceeding**  
Basuki Rahmat, S.Si., MT  
Budi Nugroho, S.Kom

**Sponsorships**  
I Gede Susrama, ST, MKom  
Prisa Marga K., S.Kom

**Program and Accomodation**  
M. Irwan Afandi, ST., MSc.  
Fetty Tri A., S.Kom  
Nur Cahyo Wibowo, S.Kom, M.Kom  
Doddy Ridwandono, S.Kom  
Ir. H. Kemal Wijaya, M.TP  
Zainal Abidin, S.Sos., M.Si  
Ir. Kartini, MT  
Achmad Junaidi, S.Kom

**Contact Address**

Informatics Departement, UPN "Veteran" Jatim  
Jl. Raya Rungkut Madya, Surabaya 60294  
Homepage: <http://www.itis-upn.net/>  
Email: itisupn@yahoo.com

## Table of Content

EXECUTIVE BOARD .....	i
ORGANIZING COMMITTEE .....	ii
PREFACE .....	iii
TABLE OF CONTENT .....	iv
1. INFORMATION QUALITY MEASUREMENT AND IMPROVEMENT (CASE STUDY: X UNIVERSITY - BANDUNG) Eva Hariyanti .....	1 - 7
2. IMPLEMENTATION OF FUZZY LOGIC CONTROLLER ON STEREO NOSE MOBILE ROBOT Faisal Hadi, Muhammad Rivai, Djoko Purwanto .....	8 - 11
3. THE APPLICATION OF HEURISTIC METHOD IN THE ANALYSIS OF A FIRE DEPARTMENT LOCATION IN SURABAYA Gunawan, Yong Saputra, Marini .....	12 - 18
4. SEMANTIC WEB MINING ON TOPIC MAP USING GALOIS LATTICE Gunawan, Lukman Zaman PCSW, Tri Kurniawan Wijaya, Novieana Dewi Sugianto .....	19 - 24
5. BRAIN-COMPUTER INTERFACE AS A MEAN FOR HUMAN-MACHINE INTERACTION Indar Sugiarto .....	25 - 29
6. ANALYSIS OF GAUSSIAN NOISE REDUCTION OF GENDING JAWA MUSIC RECORDS IN AUDIO CASSETTE TAPES Hugeng, Endah Setyaningsih, Hadian Satria Utama .....	30 - 34
7. SUCCESS FACTORS OF OPEN SOURCE SOFTWARE PROJECTS USING DATAMINING TECHNIQUE Jazi Eko Istiyanto, Andi Wahyu Rahardjo Emanuel .....	35 - 41
✓ 8. A COUNTER-PROPAGATION ARTIFICIAL NEURAL NETWORK TO INTRODUCTION OF TWO DIMENSION IMAGE Paryati .....	42 - 46
✓ 9. THE IMPLEMENTATION OF KRUSKAL'S ALGORITHM FOR MINIMUM SPANNING TREE IN A GRAPH Paryati .....	47 - 51
10. DEVELOPMENT OF CONTENT MANAGEMENT SYSTEM -BASED APPLICATION OF QUALITY ASSURANCE IN ORDER TO SUPPORT OF STRATEGIC PLANNING TO BE THE WORD CLASS UNIVERSITY (CASE STUDY: UNIVERSITY OF PARAMADINA) Yeni Nuraeni .....	52 - 57
11. FRAMEWORK LOGISTICS INFORMATION SYSTEM ON RFID BASIS CASE STUDY: BUSINESS UNIT LOGISTICS PT POS INDONESIA Nanang Ismail, Saepudin Nirwan, Liane Okdinawati .....	58 - 62



# THE IMPLEMENTATION OF KRUSKAL'S ALGORITHM FOR MINIMUM SPANNING TREE IN A GRAPH

Paryati

Department of Information Technique

UPN "Veteran" Yogyakarta

Email: [yaya\\_upn\\_cute@yahoo.com](mailto:yaya_upn_cute@yahoo.com)

## ABSTRACT

Kruskal's algorithm is an algorithm used to find minimum spanning tree in a graph connectivity which gives an option to keep processing the edge limit with the least weight. In a Kruskal's algorithm, the weight sorting of rib value will ease the finding of the shortest path. This algorithm has independent nature which will ease and enhance the path track making.

**Keywords:** Kruskal's Algorithm, Spanning Tree.

## 1. INTRODUCTION

### 1.1 Background

The theories about graphs began in 1736 and several important discoveries in graph theory were obtained in the 19th century, but not until the 1920s that the interest in graph theory bloomed. The first text about graph theory emerged in 1936. Undoubtedly, one of the reasons for interest in graph theory is due to its application in many fields, including computer science, chemistry, operation research, electrical engineering, linguistics, and economy.

The research will address the definition of graph and some terminologies along with a basic example of a graph. The shortest path algorithm which will be used is Kruskal's algorithm. It will show us how to find the shortest path between two given points.

Figure 1.1. shows a highway system that must be supervised by the investigator officer. Particularly, this investigator must make trips to all roads and make reports about their condition, the clarity of roads path, the condition of traffic signs, and so on. Because she lives in Yogyakarta, the most economic way in examining all roads must be started in Yogyakarta.

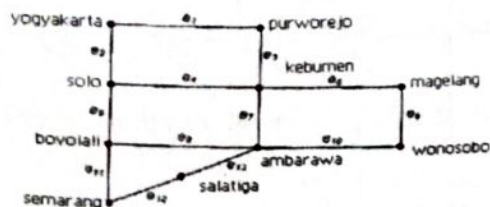


Figure 1.1 Shows a highway system that must be supervised by the investigator officer

## 2. LITERATURE REVIEW

### 2.1 The Definition of Graph and Digraph

A graph (or an undirected graph)  $G$  consists of a set  $V$  form vertices (nodes or points) and an  $E$  set from ribs (arcs) as such that every rib of  $e \in E$  is associated with the

unordered vertex pair. If there is an  $e$  rib that connects vertices  $v$  and  $w$ , it is written as  $e = (v, w)$  or  $e = (w, v)$ . In this context,  $(v, w)$  suggest a rib between  $v$  and  $w$  is in an undirected graph and not an ordered pair.

A directed graph or digraph  $G$  consists of a set  $V$  from vertices (nodes or points) and an  $E$  set from ribs (arcs) as such that every rib of  $e \in E$  connects the ordered vertex pair. If there is a single  $e$  rib that connects the ordered pair  $(v, w)$  from vertices, it is written as  $e = (v, w)$ , which suggests a rib from  $v$  to  $w$ .

An  $e$  rib in a graph (directed or undirected) which connects the vertices pair  $v$  and  $w$  is said to be incidental on  $v$  and  $w$ , then  $v$  and  $w$  is said to be incidental on  $e$  and referred to as adjacent vertices. If  $G$  is a graph (directed or undirected) with vertices  $v$  and ribs  $E$ , then it is written as  $G = (V, E)$ . If it is not specifically mentioned, the set  $E$  and  $V$  are assumed as finite and  $V$  is assumed as not empty.

In the case in the figure 1.1. above after the model making it will appear as follows:

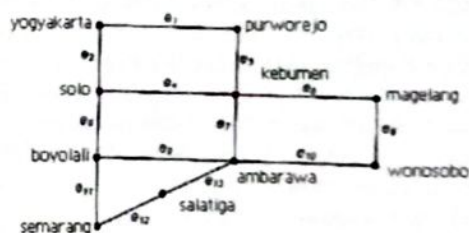


Figure 2.1. A Graph Model form a highway system shown in Figure 1.1

The points in figure 1.1. are called vertices and the lines which connect those vertices are called ribs (edge). Every vertex is named by taking the first three letters of the appropriate city. The ribs are signed by  $e_1, \dots, e_{13}$ . In drawing a graph, the only important information are the imaginary vertices linked by the imaginary ribs as well. If we start from a vertex  $v_0$ , walk along a rib to vertex  $v_1$ , walk along another rib to vertex  $v_2$  and so on, and finally arrive at vertex  $v_n$ , we refer to such complete trip as path from  $v_0$  to  $v_n$ .

### 2.2 The Definition of Spanning Tree

A  $T$  tree is a Spanning Tree of graph  $G$  if  $T$  is a sub graph of  $G$  which contains all vertices of  $G$ . The characteristics of Spanning Tree are:

- All graph  $G$  links in this tree must not form a circle.
- A graph  $G$  is said to be a tree if and only if  $G$  is linked or form a path.
- A linked graph  $G$  is said to be a tree if and only if its every edge forms a rib.



d. A linked graph G is said to be a tree if and only if it has an N peak and N-1 edge.

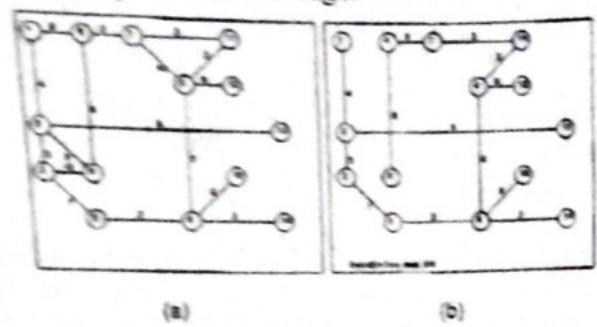


Figure 2.1: Tree

From the figure above, it can be stated that figure (a) is a wrong figure because it is not fit the characteristics of a tree as mentioned earlier. Figure (b) is a correct tree because it fulfills the characteristics of a tree.

### 2.3 The Definition of Kruskal's Algorithm

Kruskal's algorithm is one to find minimum spanning tree in a graph connectivity which gives options of always processing the edge limit with the least weight.

This algorithm run by considering the graph's biggest edge limit when searching for track in node in a graph which has been put into a spanning tree. If an edge limit is considered will integrate (with one of the points not in the spanning tree), or the integration of a point in spanning tree (one of which is not in the spanning tree), then the edge limit and the final point is included in the spanning tree. Considering one of the edge limits, algorithm continues by considering the next larger edge limit weight. In this term, when the edge limit weight values are the same, then it is unnecessary to determine which weight must be chosen first. Algorithm will stop when every point have been included in the spanning tree.

Note that, when spanning tree was made, there was a possibility that the edge limit is not connected with the part of the main tree, although in the end they will integrate with the main three.

### 2.4 Source Code

```
public class Algorithms {
    public static Graph KruskalsAlgorithm
    (Graph g) {
        int n = g.getNumberOfVertices();
        Graph result = new GraphAsLists (n);
        for(int v = 0; v < n ; ++v )
            result.addVertex (v);
        PriorityQueue queue = new BinaryHeap
        (g.getNumberOfEdges());
        Enumeration p= g.getEdges();
        while (p.hasMoreElements()) {
            Edge edge = (Edge) p.nextElement();
            Int weight= (Int) edge.getWeight();
            queue.enqueue (new Association
            (weight, edge));
        }
        Partition partition = new
        PartitionAsForest(n);
        while (!queue.isEmpty() &&
        partition.getCount() > 1) {
            Association assoc = (Association)
            queue.dequeueMin();
            Edge edge = (Edge) assoc.getValue ();
```

```
int n0 = edge.getV0 ().getNumber ();
int n1 = edge.getV1 ().getNumber ();
Set s = partition.find (n0);
Set t = partition.find (n1);
If (s !=t) {
    partition.join (s,t);
    result.addEdge (n0,n1);
}
return result;
}
```

## 3. RESEARCH METHODOLOGY

### 3.1 Algorithm of Kruskal's Algorithm

The first research conducted used the testing method on Kruskal's algorithm step-by-step. The algorithms are as follows:

E(1) is a set of ribs from Minimum Genetic Tree

E(2) is a set of the remaining ribs

V is a set of vertices

n is node or point

Steps:

E(1)=0,E(2)=E

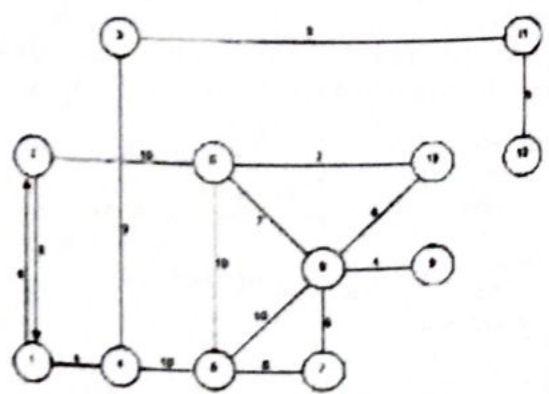
WHILE E(1) consists less than n-1 ribs and E(2)=0 DO

- Form the side of E(2) choose one with the lowest value → e(ij)
- E(1)=E(2)-{ e(ij)}
- IF V(i), V(j) is not included in the same tree, THEN
- Unite tree from V(i) and V(j) become one single tree
- END of IF
- END of WHILE
- END of Algorithm

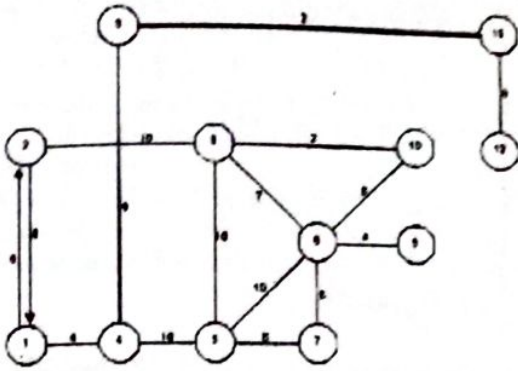
## 4. ANALYSIS

In conducting analysis for Kruskal's algorithm a test is employed by having implementation over all two algorithms with the case example as follows:

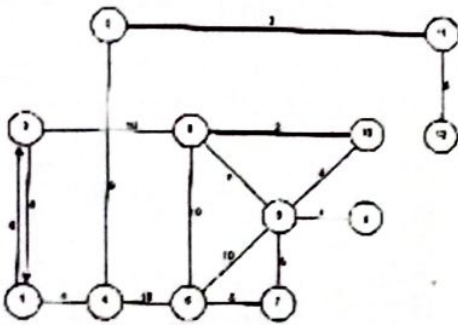
Kruskal's Algorithm



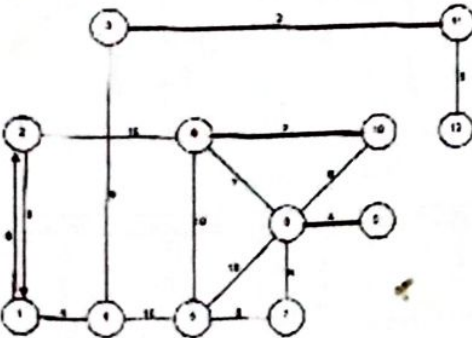
Initial step: if we look at the characteristics of a tree, then the above graph does not fulfill the characteristics of a graph, because the track is circular. Thus, we will include Kruskal's algorithm to solve the problem above. The first step to run Kruskal's algorithm is arranging e (rib weight) in the right order from the minimum value to the maximum. After the arrangement (Exhibit 4.1.a) then we make the track for both vertices with minimum weight.



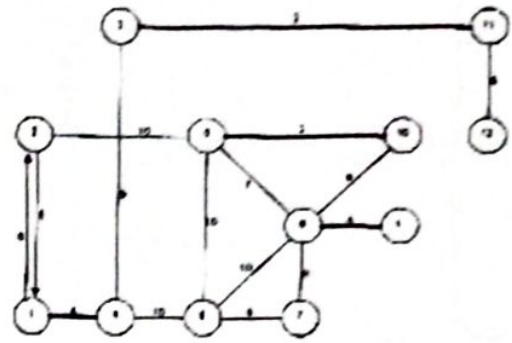
Step 1: in the line with the table order, then what to do first is making track V3 – V11 with the rib weight 2 (track with bold lines).



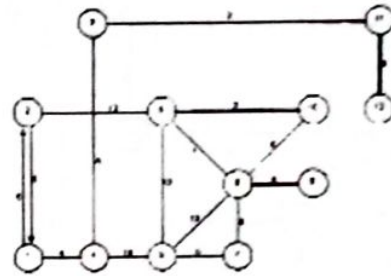
Step 2: the next order is V6 – V10 with the rib weight 2. It appears that the track is separated from step 1. However, considering the characteristics of the independent Kruskal's algorithm which means that there is a possible edge limit (vertex) unconnected with the part of the main tree, although eventually will join the main tree, it can happen.



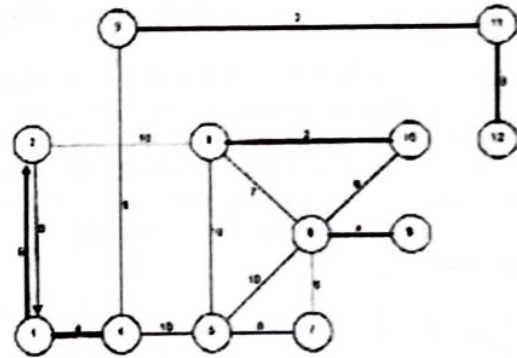
Step 3: the next order is V8 – V9 with the rib weight 4.



Step 4: the next order is V1 – V4 with the rib weight 4.

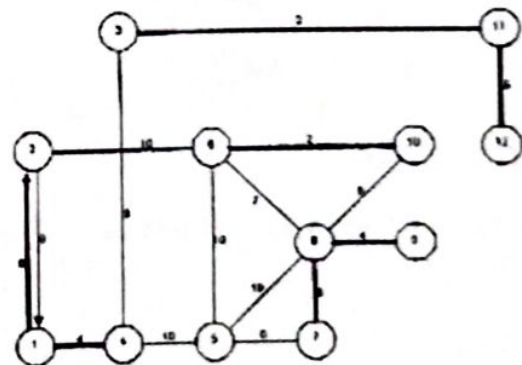


Step 5: the next order is V11 – V12 with the rib weight 6.



Step 6: the next order is V1 – V2 with the rib weight 6.

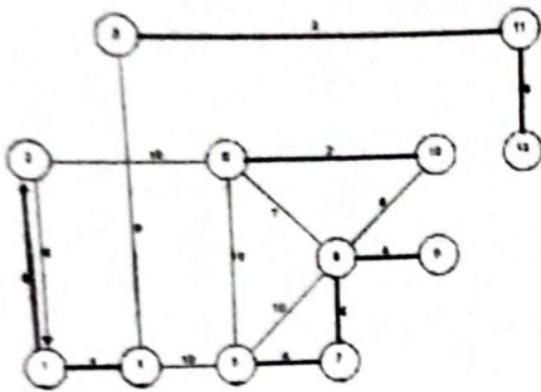
6.



Step 7: the next order is V8 – V7 with the rib weight 6.

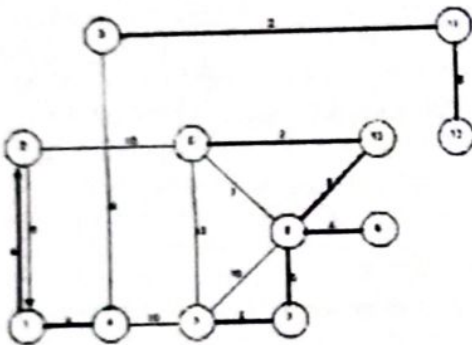
6.





Step 8: the next order is V5 – V7 with the rib weight

6.

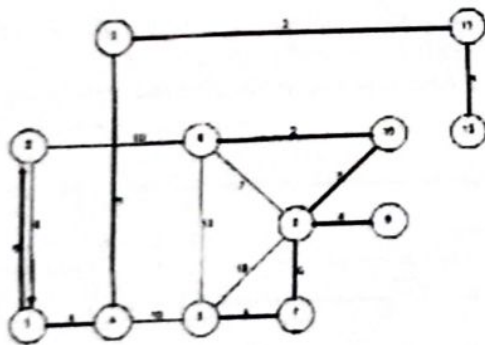


Step 9: the next order is V8 – V10 with the rib weight

6.

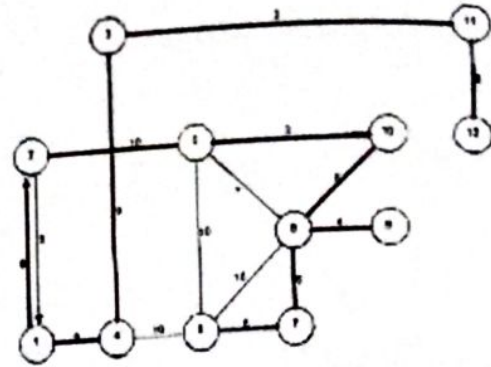
Step 10: the next order is V6 – V8 with the rib weight 7. This path is not allowed because it will form circle in path V6 – V8 – V10.

Step 11: the next order is V2 – V1 with the rib weight 8. This path is not allowed because vertex V2 – V1 has been passed. In fact, the requirement of a tree only allows it to be passed once.



Step 12: the next order is V3 – V4 with the rib weight

9.



Step 13: the next order is V2 – V6 with the rib weight 10.

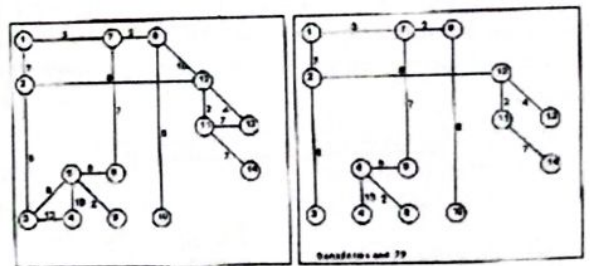
Step 14-16: this order is unnecessary to be done because it forms a circle.

Table 4.1. Step of path determination in Kruskal's algorithm

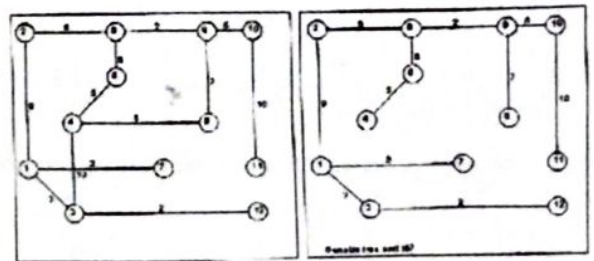
Step	v	g	
1	3-11	2	add to tree
2	6-10	2	add to tree
3	8-9	4	add to tree
4	1-4	4	add to tree
5	11-12	6	add to tree
6	1-2	6	add to tree
7	8-7	6	add to tree
8	5-7	6	add to tree
9	8-10	6	add to tree
10	6-8	7	reject because it forms a circuit
11	2-1	8	reject because it forms is turned back
12	3-4	9	add to tree
13	2-6	10	add to tree
14	6-5	10	reject because it forms a circuit
15	4-5	10	reject because it forms a circuit
16	5-8	10	reject because it forms a circuit

Total bobot → 61

Other examples of Kruskal's Algorithm are as follows:



Case (b) Solution



Case (b) Solution

## 5. CONCLUSION

After test and comparison between Kruskal's algorithm, then conclusion can be drawn as follows:

### 1. Strength

- The existence of weight sorting will ease the searching of the shortest path.
- Considering the characteristics of Kruskal's independent algorithm, it will ease and enhance the formation of path track.

2. **Weaknesses**

If the number of vertices is very large, it will slower than the Dijkstra's algorithm because it must sort thousands of vertices first, then forming the path.

## REFERENCES

- [1] AKL, Selim G, 1999, The Design and analysis of Kruskal Algorithms, New Jersey Prentis Hall Inc
- [2] Budd Timothy, 1991, Object Oriented Programming Reading, Addison Wesley Publisher co Inc
- [3] Finkel, R.A and J.L. Bentley, 1999, Quadtrees A Data Structures For Retrieval on Composite Key, Acta Informatika Vol 4. No.1
- [4] Richard Johnsonbaugh, 1999, Discrete Mathematics 4th ed., Prentice-Hall, New Jersey
- [5] Quinn, Michael, 1994, Kruskal Computing Theory and Practice Second edition Singapore Mc Graw Hill