Comparison of Floyd-Warshall Algorithm and Greedy Algorithm in Determining the Shortest Route

1st Huzain Azis
Faculty of Computer Science
Universitas Muslim Indonesia
Makassar, Indonesia
huzain.azis@umi.ac.id

2nd Rizaldi dg. Mallongi
Faculty of Computer Science
Universitas Muslim Indonesia
Makassar, Indonesia
rizaldidgm@umi.ac.id

3rd Dirgahayu Lantara
Faculty of Industrial Technology
Universitas Muslim Indonesia
Makassar, Indonesia
dirgahayu.lantara@umi.ac.id

4th Yulita Salim
Faculty of Computer Science
Universitas Muslim Indonesia
Makassar, Indonesia
yulita.salim@umi.ac.id

***Abstract*—Prior to traveling, one of the most important things to pay attention to is to determine the travel route, especially the shortest path to be taken. In this study, the method used to determine the shortest route is the conventional method and the Heuristic method. These two methods will be compared to find out methods which can provide the best result. For conventional methods, the authors use the Floyd-Warshall algorithm whereas for the Heuristic method, the greedy algorithm is employed. The Floyd-Warshall algorithm takes into account all possible routes so that there are some routes are displayed while the greedy algorithm checks every node that is passed to select the shortest route (Local Optimum) so that the time needed in searching is faster. Based on the conducted testing, the final result obtained is the Floyd-Warshall algorithm provides a better solution, namely the mileage was 22.7 km and the greedy algorithm covered a mileage of 24.8 km. This result indicated that a longer time is required because it takes the distance to all points into account.**

Keywords—greedy, floyd-warshall, shortest route, heuristic

I. INTRODUCTION

The development of digital mapping technology known as GIS (Geographic Information System) provides many methods to solve spatial allocation problems, for example solutions in the field of networks and transportation. One of the main problems that has become the focus of various studies is the computation on the search for the shortest route at a different location [1],[2]. Determining the shortest route is the problem of determining the shortest path or route from the starting point to the destination point [1]. To determine the route in this study, the author uses conventional methods and Heuristic methods. The author aims to compare the method to find the method that provides the best or the most optimum results, using the Floyd-Warshall algorithm for conventional methods and the greedy algorithm for heuristic methods. The Floyd-Warshall algorithm requires less time compared to the greedy algorithm, it is because the iterations number is the same as the number of available nodes. So, the more nodes, the more iterations will be proceeded and the more process of calculating possible routes to all points. The process of calculating all possibilities on each route or side to all nodes can be seen on a graph [4-7].

While the greedy algorithm, the process is carried out only at the point visited, if it stops at a node, the node will be checked as a destination or not, otherwise, it will proceed to the next node by selecting the smallest or optimum side between nodes, this process will be repeated until the destination point is found. The two algorithms discussed in this study have their advantages and disadvantages in determining the shortest route [3].

II. METHODOLOGY

A. Determination of the Shortest Route

In determining the shortest route can be done using two methods, namely conventional methods and heuristic methods. The conventional method uses mathematical calculations while the heuristic method uses the calculation of artificial intelligence by making its knowledge base and calculations.

1) Conventional Method

There are several conventional methods commonly used to search for the shortest routes, including the Floyd-Warshall algorithm, the Dijkstra algorithm, and the Bellman-Ford algorithm. The conventional method is an algorithm that uses ordinary mathematical calculations.

2) Heuristic Method

The Heuristic method is a sub-field of artificial intelligence to search and determine the shortest route faster. There are several algorithms in the heuristic method commonly used in the search for the shortest route, such as greedy algorithm, ant algorithm, and genetic algorithm.

B. Floyd-Warshall Algorithm

The Floyd-Warshall algorithm is an example of the application of dynamic programming to solve the shortest route search problem [7]. This method solves the problem by looking at solutions based on previous solutions that are interrelated with one another and allows for more than one solution [7][8].

The algorithm that Warshall found to find the shortest route is an algorithm that is simple and easy to implement. The Floyd -Warshall algorithm has a directed and weighted input graph (V, E), which is a list of points (nodes, V) and side lists (E). Line weight e can be given the symbol w (e). The number of side weights on a path is the total weight of the path. The side on E is allowed to have a negative weight, but it is not permissible for the W_{ij} graph to have a negative weight cycle. This algorithm calculates the smallest weight of all paths that connect a point pair and does it all at once for all pairs point by point until it reaches the destination point with the minimum number of weights [6][9].

The Floyd-Warshall algorithm to find the shortest route is shown in the following

- $W = W_0$
- for k = 1 to n, do:
- for i = 1 to n, do:
- for j = 1 to n, do:
- if $W[i,j] > W[i,k] + W[k,j]$ then swap $W[i,j]$ with $W[i, k] + W[k,j]$.
- $W^* = W$

For example, W0 is the initial adjacency matrix with a weighted graph. W * is the shortest adjacency matrix and Wij is the shortest route from point V_i to V_j.

The algorithm above only calculates the shortest total distance to all points but does not explain the route that is passed so as to produce the shortest distance [10]. To be able to find out the route that is traversed so that the shortest distance is found, it must be added with a square matrix Z of size n x n, which is arranged in (1).

$$Inisialisasi\ \mathrm{Z}^{(0)}{}_{\mathrm{ij}} = \begin{cases} j & jika\ W_{ij}^{(0)} \neq \infty \\ 0 & jika\ W_{ii}^{(0)} = \infty \end{cases} \qquad (1)$$

In the K iteration if there is an exchange of values between W_{ij} and $W_{ik} + W_{kj}$, then change the value of Z_{ij} with the value of Z_{ik}. In order to the search process to be more efficient, the change of the Z matrix with the shortest iteration of the search route should be done.

The revision of the Floyd-Warshall algorithm by involving the process of finding the shortest route through the Z matrix is as follows

- $W = W_0$; $Z = Z_0$
- for k = 1 to n, do:
- for i = 1 to n, do:
- for j = 1 to n, do:
- if $W[i,j] > W[i,k] + W[k,j]$ then:
- swap $W[i,j]$ with $W[i, k] + W[k,j]$
- replace $Z_{i,j}$ with $Z_{i,k}$
- $W^* = W$; $Z^* = Z$

Z * matrix is a matrix that shows the route each node must pass to the destination node.

In its iteration to find the shortest route, the Floyd Warshall algorithm forms n matrices, according to the k-iteration. This causes the process to be slow, especially if the value of n is large. Although the processing time is not the fastest, the Floyd-Warshall algorithm is often used to calculate the shortest route because of its simplicity.

C. Greedy Algorithm

Greedy algorithm is an algorithm that solves a problem step by step (step by step) [5].

- Take the best choice that can be obtained at that time (Local Optimum),
- The decision taken in the previous step cannot be changed or in other words it cannot return to the previous step,
- It is expected that selecting the Local Optimum at each step can produce a Global Optimum at the end of the algorithm.

The greedy algorithm assumes that Local Optimum is part of Global Optimum. In the optimization problem, the greedy algorithm is composed of the following elements:

1) Candidate Association (C)

The set that contains the elements forming the solution where each candidate will take one step.

2) Set of Solution (S)

This set contains candidates who have been chosen as solutions to the problem. This set is a subset of the Candidate Association.

3) Selection Function

It is a function that selects candidates who are most likely to get optimal solutions. The chosen candidate will not be considered again in the next step.

4) Feasible Function

It is a function that checks whether a selected candidate is able to provide a viable solution, where the candidate together with the set of solutions that have been formed does not violate the existing constraints. Candidates who are eligible are included in the solution set whereas otherwise candidates who are not eligible will be discarded and will not be reconsidered.

5) Objective Function

Functions that maximize or minimize the value of the solution. The steps taken in the greedy algorithm to find the shortest route are as follows:

- Check whether the current node is the initial node. If yes go to step (3), if not go to step (2).
- Check whether the current node is the destination node. If not go to step (3).
- Check all the sides connected to the current node and the next node.
- Look for the side with the lowest weight to the next node (Local Optimum).
- Move to the next node according to the specified local optimum.
- Return to step (1) until the destination node is found

III. Results and Discussion

Problems that sometimes occur are routes to tourist sites. In this case, the route to *Losari Beach*, and the initial location is from *Sultan Hasanuddin Airport*. In the data collection stage, the writer represents route data into graph form so that it can be used in the completion of the two algorithms used, shown in Fig. 1.

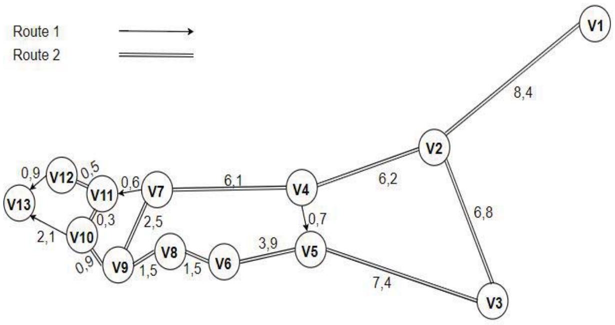

Fig. 1. Representation of the route to Losari Beach

The names of the regions passed from *Sultan Hasanuddin Airport to Losari Beach* are symbolized by a circle (node) V1 to V13, shown in Table I.

TABLE I. DESCRIPTION OF REGIONAL NAMES

Node	Regional Names
V1	Bandara Sultan Hasanuddin
V2	Daya
V3	Moncongloe
V4	Jl. Urip Sumiharjo
V5	Jl. Abdullah Dg. Sirua
V6	Jl. A.P. Pettarani
V7	Jl. G. Bawakaraeng
V8	Jl. Sungai Saddang
V9	Jl. Mongisidi
V10	Jl. Hj. Bau
V11	Jl. Jend. Sudirman
V12	Jl. Chairil Anwar
V13	Pantai Losari

The weight of each side from one node to another node expresses the distance from one place to another.

A. Floyd-Warshall Solution

From the graph in Fig. 1, the Floyd-Warshall algorithm is calculated to find the shortest path between all points in the graph. The W_0 matrix is shown in Table II, and the Z_0 matrix is shown in Table III.

TABLE II. MATRIX W_0

	V 1	*V 2*	*V 3*	…	*V 8*	*V 9*	*V 10*	*V 11*	*V 12*	V 13
V 1	0	8.4	∞	…	∞	∞	∞	∞	∞	∞
V 2	8.4	0	6.8	…	∞	∞	∞	∞	∞	∞
V 3	∞	6.8	0	…	∞	∞	∞	∞	∞	∞
V 4	∞	6.2	∞	…	∞	∞	∞	∞	∞	∞
V 5	∞	∞	7.4	…	∞	∞	∞	∞	∞	∞
V 6	∞	∞	∞	…	1.5	∞	∞	∞	∞	∞
V 7	∞	∞	∞	…	∞	2.5	∞	0.6	∞	∞
V 8	∞	∞	∞	…	0	1.5	∞	∞	∞	∞
V 9	∞	∞	∞	…	1.5	0	0.9	∞	∞	∞
V 10	∞	∞	∞	…	∞	0.9	0	0.3	∞	2.1
V 11	∞	∞	∞	…	∞	∞	0.3	0	0.5	∞
V 12	∞	∞	∞	…	∞	∞	∞	0.5	0	0.9
V 13	∞	∞	∞	…	∞	∞	∞	∞	∞	0

TABLE III. MATRIX Z_0

	V 1	*V 2*	*V 3*	*V 4*	…	*V 9*	*V 10*	*V 11*	*V 12*	*V 13*
V 1	1	2	0	0	…	0	0	0	0	0
V 2	1	2	3	4	…	0	0	0	0	0
V 3	0	2	3	0	…	0	0	0	0	0
V 4	0	2	0	4	…	0	0	0	0	0
V 5	0	0	3	0	…	0	0	0	0	0
V 6	0	0	0	0	…	0	0	0	0	0
V 7	0	0	0	4	…	9	0	11	0	0
V 8	0	0	0	0	…	9	0	0	0	0
V 9	0	0	0	0	…	9	10	0	0	0
V 10	0	0	0	0	…	9	10	11	0	13
V 11	0	0	0	0	…	0	10	11	12	0
V 12	0	0	0	0	…	0	0	11	12	13
V 13	0	0	0	0	…	0	0	0	0	13

1) Iteration k = 1

Each element in the W matrix was checked either the value of $W\ [i,\ j]$ is greater than the weight value $W\ [i,\ k]$ + $W\ [k,\ j]$. If yes, then the weight value $W\ [i,\ j]$ is replaced by the weight value $W\ [i,\ k]$ + $W\ [k,\ j]$. From the table above, for example i = 1 and j = 2, then check W [1,2] = 8,4> W [1,1] + W [1,2] = ∞, because the condition is not fulfilled, the weight of W [1,2] is not replaced. Then it will continue to be checked until the value of i and j is equal to the value of n.

2) Iteration k = 2

In this k = 2 iteration, for i = 1 and j = 3, then checking [1,3] = ∞> [1,2] + [2,3] = 15,2, due to conditions is met, then the weight of W [1,3] is replaced by the weight value of [1,2] + [2,3] which is 15,2 and it is stored in the temporary W * matrix and the value of Z1,3 is replaced by the value Z1, 2 which is stored in the temporary Z * matrix in the iteration. Then it will continue to be checked until the value of i and j is equal to the value of n. In the last iteration or iteration k = 13, the final weight value of the W * matrix is found which states the weight value of W [1.13] as the shortest distance from the initial location (V1) to the destination (V13) which is 22.7 km is shown in Table IV.

To find the route that passes according to the shortest distance from V1 to v13 can be seen in the Z* matrix shown in table V and marked with a black column.

To find the route that passes according to the shortest distance from V1 to v13 can be seen in the Z* matrix shown in Table V and marked with a yellow column.

TABLE IV. MATRIX W*

	V1	*V2*	*V3*	….	*V9*	*V10*	*V11*	*V12*	*V13*
V1	0	8,4	15,2	…	22,2	21,6	21,3	21,8	22,7
V2	8,4	0	6,8	…	13,8	13,2	12,9	13,4	14,3
V3	15,2	6,8	0	…	14,3	15,2	15,5	16	16,9
V4	14,6	6,2	8,1	…	7,6	7	6,7	7,2	8,1
V5	22,6	14,2	7,4	…	6,9	7,8	8,1	8,6	9,5
V6	26,2	17,8	11,3	…	3	3,9	4,2	4,7	5,6
V7	20,7	12,3	14,2	…	1,8	0,9	0,6	1,1	2
V8	24,7	16,3	12,8	…	1,5	2,4	2,7	3,2	4,1
V9	23,2	14,8	14,3	…	0	0,9	1,2	1,7	2,6
V10	24,1	15,7	15,2	…	0,9	0	0,3	0,8	1,7
V11	24,4	16	15,5	…	1,2	0,3	0	0,5	1,4
V12	24,9	16,5	16	…	1,7	0,8	0,5	0	0,9
V13	∞	∞	∞	…	∞	∞	∞	∞	0

TABLE V. MATRIX Z_0

	V1	**V2**	**V3**	**…**	**V9**	**V10**	**V11**	**V12**	**V13**
V1	1	2	2	…	2	2	2	2	2
V2	1	2	3	…	4	4	4	4	4
V3	2	2	3	…	5	5	5	5	5
V4	2	2	5	…	5	7	7	7	7
V5	3	3	3	…	6	6	6	6	6
V6	8	8	5	…	8	8	8	8	8
V7	4	4	4	…	11	11	11	11	11
V8	9	9	6	…	9	9	9	9	9
V9	7	7	8	…	9	10	10	10	10
V10	9	9	9	…	9	10	11	11	11
V11	10	10	10	…	10	10	11	12	12
V12	11	11	11	…	11	11	11	12	13
V13	0	0	0	…	0	0	0	0	13

B. *Greedy Algorithm Solution*

Using the greedy algorithm, the first route that must be passed to get to V13 is from node V1 to node V2 (side V1-V2) with a weight of 8.4 shown in Fig. 2.

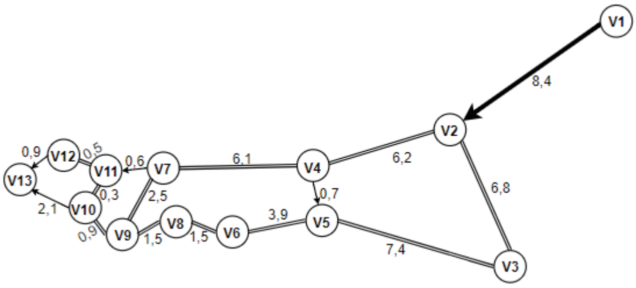

Fig. 2. The first route from node V1 to node V2

At node V2 there is an intersection or two different routes, then select the side that has the lowest weight (Local Optimum), the V2-V4 side with a weight of 6.2 (Fig. 3).

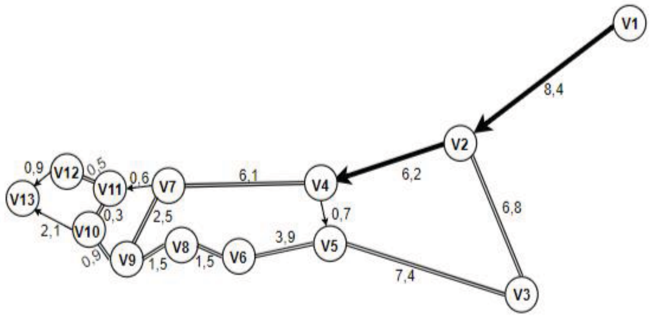

Fig. 3. The second route from node V2 to node V4

On node V4 there are intersections or two different routes, then choose the side that has the lowest weight (Local Optimum) that is the V4-V5 side with a weight of 0.7 (Fig. 4)

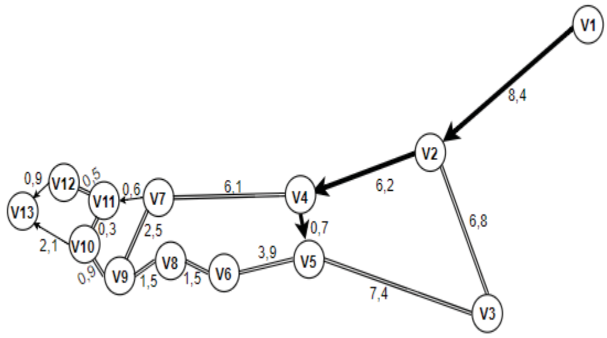

Fig. 4. The second route from node V4 to node V5

Furthermore, at node V5 there are intersections or two different routes, then select the side that has the lowest weight (Local Optimum) which is the V5-V6 side with a weight of 3.9. Then for V6 and V8 nodes, because they do not have an intersection it will go directly to node V9, with the total number of weights from V6 to V9 that pass through the V6-V8 and V8-V9 sides are 3. (Fig. 5)

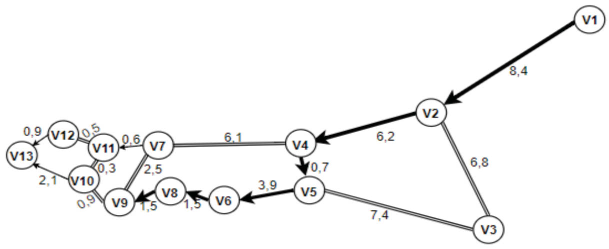

Fig. 5. The fourth, fifth and sixth routes from node V5 to node V9

On node V9 there are intersections or two different routes, then choose the side that has the lowest weight (Local Optimum), which is the V9-V10 side with a weight of 0.9 (Fig. 6).

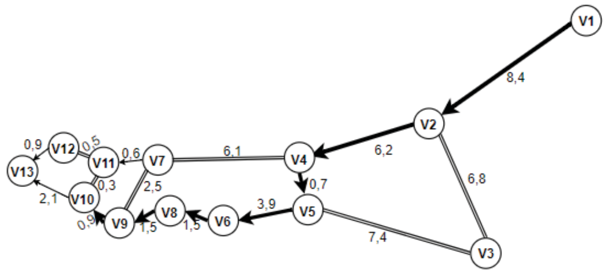

Fig. 6. The seventh route, from node V9 to node V10

Furthermore, at node V10 there are intersections or two different routes, then select the side that has the lowest weight (Local Optimum) which is the V10-V11 side with a weight of 0.3 (Fig. 7).

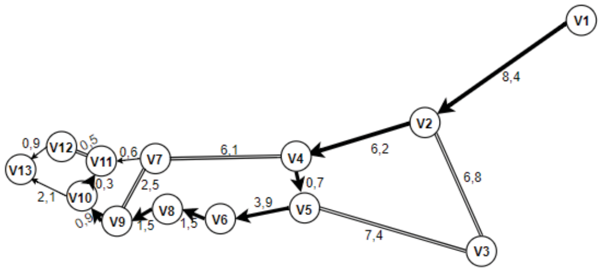

Fig. 7. The eight route, from node V10 to node V11

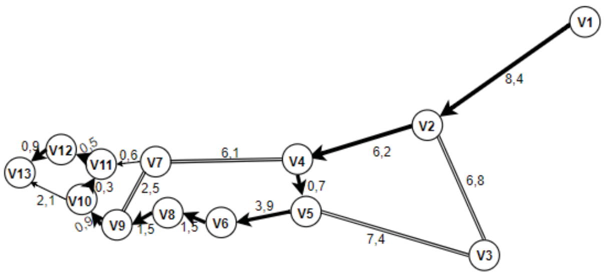

Fig. 8. The nine and ten route, from node V11 to node V13

Fig. 8 shows that at the V11 node there are no intersections or two different routes, so it will immediately choose the next side, that is, the V11-V12 side weighing 0.5. Then for node V12 does not have an intersection so go directly to the next node, V13 by passing the side V12-V13 which has a weight of 0.9.

Destination point V13 or node V13, the search is stopped. Then the total travel distance from V1 to V13 is calculated, which is the total weight of the crossing side, which is 24.8 km.

IV. CONCLUSION

Overall this research represents that the Floyd-Warshall algorithm is the shortest route generated in the path of *Sultan Hasanuddin Airport - Daya - Urip Sumiharjo - G. Bawakaraeng - Jend. Sudirman - Chairil Anwar - Losari Beach* with a distance of 22.7 km. whereas on the greedy algorithm the shortest route produced is *Sultan Hasanuddin Airport - Daya - Urip Sumiharjo - A. Dg. Sirua - PP. Pettarani - Sungai Saddang - Mongisidi - Hj. Bau - Jend. Sudirman - Chairil Anwar - Losari Beach* with a distance of 24.8 km. The data above show that the Floyd-Warshall algorithm takes more time to do the route search than the greedy algorithm, but it gives the most accurate results. This is because the Floyd-Warshall algorithm takes all points and routes into account so that they can find the best results, while the greedy algorithm only selects the lowest weight for each iteration (Local Optimum) so that the search time is faster but the final result is not always optimal (Global Optimum).

REFERENCES

[1] M. Muthulakshmi and M.M. Shanmugapriya, "Shortest Path Algorithm and its implementation," 2016 International Journal of Mathematics Trends and Technology (IJMTT) – Volume 36 Number 2- August 2016

[2] J. Leng and W. Zeng, "An Improved Shortest Path Algorithm for Computing One-to-One Shortest Paths on Road Networks," 2009 First International Conference on Information Science and Engineering, Nanjing, 2009, pp. 1979-1982.

[3] Y. Guo, Z. Qin and Y. Chang, "A novel hybrid algorithm for the dynamic shortest path problem," 2010 Sixth International Conference on Natural Computation, Yantai, 2010, pp. 2545-2550.

[4] A. Aziz, M. M. Farid and E. Suryani, "Floyd warshall algorithm with FIS Sugeno for search evacuation route optimization," 2017 International Seminar on Application for Technology of Information and Communication (iSemantic), Semarang, 2017, pp. 147-151.

[5] Risald, A. E. Mirino and Suyoto, "Best routes selection using Dijkstra and Floyd-Warshall algorithm," 2017 11th International Conference on Information & Communication Technology and System (ICTS), Surabaya, 2017, pp. 155-158.

[6] Esuabana, Ita. "Shortest Transportation Route Network in Nigeria Using Floyd- Warshall's Algorithm," Mathematical Theory and Modeling, 5. 145-160, 2018.

[7] Y. S. Triana and I. Syahputri, "Implementation Floyd-Warshall Algorithm for the Shortest Path of Garage," International Journal of Innovative Science and Research Technology, Volume 3, Issue 2, February 2018.

[8] Ramadiani, D Bukhori, Azainil and N Dengen, "Floyd-warshall algorithm to determine the shortest path based on android," International**.**

[9] Z. Ramadhan, A. P. U. Siahaan, M. Mesran, "Prim and Floyd-Warshall Comparative Algorithms in Shortest Path Problem," 2018 The 1st International Conference On Advance & Scientific Innovation (ICASI), Medan, 2018, pp. 47-58.

[10] Wang B., Yao S., Lu K., Zhao H, "Research on the Shortest Path Problem Based on Improved Genetic Algorithm," 2013 International Journal of Application or Innovation in Engineering & Management (IJAIEM), 2013, pp. 316-320.