Informatic Engineering / 2017 / Delvine Natalie Caroline / 1103130123

Design and Realization Group NN Queries on Load Distance Method to Choose Location of Warehouse

Delvine Natalie Caroline

Department of Informatic Engineering
Telkom University
Bandung, Indonesia
delvinetoleng@gmail.com

Abstract

Some company has strategy in the operational sector that to choose a location for the warehouse's company, especiallly for companies which give priority to distribution to their customers. Customer in here what it meant is a company that supplies product to them. Selection for path distribution warehouse is considered by distance from the warehouse to the customer, then the path distribution of warehouse was oriented by distribution. In the fact, there is some company that distributed their product to many customers. Usually, this company will analyze about their location to find out cost for distribution is minimized or not. Cost distribution was included transportation cost, customer service cost, and warehouse operational cost.

In this final project, writer will bring about the method from spatial science into management science which is the method use to find out the distance from each warehouse to the customer by a system. This method, called NN Queries. But in this case, the customer has many branchs. So, in this final project, writer proposes Group NN Queries to solve this problem to find out the distance from each customer. In management science, there is a method that can use to make a decision for location warehouse from candidates location of warehouse. This method called Load Distance Method. This method considers to distance from each warehouse to customer and how much the warehouse can load the product to each customer. The system will find out which the path distribution was optimum.

With this system created, hopefully can help operational activity to get the location of warehouse easier and optimum.

Keyword : Group NN Queries, NN Queries, Load Distance Methode, cost distribution

1. Introduction

Some company has strategy in the operational sector that to choose a location for the warehouse's company, expecially for companies which one give priority to distribution to their customers. There are so many to consider objectively to choose location of the warehouse's company. The purpose of location strategy is related to decision how to organize capacity. Location of company demand to commit their resources for the long term. With this fact, location and distribution become more important for base strategy to access markets. Criteria to choose location is to maximize profit and minimize distribution cost. If location of the factory was defined by their resources, but differ with location of warehouse which is oriented with the distribution. Warehouse location, consider the distance from warehouse to each customer. In the fact, there are some company that distributed their product to several customer. Escpecially for this company, warehouse location must be the first priority.

In many situations, criteria that be used to choose which location that can coverage one of objectivity can be quantified, example cost, time, or distance. There is a lot of technique in operational management that used to identify location which is the location was optimum with demand calculation moderate. Almost all of allocation model, the target was to minimize the sum of cost that can give huge effect. Effort the company in section location and distribution, concentrate to cost distribution that it must be constant. Characteristic in location warehouse problem is how making cost distribution minimize. Cost distribution was included transportation cost, customer service cost, and warehouse operational cost. In every analysis about location company, the location that become a candidate for warehouse must be identified[7]. There is some factor to choose warehouse location that related with distance is the average distance to customer, distance with their supplier, and distance with their opponent company.

With this background, there is a method can handle this problem. It is called Load Distance Method. Load Distance Method is mathematic model that use to evaluation location company. Objectivity for this method is to choose the path distribution which has a minimum load product and cover a distance. To calculate the value of load distance, system needs how much candidate warehouse can load the product to their customer and distance between the warehouse and the customer. But in this method, the distance between the candidate warehouse and customer is just use Euclidean Distance. In the fact, the company will calculate how far they must to distribute their product. Because of that, system will apply Group NN queries to find the distance. This query will find out the shortest path from warehouse to customer. But in the fact many companies that have various customers, the customer is a company that has many branches. Based of that fact, the writer use Group NN Queries to find out the distance each customer to the warehouse. But in this final project writer will calculate the value from customers side.

2. Related – Work

2. 1. Load Distance Method

In every facility location, attractive candidate locations must be identified and compared on the basis of quantitative factors. The load distance method is one way to facilitate this step. Several location factors related directly to distance: proximity to markets, average distance to target customers, proximity to suppliers and resources, and proximity to other company facilities. The load distance method is a mathematical model used to evaluate locations based on proximity factors. This approach assumes that theres is only one facility to be located, it must serve a predetermined set of nodes (customer, supplier)[7]. In a logistic network, and it is independent of any other facility that may be in the network[7]. The objective is to select a location that minimizes the sum of the loads from the facility to each node, multiplied by the distance the load travels.

To calculate the ld score for any potential location, the system could ues the actual distance between any two points using geographical information system and simply multiply the loads flowing to and from the facility by the distances traveled. To find the lowest load-distance score with this approach would involve a lot of trial and error as each propective location would have to be evaluated. Alternately, rectilinear or Euclidean measures can also be used as an approximation for distance using the x and y coordinates for each node in the network. The use of coordinates on a two-dimensional graph, in conjuction with a mathematical model, can be helpful in finding a good starting point for a final location. Travel time, actual miles, or Euclidean or rectilinear distances when using a graph approach, are all appropriate measures for distance.

$$ld = \sum_i l_i d_i$$

Where

l_i $= load\ travelling\ between\ location\ i\ and\ the\ proposed\ warehouse$

$d_i = distance\ between\ location\ i\ and\ the\ proposed\ warehouse$

To implementation Group NN Queries in this method, writer use Group Closest Algorithm (GCP) to choose which one the most optimum warehouse for the customer. To find distance between each customer and warehouse, writer use Djikstra Algorithm to calculate the distance.

2. 1. 1. Group Closest Algorithm

Assume an incremental CP algorithm that outputs closest pairs <p,q> in ascending order of their distance. Consider that keep the count(p) of pairs in which p has appeared, as well as, the accumulated distance. When the count of p equals the cardinality n of Q, the global distance of p, with respect to all query points, has been computed. In general, the list of qualifying points keeps increasing until a complete NN is found. Then, non qualifying points can be gradually removed from the list based on the following heuristic.

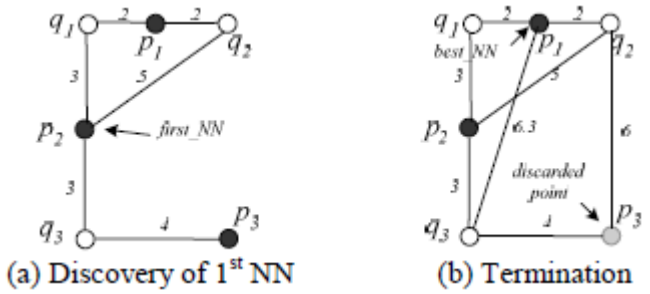

Figure 2. 1 Example of GCP

In this figure, for instance, treshold T=t=7, meaning that when the output pair has distance > 7, the algorithm can terminate. GCP terminates when (i) at least a GNN has been found and (ii) the qualiflying list is empty, or the distance of the current pair becomes larger than the global treshold.

```
GCP
best_NN = NULL; best_dist = ∞; /* initialization
repeat
 output next closest pair <p_i,q_j> and dist(p_i,q_j)
  if p_i is not in list
    if best_dist < ∞ continue; /* discard p_i and process next pair
    else add < p_i, 1, dist(p_i,q_j)> in list;
 else /* p_i has been encountered before and still resides in list
     counter(p_i)++; curr_dist(p_i)= curr_dist(p_i)+ dist(p_i,q_j);
     if counter(p_i)= n
        if curr_dist(p_i)< best_dist
           best_NN = p_i;  //Update current GNN
           best_dist = curr_dist(p_i);  T=0;
           for each candidate point p in list
              if (n-counter(p))· dist(p_i,q_j)+curr_dist(p) ≥ best_dist
                   remove p from list; /* pruned by heuristic 6
              else /* p not pruned by heuristic 6
                   t= (best_dist-curr_dist(p)) / (n-counter(p));
                   if t > T then T = t; /* update threshold
        else remove p_i from list;
     else /* counter(p_i)< n
       if best_dist < ∞ /* a NN has been found already
          if (n-counter(p_i))· dist(p_i,q_j)+curr_dist(p_i) ≥ best_dist
              remove p_i from list; /* pruned by heuristic 6
          else /*not pruned by heuristic 6
              t_i= (best_dist-curr_dist(p_i)) / (n-counter(p_i));
              if t_i > T then T = t_i; /* update threshold
until (best_dist < ∞) and (dist(p_i,q_j) ≥ T or list is empty);
return best_NN;
```

Figure 2. 2 GCP Algorithm

2. 1. 2. Djikstra Algorithm

Djikstra algorithm had been found by Edsger Wybe Djikstra in 1959[9]. This algorithm can solve about finding shortest path in graph that had value of edge is positive[9]. Djikstra algorithm is equal with Greedy algorithm. Greedy algorithm is algorithm that solve problem with metaheursitic to make optimal choose in each step with high hope that get the global optimum[10]. Approach to Greedy algorithm is visit each nodes, from tge start node[1]. And

then this algorithm choose which the closest node and do it repeatedly then calculate the cost each edges that had been through to target node.

Algorithm 1 Djikstra ALgorithm[1]

```
1: DJIKSTRA(G,s,w)
2: for eachvertex u in V
3:     d[u] = infinity
4:     p[u] = u
5:     color[u] = WHITE
6: end for
7: color[s] = GRAY
8: d[s] = 0
9: INSERT(Q,s)
10: while Q!=null
11:     u = EXTRACT-MIN(Q)
12:     S = S U u
13:     for each vertex v in Adj[u]
14:         if w(u,v) + d[u] ¡ d[v]
15:             d[v] = w(u,v) + d[u]
16:             p[v] = u
17:         if (color[v] = WHITE)
18:             color[v] = GRAY
19:             INSERT(Q,v)
20:         else if (color[v] = GRAY)
21:             DECREASE-KEY(Q,v)
22:         else
23:             ...
24:     end for
25:     color[u] = BLACK
26: end while
27: return (d,p)
```

In line 9 do searching the vertex s, and then in line 11 do checking to vertex u[1]. And then in line 13, do checking each point (u,v)[1]. In line 19, do searching on vertex v[1]. And in the end in line 24, finishing vertex u[1].

2. 2. Road Network Model

A key enabling technology in developing traffic management strategies are accurate traffic models that can be easily used for both prediction and control[4]. Road network can be regarded as a graph which is composed of lines and points[6]. In a national system, cross-city path searches which are often carried out by using urban road network and national highway network, synthetically. For example,to search path from Westlake of Hangzhou to Oriental Pearl of Shanghai, we first neet to take advantage of the urban road network looking for the best highway entrance path for Shanghai, and then look for a high- speed roat to Shanghai, and lastly fint the city's path from highway intersection to Oriental Pearl in the use of urban road network. However, in cintrast to dense distributed road network, rural road network and highway network are manifested in a larger range of road space, with lower density, different spatial scales and corresponding traffic rules (such a moving object does not change the direction of movement for a long time). This requires to build a multi-levels road network model to support cross-city search of roads which could switch at different levels of detail or at different levels of road network[6].

The analysis of road network operation schemes often employs modelling[3]. There are several modelling techniques that can be used. It is important that the appropriate technique is selected based on technical and practical grounds. It has been recognised that sometimes the right modelling tool has not been employed for a specific application. The objective of this study is to develop guidelines for the selection of an appropriate modelling technique. This was approached at two levels. General guidelines that are applicable to a wide range of road network operation schemes were first developed. The general guidelines were then refined for a specific

application. Signalised intersection studies constitute a major share of work of road/transport agencies.

A road network is defined as a two-tupleRN = (G, coE), where G is a directed, labeled graph and coE is a binary, so-called co-edge, relationship on edges[12]. The graphG is itself a two-tuple (V,E), where V is a set of vertices and E is a set of edges[12]. Vertices model intersections and starts and ends of roads[12].

Figure 2. 3 Road Network Model

In figure [2.3], road network model related to graph. It means, road network model is made from points or node that define data for the application. To build a road network model for Bandung, the application needs dataset for each intersection and sharp turn in Bandung. After had the dataset, the application connect each point to another point that related to the point until had build a graph.

2. 3. NN Query

The efficient implementation of Nearest Neighbor (NN) queries is of a particular interest in Geographic Information System (GIS[8]. Fo example, a user may point to a specific location or an object in the screen, and request the system to find the five nearest objects. Another situation where NN query, is useful is when the user is not familiar with the layout of the spatial objects. Another even more complex query that could be handled by an NN techniques is to find the four nearest stars which are at least ten light-year away.

Nearest neighbor (NN) search is one of the oldest problems in computer science. Several algorithms and theoretical performance bounds have been devised for exact and approximate processing in main memory[2]. Furthermore, the application of NN search to content-based and similarity retrieval has led to the development of numerous cost models and indexing techniques for high dimensional versions of the problem[2]. Nearest neighbor searching is an important problem in a variety of applications, including knowledge discovery and data mining [Fayyad et al. 1996] pattern recognition and classification [Cover and Hart 1967; Duda and Hart 1973], machine learning [Cost and Salzberg 1993], data compression [Gersho and Gray 1991], multimedia databases [Flickner et al. 1995], document retrieval [Deerwester et al. 1990], and statistics [Devroye and Wagner 1982[11]. Formally, the nearest-neighbor (NN) search problem is defined as follows: given a set S of points in a space M and a query point q ∈ M, find the closest point in S to q. Few of these methods can be used in NN Queries.

2. 4. Group NN Query

Group nearest neighbor (GNN) queries, a novel form of NN search. The input of the problem consists of a set of static data points in multidimensional space.

$$P = p_1, \dots\dots, p_n$$

And a group of query points.

$$Q = q_1, \dots\dots, q_n$$

The output contains the k 1 data point(s) with the smallest sum of distances to all points in Q. The distance between a data point p and Q is defined as dist with this formula.

$$(P, Q) = \sum_{i}^{n} 1 - n|pq_i|$$

where =

pQi : the Euclidean distance between p and query point qi.

In addition to its relevance in geographic information systems and mobile computing applications, GNN search is important in several other domains. For instance, in clustering and outlier detection, the quality of a solution can be evaluated by the distances between the points and their nearest cluster centroid. Furthermore, the operability and speed of very large circuits depends on the relative distance between the various components in them. GNN can be applied to detect abnormalities and guide relocation of components. For another example implementation of GNN is given two sets of points P and Q, a group nearest neighbor (GNN) query retrieves the point(s) of P with the smallest sum of distances to all points in Q. Consider, for instance, three users at locations q/sub 1/ q/sub 2/ and q/sub 3/ that want to find a meeting point (e.g., a restaurant); the corresponding query returns the data point p that minimizes the sum of Euclidean distances\cite{GNN1}. In this final project, the writer will use GNN search to find every distance between warehouse and each customer.

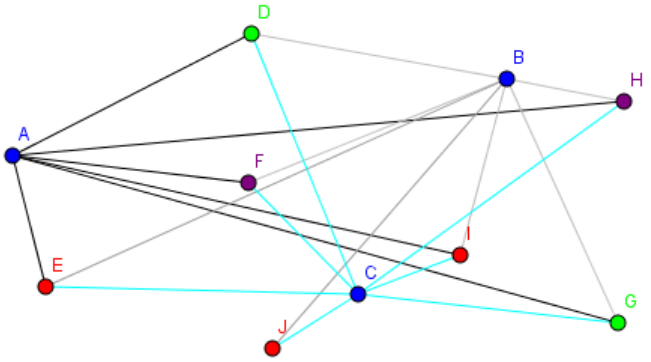

Figure 2. 4 Group NN Simulation

In figure [2.4], for each customers has to calculate the distance with each warehouse. In this figure, GNN is use Euclidean Distance which means take the line from customers to warehouses. After each customer had distance value, then will be comparing for each distance to warehouse which one the optimum location of warehouse.

2. 5. Graph

Defining of graph is pair for association (V,E), that been write with notation G = (V,E), that V as association not null from nodes and E which as association of side that connect each node into another node[5]. In geometry, graph explain an association of nodes *Dwimatra* area that had been connected by lines.

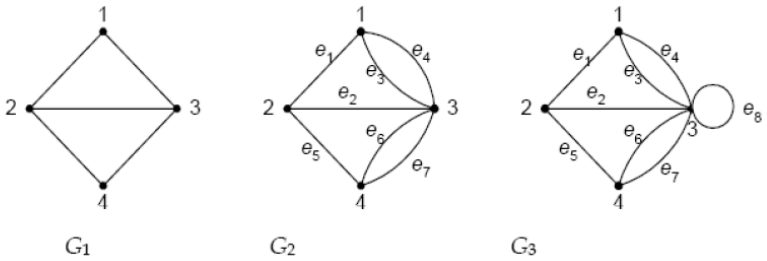

Figure 2. 5 Graph

In the figure [2.5] there are three graph with different edges and nodes. G1 had nodes and edges. In G2, edge e3 = (1,3) and e4 = (1,3) called multiple edges or parallel edges[5]. in G3, edge e8 = (3,3) called loop because it begin and end to the same node.

3. System Design

3. 1. System Description

In this final project, we will design an application to choose which one the most optimal warehouse for distribution path to the company. The application will calculate distance from each customer and how much product for each customer. The user from this application will need to input which one the user that want to be calculated.

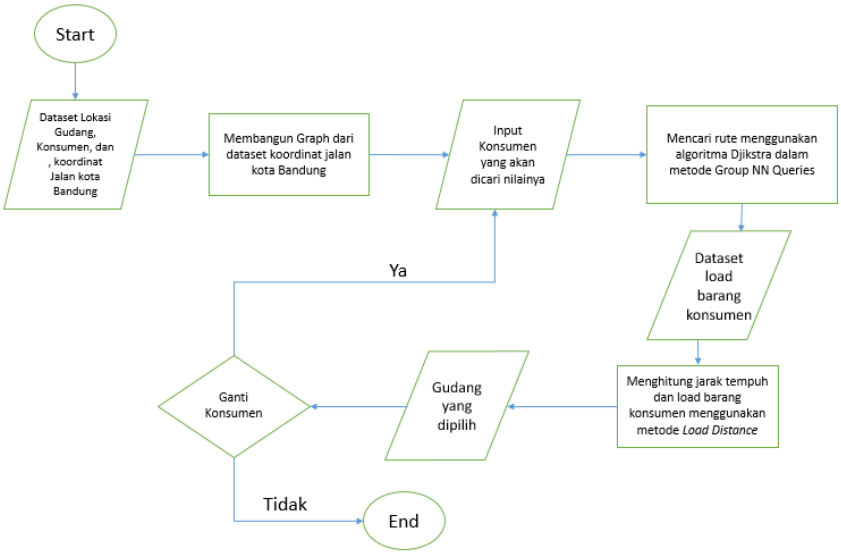

Figure 2. 6 Flowchart Modelling System

3. 2. System Architecture

3. 2. 1. Manufacture Dataset in Table

In this state, the application will use dataset about location of the warehouse, location for the customers, and data about road of kota Bandung. Dataset for location warehouse and customers is fill with lattitude and longitude from each position.

ID	Warehouse	Lattitude	Longitude
W-1	Gudang1	-6.884457	107.600801
W-2	Gudang2	-6.904146	107.620479
W-3	Gudang3	-6.926530	107.597587

Tabel 3.1: Dataset for location warehouses

In table [3.1], the dataset is for location of warehouses. In this application, there are three

warehouses which is "Gudang1", "Gudang2", and "Gudang3". This table is fill with lattitude and longitude from each warehouses.

ID	Customer	Lattitude	Longitude
C-1	K24-1	-6.873257	107.552399
C-2	K24-2	-6.954672	107.611685
C-3	K24-3	-6.954429	107.639564
..	.....	.....	.....

Tabel 3.2: Example Dataset for Location Customers K24

Same as table [3.1], table [3.2] is for location of each customers. In this application, there are three type of customers which is "K24", "Kimia Farma", and "Kita Jaya". The customer has each branch. Then, this table is fill with each lattitude and longitude from each branch of customer. Dataset for road Bandung city was fill with lattitude and longitude from each road. This dataset will be read by application and make a graph that can be used for looking the most optimal path.

ID	Node	Lattitude	Longitude
N-1	Kopo-1	-6.960372	107.579688
N-2	Kopo-2	-6.945566	107.589726
N-3	Cibaduyut-1	-6.962401	107.59357
..	.....	.....	.....

Tabel 3.3: Dataset for Road Network Model

In table [3.3], this table is for point to make a Node. This table will explanation which one the road that this application used for Road Network Model. In this table, fill with lattitude and longitude from each point. Point in this tables definite intersection and sharp turn from Road Network Model of Bandung. This application also need dataset how many product that distribute to each customer because this application using Load Distance Method.

ID	Customer	LoadProduct
L-1	K24-1	120
L-2	K24-2	70
L-3	KimiaFarma-1	50
..	.....	.....

Tabel 3.4: Dataset for Load Product

From table [3.4], using for initialize how much product that been load for each customer and warehouse. In database, the application will had dataset for load product from each warehouse. Example of dataset of load product is in table 3.4 . This dataset will be used in step how to calculate cost distribution in Load Distance Method.

3. 2. 2. Manufacture Graph from Dataset

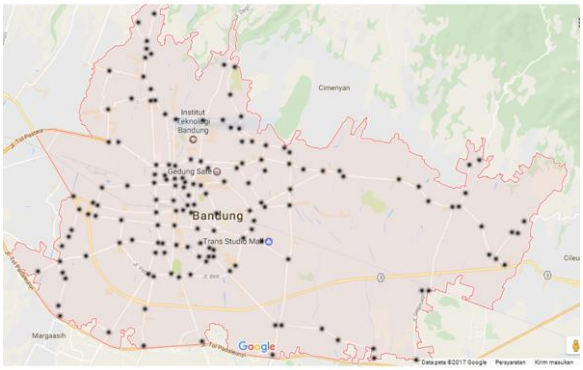

Figure 3. 1 Map Nodes of Bandung

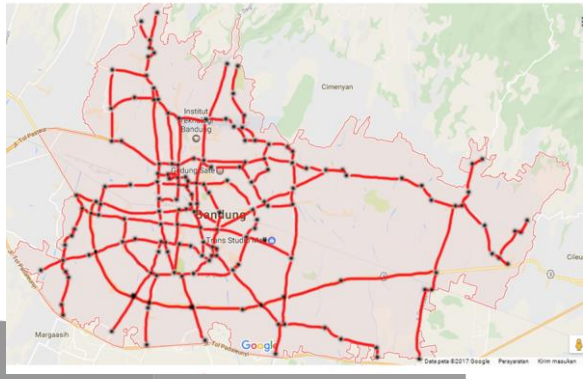

Figure 3. 2 Visualize Graph of Bandung

In this state, graph will have dataset for coordinate each road in Bandung. Before build a graph, application will make nodes from the dataset. In figure [3.3] the graph is already been created. In below is algorithm for add the node and to add edge from each road. In this figure (figure[3.2]), was mapping of intersection and sharp turn in Bandung. This figure is visualize in step that the application read the database to make a graph. From this points, the application can connect for each point that must be connected. The road that application use is road with type of III in Indonesia. The application just read the dataset from table [3.3] and make the point from the dataset. After the application (figure [3.3])made the points from dataset, the application will make a line that connected points which one the point is must be connected with another one of point. The line will be draw by red marks. After the application made the points, the application will add the graph with points for each customer. In figure [3.4], visualize for the grah that had been made from the application.

3. 2. 3. Finding Shortest Path using Djikstra Algorithm

This application using Djikstra Algorithm to find a shortest path from the graph that had been created. But the algorithm will be using Group NN Queries.

Algorithm 4 Djikstra Algorithm[1]

```
1: djikstra()
2: while list Edges available
3:     Initialize : next as cheapest vertex
4:     dist = getDistance(next)
5:     Initialize : list of edge
6:     for each list edge
7:         Initialize : Vertex v as neighbor next
8:         if v already visit
9:             continue
10:        w = distance from v
11:        x = distance to next Vertex + getWeight(e)
12:        if x ¡ w
13:            put(v,next)                    ▷ neighbor of next
14:            put(v,x)                       ▷ add weight to new edge
15:            remove(v)
16:            add(v)
```

Djikstra algorithm is an algorithm to find shortest path by choose the minimize for the distance. This algorithm had been many used for Java coder.

3. 2. 4. Manufacture Load Distance Method

This state was using the dataset how much the company load product to each customer. Which one, the application just multiply the distance using Djikstra Algorithm and how much the company load the product to the customer.

Algorithm 5 GCP and Load Distance Algorithm[1]

```
processGCP(list of warehouse, list of customer)
Initialize : bestVertex as Vertex
bestVertex = null
Initialize: bestDist, currDist as double
Initialize: intGudang as int
Initialize: list of warehouse
for each warehouse
    for each customer
        currDist = currDist + getDist(customer[i],warehouse[j])
  if currDist smaller than bestDist
      bestDist = currDist
      idGudang = j
      for each warehouse in list
          if n-counter(warehouse) getDist(customer[i],warehouse[j]) + currDist more than
          else
              t = bestDist-getDist(customer[i],warehouse[j])/n-counter(warehouse)
              if t more than T then T = t                    ▷ update treshold
  else remove warehouse from list
      currDist = 0
else
  if n-counter(warehouse) getDist(customer[i],warehouse[j]) + currDist more than bestD
      remove warehouse from list
  else
      t = bestDist-getDist(customer[i],warehouse[j])/n-counter(warehouse)
      if t more than T then T = t                    ▷ update treshold
bestVertex = graph.getVertex(warehoue[j])
return warehouse[j]
```

For GNN Queries in this application, writer using Group Closest Pair Algorithm to find the closest pair for each customer. GCP Algorithm applies an incremental CP algorithm that must keep all closest pairs in the heap until the first NN is found.

4. Experimental Studies

4. 1. Data Scenario

In this experiment, the application will represented a graph as in chapter three.

ID	Warehouse	Lattitude	Longitude
W-1	Gudang1	-6.884457	107.600801
W-2	Gudang2	-6.904146	107.620479
W-3	Gudang3	-6.926530	107.597587

Tabel 4.1: Dataset Experiment for location warehouses

In table [4.1], the dataset is for location of warehouses. In this application, there are three warehouses which is "Gudang1", "Gudang2", and "Gudang3". This table is fill with lattitude and longitude from each warehouses.

Same as table [4.1], table [4.2] is for location of each customers. In this application, there are three type of customers which is "K24", "Kimia Farma", and "Kita Jaya". The customer has each branch. Then, this table is fill with each lattitude and longitude from each branch of customer.

ID	Customer	Lattitude	Longitude
C-1	K24-1	-6.944638	107.661001
C-2	K24-2	-6.943946	107.667032
C-3	K24-3	-6.961223	107.666783
C-4	K24-4	-6.913333	107.701951
C-5	K24-5	-6.933479	107.642643
C-6	K24-6	-6.904009	107.666093
C-7	K24-7	-6.905841	107.682839
C-8	KitaJaya-1	-6.927285	107.613074
C-9	KitaJaya-2	-6.944272	107.629573
C-10	KimiaFarma-1	-6.914037	107.698098
C-11	KimiaFarma-2	-6.915969	107.654214
C-12	KimiaFarma-3	-6.942858	107.591320
C-13	KimiaFarma-4	-6.945883	107.631481
C-14	KimiaFarma-5	-6.942255	107.627827
C-15	KimiaFarma-6	-6.938064	107.614696

Tabel 4.2: Dataset Experiment for Location Customers

4. 2. Experiment Scenario

In this experiment too, the application will use many of case of dataset.

4. 2. 1. Customers choose the optimum warehouse.

4. 2. 2. Customers choose the choosen warehouse in the real company.

The experiment will take two condition for a searching the shortest path. It's mean each case have two condition. First one is use the Road Network Model and second is use Euclidean Distance. Every single experiment will tell the answer for each case and for each condition. This experiment will calculate how much value of load distance for each case. The result of this experiment will comparison with another condition.

5. Experiments

5. 1. 1st Case

5. 1. 1. Using Road Network Model

Numb	Warehouse	Customer	Load Product	Distance	Load Distance
1	Gudang1	K24-1	75	471	35325
2	Gudang1	K24-2	34	467	15878
3	Gudang1	K24-3	12	591	7092
4	Gudang1	K24-4	58	743	43094
5	Gudang1	K24-5	90	536	48240
6	Gudang1	K24-6	120	391	46920
7	Gudang1	K24-7	68	721	49028
8	Gudang1	K24-12	100	809	80900
9	Gudang2	K24-1	75	103	7725
10	Gudang2	K24-2	34	99	3366
11	Gudang2	K24-3	12	223	2676
12	Gudang2	K24-4	58	245	14210
13	Gudang2	K24-5	90	404	36360
14	Gudang2	K24-6	120	259	31080
15	Gudang2	K24-7	68	421	28628
16	Gudang2	K24-12	100	362	36200
17	Gudang3	K24-1	75	677	50775
18	Gudang3	K24-2	34	673	22882
19	Gudang3	K24-3	12	797	9564
20	Gudang3	K24-4	58	358	20764
21	Gudang3	K24-5	90	555	49950
22	Gudang3	K24-6	120	700	84000
23	Gudang3	K24-7	68	153	10404
24	Gudang3	K24-12	100	241	24100
	Gudang1			=	326477
	Gudang2			=	160245
	Gudang3			=	272439

Tabel 4.4: Result Experiment K24 1st case using Road Network Model

In table [4.4], writer calculate the result for customer "K24". Value of Load Distance get using data load product and the distance between each customers and warehouses. After the calculation, this case got "Gudang2" as the optimum warehouse because it has smaller of value load distance than another warehouse. The application recommended "Gudang2" for for "K24" customer.

a

Numb	Warehouse	Customer	Load Product	Distance	Load Distance
1	Gudang1	KimiaFarma-1	36	764	27504
2	Gudang1	KimiaFarma-2	75	576	43200
3	Gudang1	KimiaFarma-6	100	68	6800
4	Gudang1	KimiaFarma-7	45	284	12780
5	Gudang1	KimiaFarma-8	34	292	9928
6	Gudang1	KimiaFarma-10	12	211	2532
7	Gudang2	KimiaFarma-1	36	266	9576
8	Gudang2	KimiaFarma-2	75	444	33300
9	Gudang2	KimiaFarma-6	100	464	46400
10	Gudang2	KimiaFarma-7	45	246	11070
11	Gudang2	KimiaFarma-8	34	273	9282
12	Gudang2	KimiaFarma-10	12	354	4248
13	Gudang3	KimiaFarma-1	36	337	12132
14	Gudang3	KimiaFarma-2	75	526	39450
15	Gudang3	KimiaFarma-6	100	609	60900
16	Gudang3	KimiaFarma-7	45	725	32625
17	Gudang3	KimiaFarma-8	34	698	23732
18	Gudang3	KimiaFarma-10	12	617	7404
	Gudang1			=	102744
	Gudang2			=	113876
	Gudang3			=	176243

Tabel 4.5: Result Experiment Kimia Farma 1st Case using Road Network Model

In table [4.5], writer calculate the result for customer "Kimia Farma". Value of Load Distance get using data load product and the distance between each customers and warehouses. After the calculation, this case got "Gudang1" as the optimum warehouse because it has smaller of value load distance than another warehouse. The application recommended "Gudang1" for for "Kimia Farma" customer.

Numb	Warehouse	Customer	Load Product	Distance	Load Distance
1	Gudang1	KitaJaya-1	34	331	11254
2	Gudang1	KitaJaya-2	23	295	6785
3	Gudang2	KitaJaya-1	34	394	13396
4	Gudang2	KitaJaya-2	23	257	5911
5	Gudang3	KitaJaya-1	34	589	20026
6	Gudang3	KitaJaya-2	23	714	16422
	Gudang1			=	18039
	Gudang2			=	19307
	Gudang3			=	36448

Tabel 4.6: Result Experiment Kita Jaya 1st Case using Road Network Model

In table [4.6], writer calculate the result for customer "Kita Jaya". Value of Load Distance get using data load product and the distance between each customers and warehouses. After the calculation, this case got "Gudang1" as the optimum warehouse because it has smaller of value load distance than another warehouse. The application recommended "Gudang1" for for "Kita Jaya" customer.

5. 1. 2. Using Euclidean Distance

Numb	Warehouse	Customer	Load Product	Distance	Load Distance
1	Gudang1	K24-1	75	398	29850
2	Gudang1	K24-2	34	428	14552
3	Gudang1	K24-3	12	444	5328
4	Gudang1	K24-4	58	635	36830
5	Gudang1	K24-5	90	379	34110
6	Gudang1	K24-6	120	311	37320
7	Gudang1	K24-7	68	488	33184
8	Gudang1	K24-12	100	561	56100
9	Gudang2	K24-1	75	90	6750
10	Gudang2	K24-2	34	62	2108
11	Gudang2	K24-3	12	155	1860
12	Gudang2	K24-4	58	202	11716
13	Gudang2	K24-5	90	217	19530
14	Gudang2	K24-6	120	174	20880
15	Gudang2	K24-7	68	224	15232
16	Gudang2	K24-12	100	207	20700
17	Gudang3	K24-1	75	241	18075
18	Gudang3	K24-2	34	252	8568
19	Gudang3	K24-3	12	349	4188
20	Gudang3	K24-4	58	319	18502
21	Gudang3	K24-5	90	43	3870
22	Gudang3	K24-6	120	145	17400
23	Gudang3	K24-7	68	141	9588
24	Gudang3	K24-12	100	224	22400
	Gudang1			=	326477
	Gudang2			=	160245
	Gudang3			=	272439

Tabel 4.8: Result Experiment K24 Case 1 using Euclidean Distance

In table [4.8], writer calculate the result for customer "K24" when using Euclidean Distance. Value of Load Distance get using data load product and the distance between each customers and warehouses. After the calculation, this case got "Gudang2" as the optimum warehouse because it has smaller of value load distance than another warehouse. The application recommended "Gudang2" for for "K24" customer.

Numb	Warehouse	Customer	Load Product	Distance	Load Distance
1	Gudang1	KimiaFarma-1	36	614	22104
2	Gudang1	KimiaFarma-2	75	398	29850
3	Gudang1	KimiaFarma-6	100	47	4700
4	Gudang1	KimiaFarma-7	45	248	11160
5	Gudang1	KimiaFarma-8	34	232	7888
6	Gudang1	KimiaFarma-10	12	167	2004
7	Gudang2	KimiaFarma-1	36	186	6696
8	Gudang2	KimiaFarma-2	75	183	13725
9	Gudang2	KimiaFarma-6	100	434	43400
10	Gudang2	KimiaFarma-7	45	236	10620
11	Gudang2	KimiaFarma-8	34	248	8432
12	Gudang2	KimiaFarma-10	12	313	3756
13	Gudang3	KimiaFarma-1	36	299	10764
14	Gudang3	KimiaFarma-2	75	79	5925
15	Gudang3	KimiaFarma-6	100	322	32200
16	Gudang3	KimiaFarma-7	45	228	10260
17	Gudang3	KimiaFarma-8	34	210	7140
18	Gudang3	KimiaFarma-10	12	216	2592
	Gudang1			=	77706
	Gudang2			=	86629
	Gudang3			=	68881

Tabel 4.9: Result Experiment Kimia Farma 1st Case using Euclidean Distance

In table [4.9], writer calculate the result for customer "Kimia Farma" when using Euclidean Distance. Value of Load Distance get using data load product and the distance between each customers and warehouses. After the calculation, this case got "Gudang3" as the optimum warehouse because it has smaller of value load distance than another warehouse. The application recommended "Gudang3" for for "Kimia Farma" customer.

Numb	Warehouse	Customer	Load Product	Distance	Load Distance
1	Gudang1	KitaJaya-1	34	178	6052
2	Gudang1	KitaJaya-2	23	242	5566
3	Gudang2	KitaJaya-1	34	331	11254
4	Gudang2	KitaJaya-2	23	240	5520
5	Gudang3	KitaJaya-1	34	174	5916
6	Gudang3	KitaJaya-2	23	219	5037
	Gudang1			=	11618
	Gudang2			=	16774
	Gudang3			=	10953

Tabel 4.10: Result Experiment Kita Jaya 1st Case using Euclidean Distance

In table [4.10], writer calculate the result for customer "Kita Jaya" when using Euclidean Distance. Value of Load Distance get using data load product and the distance between each customers and warehouses. After the calculation, this case got "Gudang3" as the optimum warehouse because it has smaller of value load distance than another warehouse. The application recommended “Gudang3” for for “Kita Jaya” customer.

5. 1. 3. Comparison

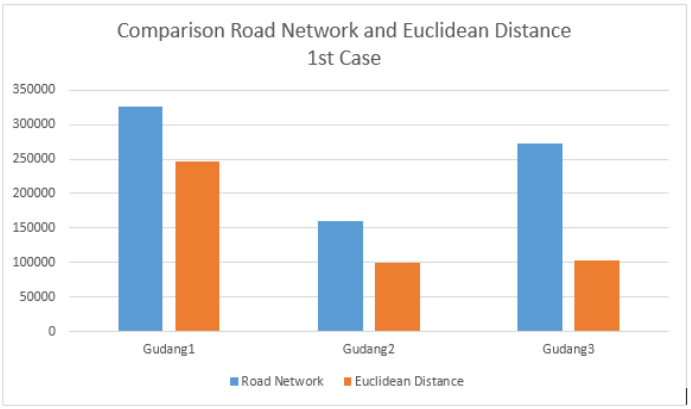

Figure 5. 1 Comparison K24

In figure [5.1], the different between using Euclidean Distance and road network model. If using road network model, the value of load distance much bigger than using euclidean distance. Figure [5.1] define the comparison for customer "K24".

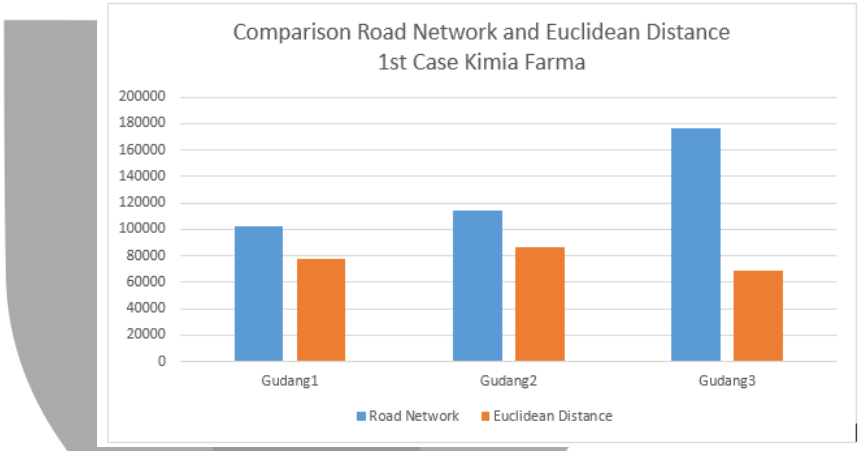

Figure 5. 2 Comparison Kimia Farma

In figure [5.2], the different between using Euclidean Distance and road network model. If using road network model, the value of load distance much bigger than using euclidean distance. Figure [5.2] define the comparison for customer "Kimia Farma".

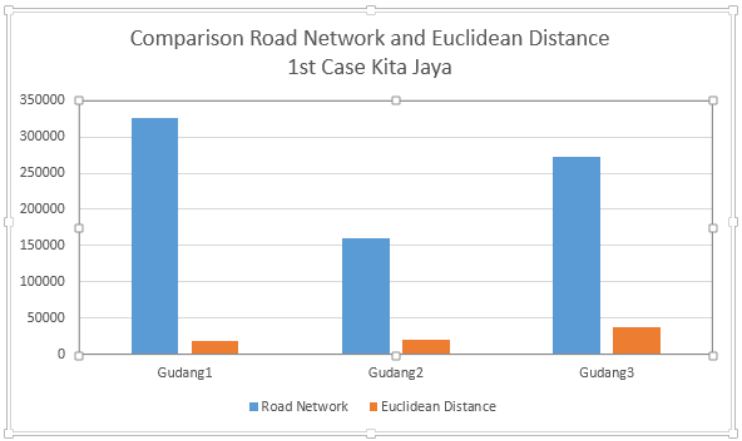

Figure 5. 3 Comparison Kita Jaya

In figure [5.3], the different between using Euclidean Distance and road network model. If using road network model, the value of load distance much bigger than using euclidean distance. Figure [5.3] define the comparison for customer "Kita Jaya".

From this figures of the comparison graph, the different from using Road Network and euclid was in the distance. This things happen because Euclidean Distance just pull a line from start node to target node. That things makes the distance if using Euclidean Distance more small than using Road Network. As least sum of the customer then make the different so highly different. In the real, the distance between each customer and warehouse is not a straight line from customer to the warehouse. In this case, the optimum of location the warehouse is have the minimal from load distance value.

5. 2. 2nd Case

5. 2. 1. Using Road Network Model

Numb	Customer	Warehouse
1	K 24	Gudang3
2	Kimia Farma	Gudang2
3	Kita Jaya	Gudang1

Tabel 4.11: Dataset Real for location warehouses

Numb	Customer	Warehouse	Load Distance
1	K 24	Gudang3	272439
2	Kimia Farma	Gudang2	113876
3	Kita Jaya	Gudang1	18039

Tabel 4.12: Dataset Experiment for location warehouses in 2nd case

In table [4.11], was define result from the company. In this table, each customer had a different result of warehouse. As the result, customer "K24" has "Gudang3" as the most optimum warehouse. Customer "Kimia Farma" has "Gudang2" as the most optimum warehouse. Customer "Kita Jaya" has "Gudang1" as the most optimum warehouse. From table [4.4] , table [4.5], and table [4.6], can be fill the table on this below. The tables is about how much the load distance from each customers. In table [4.12] , is define the path distribution. The choosen warehouse is based on the real company. As the result, customer "K24" has "Gudang3" as the most optimum warehouse with 272439 as Load Distance value. Customer "Kimia Farma" has "Gudang2" as the most optimum warehouse with 113876 as Load Distance value. Customer "Kita Jaya" has "Gudang1" as the most optimum warehouse with 18039 as Load Distance value.

Numb	Customer	Warehouse	Load Distance
1	K 24	Gudang2	160245
2	Kimia Farma	Gudang1	102744
3	Kita Jaya	Gudang1	18039

Tabel 4.13: Dataset Recommendation Experiment for location warehouses in 2nd case

In table [4.13], this table get from table [4.4] , table [4.5], and table [4.6]. This result is based on recommendation from this application. As the result, customer "K24" has "Gudang2" as the most optimum warehouse with 160245 as Load Distance value. Customer "Kimia Farma" has "Gudang1" as the most optimum warehouse with 102744 as Load Distance value. Customer

"Kita Jaya" has "Gudang1" as the most optimum warehouse with 18039 as Load Distance value.

From the table about load distances each customers in real and recommendation, can be represented by graph in figure [4.13]. As can see the difference from K24 customer. The difference is almost the half oh the value. But for Kimia Farma customer and Kita Jaya customer the different is not as big as K24 customer.

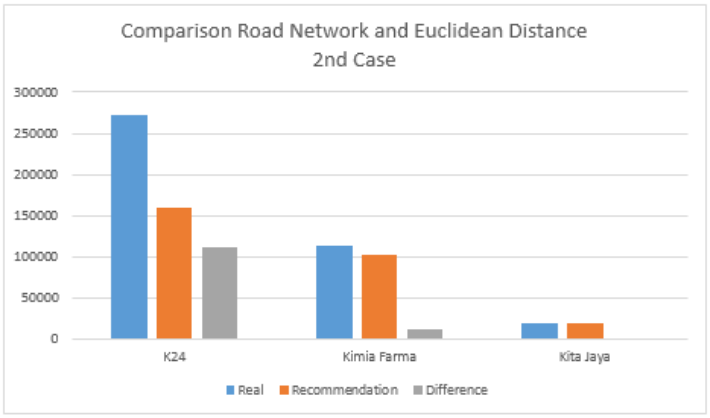

Figure 5. 4 Comparison for 2nd case

6. Conclusion

After doing the experiment and analysis for each case, then the conclusion is.

- Group NN Queries can be implemented on Load Distance Method to calculate the real distance between each customers and warehouses. In Group NN Queries, there's Djikstra algorithm to calculate the distance.
- Influence to use Road Network Model and Euclidean Distance very give impact the result. Because the Euclidean Distance is just calculate the segment between each customers and warehouses.

References

[1] Budiarsyah, D. K. Djikstra algorithm, bellman-ford, and oyd-warshall for finding shortest path from a graph, 2013.
[2] Dimitris Papadias, Qionmao Shen, Y. T. K. M. Group nearest neighbor queries.
[3] Dr. Ian Espada, Dr. James Luk, D. Y. Y. Guidelines for selecting techniques for the modelling of road network operations, 2010.
[4] Emanuele Crisostomi, Steve Kirkland, R. S. A google-like model of road network dynamics and its application to regulation and control, 2010.
[5] Fitria, A. T. Implementation djikstra algorithm to choose shortest path for main road in south sumatra, 2013.
[6] Jun Feng, T. W. Index and Query Method in Road Networks. Springer & KES, 2014.
[7] Lee J. Krajweski, Manoj K. Mallhotra, L. P. R. Operation Management;Processes and Supply Chains : 9th Edition. PEARSON, 2015.
[8] Nick Roussopoulos, Stephen Kellet, F. V. Nearest neighbor queries, 1995.
[9] Priatmoko, S. B. Using djikstra algorithm to find shortest path and recommendation for object interest tourist in bali, 2010.
[10] Setiawan, W. Finding shortest path using djikstra algorithm and a*, 2011.
[11] Sunil Arya, David M. Mount, N. S. N. R. S. A. Y. W. An optimal algorithm for approximate nearest neighbor searching in _xed dimensions, 1998.
[12] Xuegang Huang, Christian S. Jensen, S. S. Multiple k nearest neighbor query processing in spatial network databases, 2006.