PENYELESAIAN *TRAVELING SALESMAN PROBLEM* PADA PERUSAHAAN DISTRIBUSI PRODUK DENGAN ALGORITMA *FARTHEST INSERTION*

Lisnawanty
Program Studi Komputerisasi Akuntansi, AMIK BSI Pontianak
Jalan Abdurahman Saleh No. 18 A, Pontianak
Email: lisnawanty.lsy@bsi.ac.id

ABSTRACT

A Company is an organization that has a complexity of problems in managing the relationship between retailers and distributors to keep the process running smoothly distributing products. The problems that often occur in the distribution company is to determine the best route for delivering products to a number of retailers that will be visited. In the case of Traveling Salesman Problem, this study made some observations on Yakult Company in Pontianak. Based on observations, it is known that the distribution system is done simply. In other words, there is no strategy developed to make the process of distribution, so that in one distribution process not all retailers can be visited. This study discusses the Traveling Salesman Problem solving problems in product distribution company. The method used is the method of heuristic algorithms using Farthest Insertion.

Keywords: *Traveling Salesman Problem, farthest insertion, retailers, distribution*

1. PENDAHULUAN

Perusahaan distribusi produk adalah perusahaan yang berperan sebagai perantara dalam menyalurkan produk dari pabrik ke retailer. Selain menentukan jumlah permintaan produk dari masing-masing perusahaan retailer, permasalahan dalam perusahaan distribusi adalah penentuan rute yang harus dilalui untuk dapat menyalurkan produk secara optimal, sehingga menghindari adanya pemborosan biaya transportasi dan lamanya waktu transportasi karena rute tempuh yang ditentukan tidak optimal. Berdasarkan dari observasi yang dilakukan pada Perusahaan Yakult, maka diketahui bahwa sistem pendistribusian produk yang umumnya digunakan oleh beberapa perusahaan distribusi tersebut adalah *push distribution system* dan pendekatan penjadwalan distribusi yang sederhana. Dalam *push distribution system*, keputusan untuk melakukan *resupply* (pemesanan ulang) berada di tangan retailer. Dalam hal ini, perusahaan distributor harus benar-benar proaktif dalam melakukan penawaran produk mereka kepada retailer. Rute tempuh dalam satu kali proses distribusi masih ditentukan dengan sederhana. Maka dari itu, penelitian ini membahas penyelesaian *Traveling Salesman Problem* (TSP) dengan tujuan memberikan solusi suboptimal pada perusahaan mengenai rute yang harus dilalui dalam melakukan distribusi produk.

2. LANDASAN TEORI

Salah satu algoritma dari pendekatan heuristik (*insertion heuristic*) yang dapat diaplikasikan untuk pencarian rute tour kendaraan yaitu algoritma *farthest insertion*. Algoritma *farthest insertion* dapat menentukan jarak minimal suatu lokasi tour maksimal dari titik-titik (*node*) yang di *input* ke dalam peta secara geografis. Dibandingkan dengan algoritma *insertion heuristic* yang lain, algoritma *farthest insertion* sering digunakan dalam memecahkan permasalahan kombinatorial yang memerlukan waktu komputasi panjang dengan kompleksitas *procedure* pemrograman yang dapat menghasilkan peformansi yang kompetitif dan memberikan solusi dalam memilih rute tour lebih ringkas. Pendekatan *farthest insertion* memberikan solusi yang cukup

baik meskipun solusi *feasible* tidak ditemukan (Saliba, 2007: 89).

3. METODE PENELITIAN

Metode yang digunakan dalam penelitian ini adalah metode heuristik. Algoritma yang digunakan dalam metode heuristik ini adalah algoritma *farthest insertion*. Penelitian ini menggunakan teknik pengumpulan data sebagai berikut.

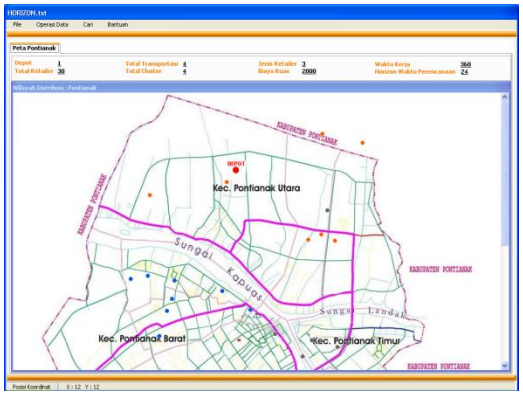
a. Observasi
   Untuk mengetahui permasalahan yang terjadi dalam konsep Traveling Salesman Problem, maka observasi dilakukan dengan melakukan pengamatan pada Perusahaan Yakult di Kota Pontianak untuk mengetahui prosedur distribusi produr yang berjalan pada perusahaan Yakult tersebut.

b. Wawancara
   Wawancara dilakukan dengan mengajukan beberapa pertanyaan sehubungan dengan pendataan permintaan produk oleh retailer serta proses distribusi yang dilakukan oleh Perusahaan Yakult ke masing-masing retailer.

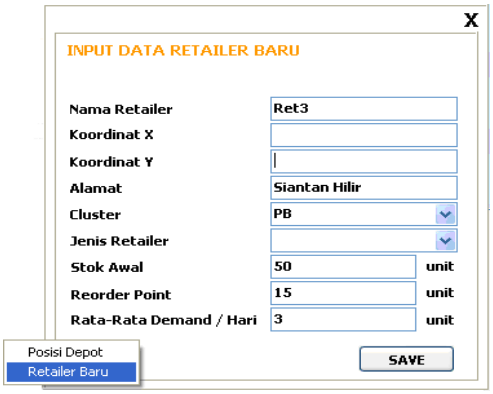
c. Studi literatur
   Studi literatur digunakan dengan mengumpulkan beberapa referensi yang bersumber dari buku, jurnal, maupun referensi lainnya yang berkaitan dalam penelitian ini.

4. PEMBAHASAN

Pada aplikasi ini, data utama yang harus dimiliki adalah data depot (perusahaan distribusi) dan data retailer dengan posisi titik koordinat masing-masing. Gambar 1 berikut ini merupakan layout utama dari aplikasi.



Gambar 1 Layout Utama Aplikasi



Gambar 2 Form Input Data Retailer

Tahap selanjutnya setelah mendata koordinat depot dan sejumlah retailer adalah memperhitungkan jarak antara retailer asal dan retailer tujuan.

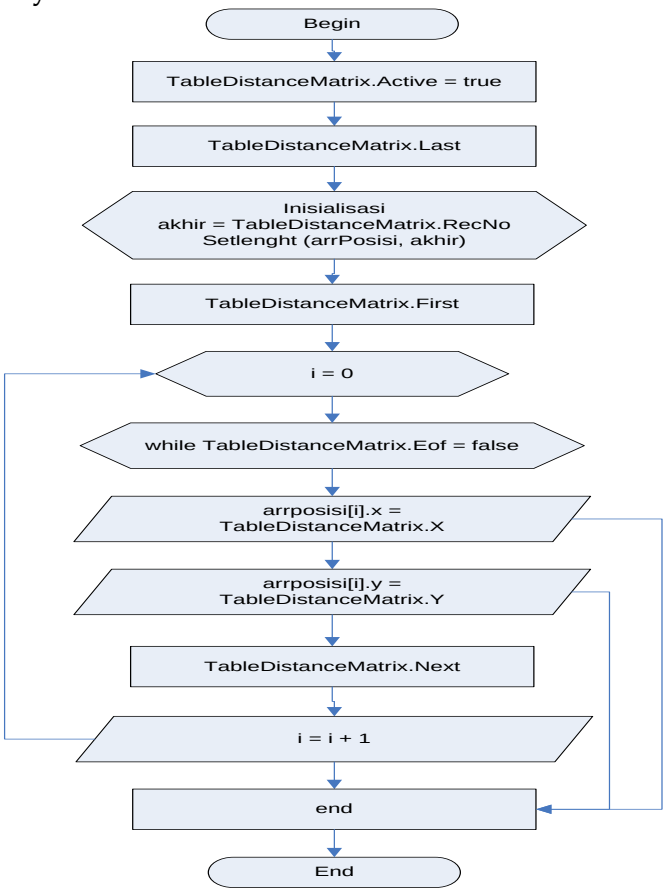
Perhitungan Jarak

Implementasi perhitungan jarak dibuat dalam matrik pada stringgrid. Jarak yang diperhitungkan hanya depot dan jarak antar retailer yang yang akan dituju. Jarak antar retailer diperhitungkan dengan mengetahui posisi koordinat X dan Y retailer asal dan retailer tujuan. Misalkan diketahui ada 19 retailer yang akan dikunjungi sebagai berikut.

Ret 1	Ret 7	Ret 17	Ret 24
Ret 3	Ret 9	Ret 19	Ret 27
Ret 4	Ret 12	Ret 20	Ret 29
Ret 5	Ret 15	Ret 22	Ret 30
Ret 6	Ret 16	Ret 23	

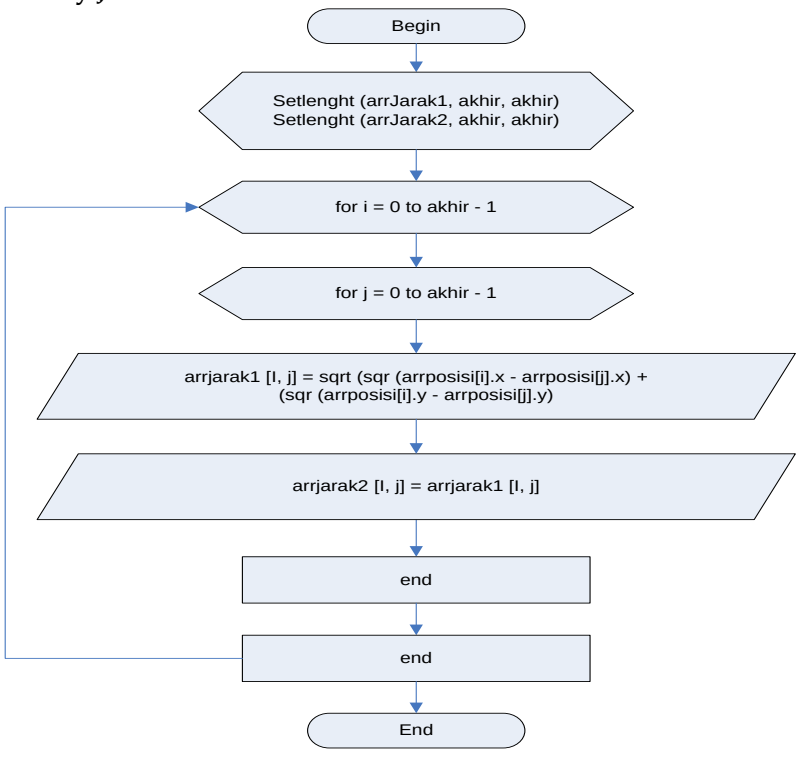
Perhitungan jarak tiap ruas rute ditentukan dengan tiga prosedur sebagai berikut.

1. **Prosedur Isi Array Posisi**



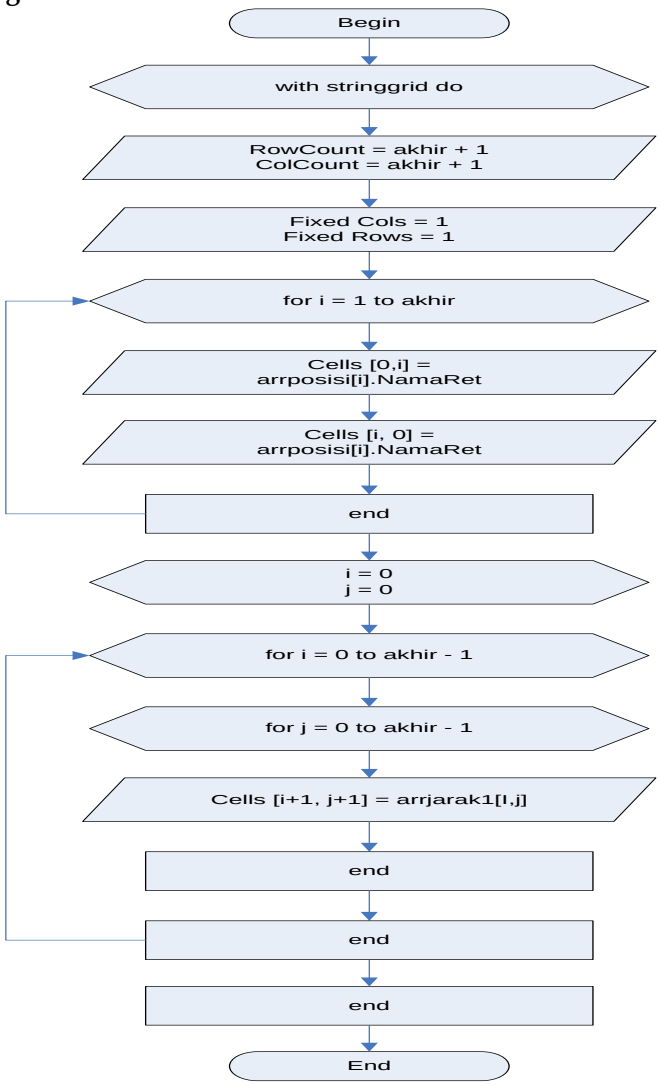
Gambar 3 Prosedur Isi Array Posisi

2. **Prosedur Isi Array Jarak**



Gambar 4 Prosedur Isi Array Jarak

3. Prosedur Isi StringGrid



Gambar 5 Prosedur Isi StringGrid

Adapun hasil perhitungan jarak antar retailer tersebut dapat terlihat pada Tabel 1 berikut ini.

Tabel 1
Perhitungan Jarak Antar Retailer

Retailer Asal - Tujuan	Koordinat (x,y)	Jarak (jarak² = x² + y²)
Ret1 – Ret3	(358,352) - (192,346)	34.53
Ret1 – Ret4	(358,352) - (245,355)	87.05
Ret1 – Ret5	(358,352) - (240,390)	90.38
Ret1 – Ret6	(358,352) - (230,413)	94.37
Ret1 – Ret7	(358,352) - (340,375)	183.45
Ret1 – Ret9	(358,352) - (355,530)	265.51
Ret1 – Ret12	(358,352) - (225,600)	256.89
Ret1 – Ret15	(358,352) - (196,192)	164.45
Ret1 – Ret16	(358,352) - (619,92)	529.26
Ret1 – Ret17	(358,352) - (539,75)	471.05
Ret1 – Ret19	(358,352) – (349,167)	265.91
Ret1 – Ret20	(358,352) – (564,279)	412.51
Ret1 – Ret22	(358,352) –	575.84

	(553,771)	
Ret1 – Ret23	(358,352) – (549,221)	412.36
Ret1 – Ret24	(358,352) – (549,804)	597.65
Ret1 – Ret27	(358,352) – (445,457)	305.6
Ret1 – Ret29	(358,352) – (600,600)	506.82
Ret1 – Ret30	(358,352) – (500,500)	372.65
Ret3 – Ret1	(192,346) -	34.53

	(358,352)	
Ret3 – Ret4	(192,346) - (245,355)	53.76
Ret3 – Ret5	(192,346) - (240,390)	65.12
Ret3 – Ret6	(192,346) - (230,413)	77.03
dst.	dst.	dst.

Gambar 6 berikut merupakan stringgrid yang menampilkan hasil perhitungan jarak sebagaimana yang tampil pada sistem yang dibangun.

	Depot	Ret3	Ret4	Ret5	Ret6	Ret7	Ret8	Ret9	Ret10	Ret11	Ret12	Ret14	Ret15	Ret1
Depot	0	317.72	288.14	319.1	343.68	318.68	513.11	406.08	282.49	512.08	499.8	472.85	237.24	197.
Ret3	317.72	0	53.76	65.12	77.03	169.81	287.54	245.81	214.11	244.13	261.7	483.01	154.05	496.
Ret4	288.14	53.76	0	35.36	59.91	121.83	267.8	206.7	161.05	239.27	245.2	431.71	170.21	457.
Ret5	319.1	65.12	35.36	0	25.08	106.45	233.79	181.18	157.41	203.96	210.54	420.98	202.83	482.
Ret6	343.68	77.03	59.91	25.08	0	108.47	213.03	171.21	166.01	179.52	188.66	421.71	223.6	504.
Ret7	318.68	169.81	121.83	106.45	108.47	0	194.49	95.9	65	205.46	182.91	314.62	281.31	445.
Ret8	513.11	287.54	267.8	233.79	213.03	194.49	0	120.51	241.87	82.08	29.7	355.62	436.4	630.
Ret9	406.08	245.81	206.7	181.18	171.21	95.9	120.51	0	125.87	166.21	122.07	272.44	373.53	511.
Ret10	282.49	214.11	161.05	157.41	166.01	65	241.87	125.87	0	265.44	235.8	272.96	296.58	389.
Ret11	512.08	244.13	239.27	203.96	179.52	205.46	82.08	166.21	265.44	0	55.9	427.2	398.02	650.
Ret12	499.8	261.7	245.2	210.54	188.66	182.91	29.7	122.07	235.8	55.9	0	373.37	412.24	624.
Ret14	472.85	483.01	431.71	420.98	421.71	314.62	355.62	272.44	272.96	427.2	373.37	0	564.67	466.
Ret15	237.24	154.05	170.21	202.83	223.6	281.31	436.4	373.53	296.58	398.02	412.24	564.67	0	434.
Ret16	197.69	496.83	457.21	482.13	504.34	445.45	630.47	511.41	389.22	650.82	624.95	466.05	434.66	0
Ret17	126.57	440.28	406	434.31	457.96	414.16	606.04	490.8	365.16	616.56	596.89	490.77	362.41	81.7
Ret18	170.31	326.38	277.2	293.62	312.18	236.44	416.61	297.23	176.58	441.24	412.15	303.62	326.27	214.
Ret19	84.53	238.1	214.85	248.21	273.27	269.31	459.83	363.05	248.49	448.48	443.09	479.18	155.03	280.
Ret20	203.77	377.99	327.93	342.49	359.88	276.83	447.11	326.62	213.65	478.77	445.56	285.81	378.14	194.
Ret21	176.95	353.93	304.97	321.46	339.95	262.61	439.81	319.83	201.54	466.8	436.36	304.31	349.15	193.
Ret22	653.66	557.63	517.61	493.08	482.18	399.14	315.01	311.91	392.75	396.7	343.58	225.16	680.21	682.
Ret24	685.31	580.7	542.23	516.6	504.62	425.2	328.66	335.73	421.73	409.39	357.84	257.77	706.51	715.
Ret25	382	417.43	364.33	359.2	364.31	255.99	345.38	240.38	203.33	405.64	356.62	90.87	484.29	384.

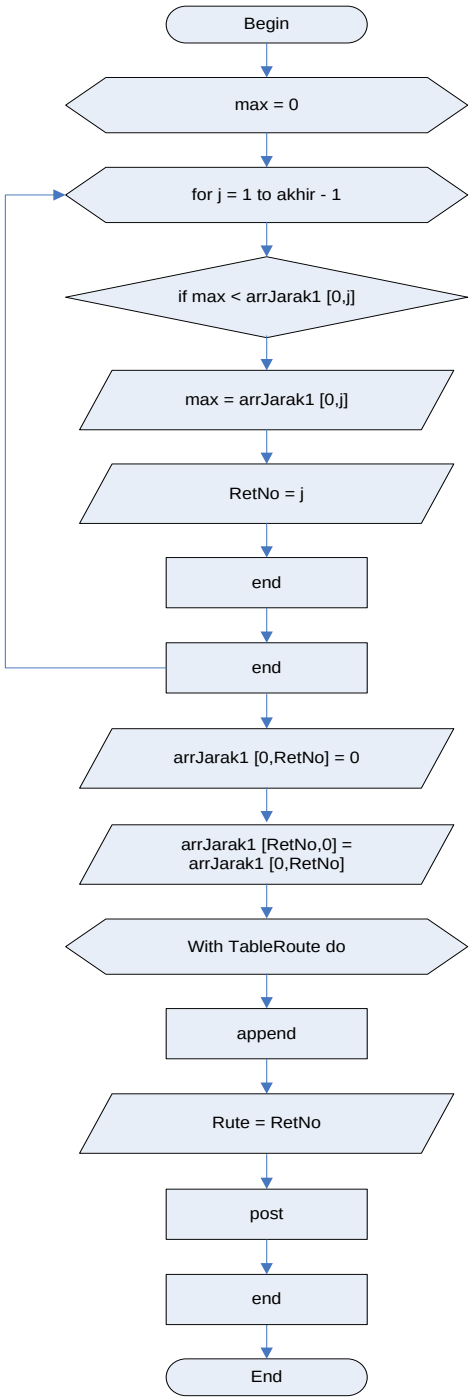
Gambar 6 Stringgrid Hasil Perhitungan Jarak

Penentuan Rute Distribusi

Selanjutnya rute distribusi dapat diperkirakan dengan menggunakan algoritma *farthest insertion* untuk menentukan rute distribusi agar penentuan rute yang akan dilewati untuk proses

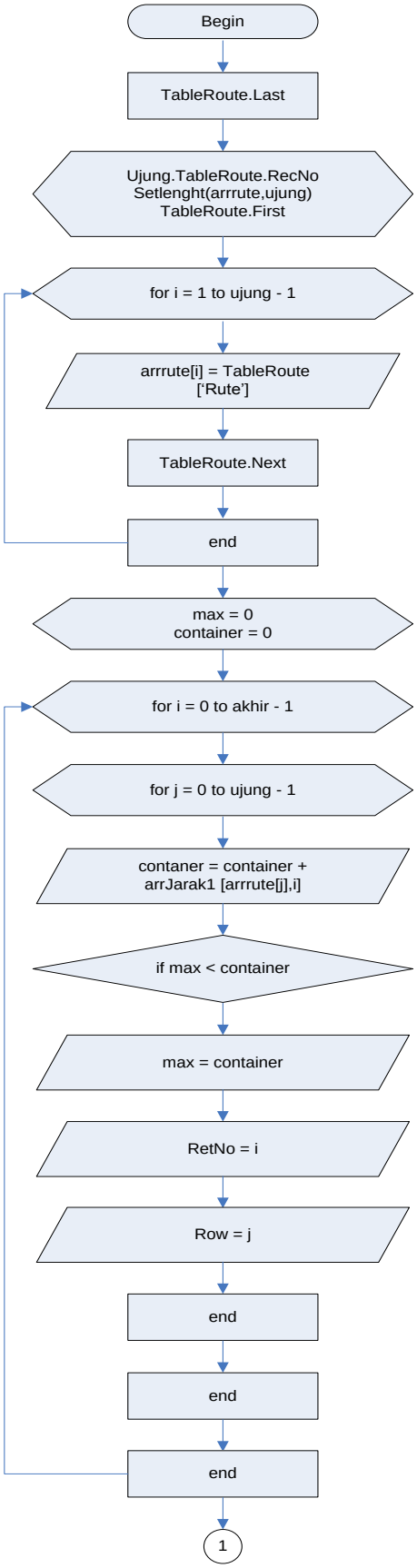
distribusi lebih optimal. Penentuan jalur terpendek yang menggunakan algoritma *farthest insertion* diproses dengan tiga prosedur, yaitu prosedur iterasi pertama, prosedur iterasi kedua, dan prosedur array rute. Berikut ini diuraikan alur dari masing-masing prosedur tersebut.

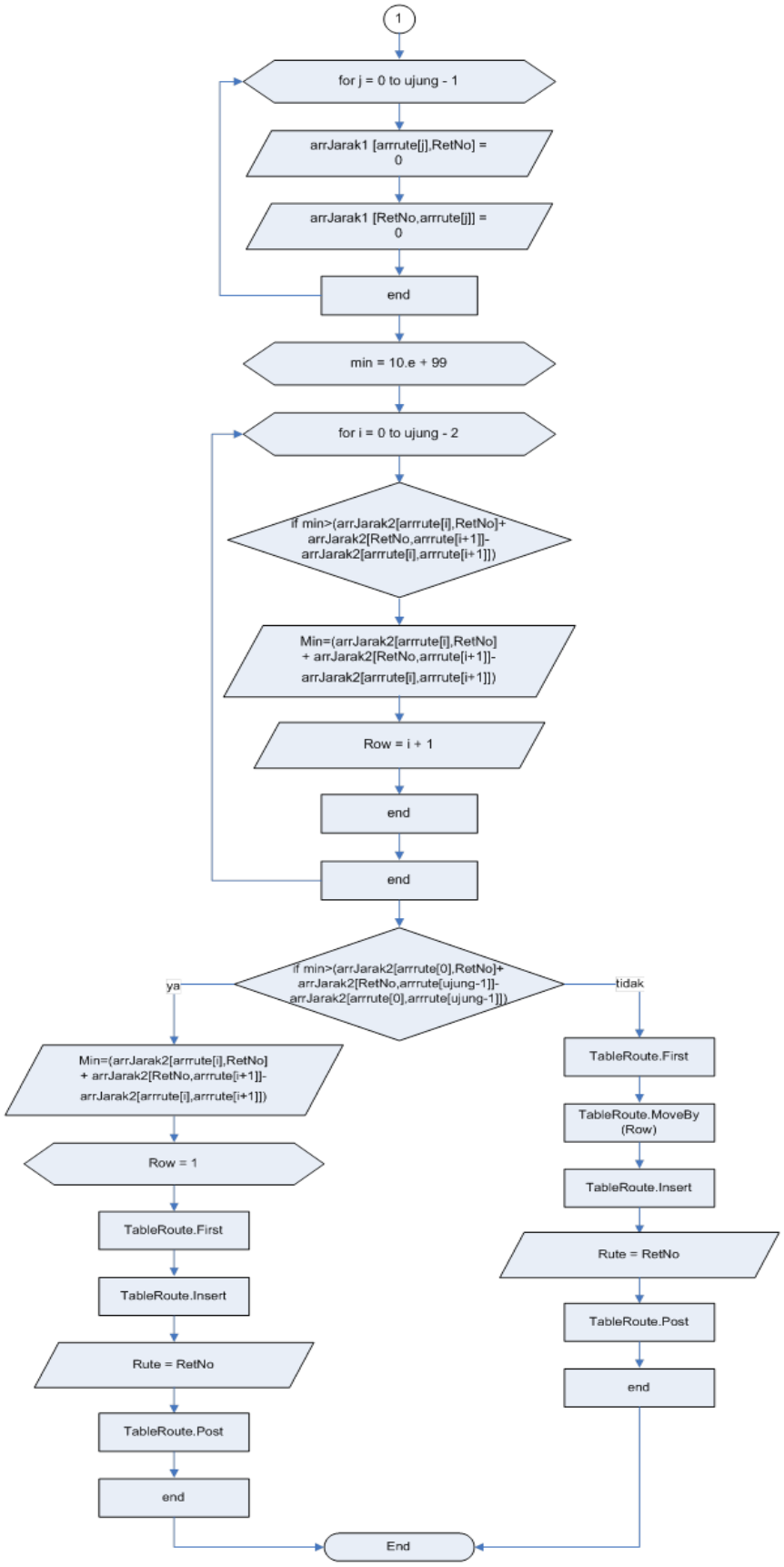
1. Prosedur Iterasi Pertama



Gambar 7 Prosedur Iterasi1

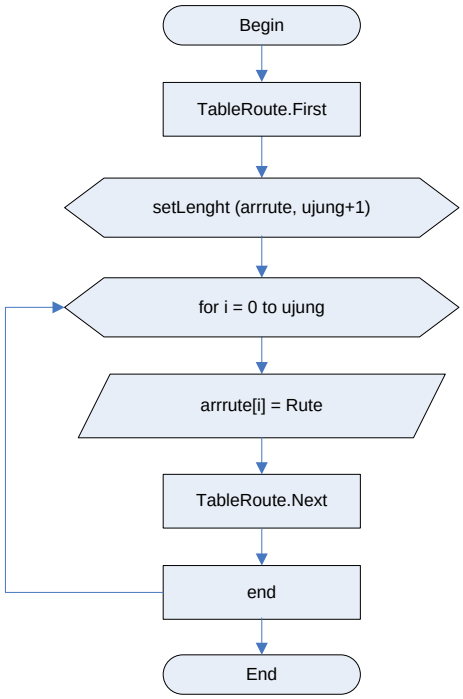
2. Prosedur Iterasi Kedua





Gambar 8 Prosedur Iterasi2

3. Prosedur Array Rute



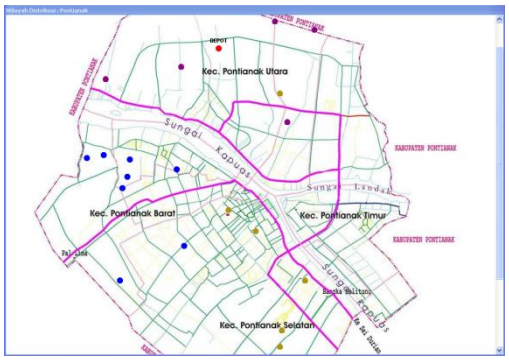
Gambar 9 Prosedur IsiArrayRute

Perhitungan rute optimal dari retailer satu ke retailer lain ini tergantung pada jarak yang antar retailer tersebut. Adapun urutan rute yang dihasilkan adalah sebagai berikut.

1.	Ret22	11.	Ret3
2.	Ret29	12.	Ret7
3.	Ret20	13.	Ret4
4.	Ret23	14.	Ret5
5.	Ret16	15.	Ret6
6.	Ret17	16.	Ret12
7.	Depot	17.	Ret9
8.	Ret19	18.	Ret27
9.	Ret15	19.	Ret30
10.	Ret1	20.	Ret24

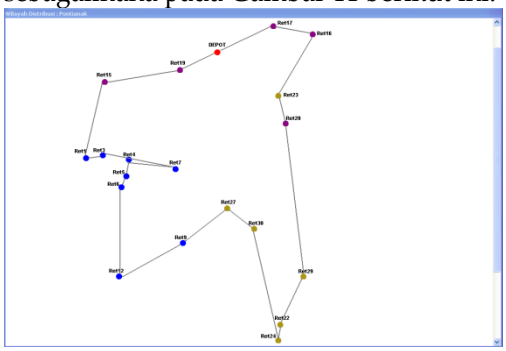
Untuk disusunnya suatu perencanaan penjadwalan, rute dimulai dari depot dan kembali ke depot. Oleh karena itu, urutan rute yang dijadwalkan dalam perencanaan penjadwalan adalah sebagai berikut.

1.	Depot	11.	Ret9
2.	Ret19	12.	Ret27
3.	Ret15	13.	Ret30
4.	Ret1	14.	Ret24
5.	Ret3	15.	Ret22
6.	Ret7	16.	Ret29
7.	Ret4	17.	Ret20
8.	Ret5	18.	Ret23
9.	Ret6	19.	Ret16
10.	Ret12	20.	Ret17



Gambar 10 Posisi Retailer

Jika divisualisasikan dengan garis lurus, maka optimalisasi urutan rute terlihat sebagaimana pada Gambar 11 berikut ini.



Gambar 11 Visualisasi Rute

5. PENUTUP

Berdasarkan hasil penelitian, maka dapat disimpulkan sebagai berikut.

a. Algoritma *Farthest Insertion* dapat menjadi solusi dalam menyelesaikan permasalahan *Traveling Salesman Problem* pada perusahaan distribusi secara optimal.
b. Jalur tempuh dimulai dari depot ke sejumlah retailer dan kembali ke depot.
c. Dengan parameter rata-rata masing-masing waktu pelayanan dan waktu transportasi selama 30 menit (dengan simpangan waktu pelayanan ±20 menit), serta asumsi waktu kerja selama 6 jam, perencanaan

penjadwalan distribusi yang telah dihasilkan menunjukkan bahwa dalam satu hari depot dapat melayani retailer sebanyak ± 6 retailer.

DAFTAR PUSTAKA

Kadarsah, S., Ali R. M. 2000. Sistem Pendukung Keputusan. Bandung: PT. Remaja Rosdakarya.

Hermawan, Julius. 2005. Membangun *Decision Support System*. Yogyakarta: ANDI.

Turban, Efraim, Jay E. Aronson, and Ting-Peng Liang. 2005. *Decision Support Systems and Intelligent Systems*. Yogyakarta: ANDI.