

PENGELOMPOKAN DATA PELANGGAN PDAM SURABAYA DENGAN ALGORITMA *ANT COLONY OPTIMIZATION*

Purbandini¹⁾, Dyah Herawatie²⁾, Rini Semiati³⁾

^{1,2,3)} Prodi Sistem Informasi, Departemen Matematika, Fakultas Sains dan Teknologi, Universitas
Airlangga Surabaya
Kampus C Jl Mulyorejo, Surabaya, 60115
Telp : (031) 5965257, Fax : (031) 5936502
E-mail : purbandini@fst.unair.ac.id¹⁾, dy4h_h3r4@yahoo.com²⁾, semiati_rini@yahoo.com³⁾

Abstrak

Perusahaan Daerah Air Minum (PDAM) Surya Sembada Kota Surabaya dinilai lamban dalam memperluas cakupan jumlah pelanggannya. Ketika pelanggan baru ingin mengajukan pemasangan saluran air, pelanggan menunggu lama realisasinya. Hal ini disebabkan karena data-data lokasi pelanggan baru harus disesuaikan dengan data-data lokasi dari pipa saluran air utama membutuhkan waktu yang lama sehingga waktu tunggu untuk pemasangan saluran air sangat lama. Untuk mengatasi permasalahan tersebut maka digunakan algoritma *Ant Colony Optimization (ACO)* untuk pengelompokan data-data pelanggannya. Algoritma *ACO* merupakan pendekatan yang lebih baik untuk mencari solusi yang mendekati optimal. Nilai optimal yang digunakan pada algoritma *ACO* menggunakan bantuan graf atau rumus sehingga nilai optimal dapat diperkirakan dengan cepat dan tepat. Dibandingkan dengan pendekatan-pendekatan lainnya, *ACO* memiliki keunggulan waktu pemrosesan yang sangat cepat dan kemampuan adaptasi dengan graf yang berubah secara dinamis. Pengelompokan data-data pelanggan PDAM Surabaya dapat dimanfaatkan dalam peningkatan pelayanan kepada pelanggan khususnya pelanggan baru untuk memasang saluran air minum sehingga cakupan layanan PDAM Surabaya lebih luas banyak kawasan kota yang teraliri air PDAM.

Kata kunci: pengelompokan, *Ant Colony Optimization*, nilai optimal

Abstract

Perusahaan Daerah Air Minum (PDAM) Surya Sembada Kota Surabaya was considered slow in expanding the coverage of its customers. When a new customer want to propose the installation of the water line, customers wait long realization. Because the customer location data must be adapted to new data from the location of the main water pipeline takes a long time so that the waiting time for the installation of the water line was very long. To overcome these problems we used *Ant Colony Optimization algorithm (ACO)* for clustering data customers. *ACO* algorithm was a better approach to search for near-optimal solutions. Optimum value used in the *ACO* algorithm uses a graph or formula so that the optimal value can be estimated quickly and appropriately. If *ACO* compared with other approaches, the *ACO* has the advantage of very fast processing time and the ability to adapt to a dynamically changing graph. Clustering customer data PDAM Surabaya can be utilized in improving service to customers, especially new customers to install water channels so that the service coverage of PDAM Surabaya much wider area of the city that have both running water taps.

Keywords: clustering, *Ant Colony Optimization*, optimal values

1. PENDAHULUAN

Perkembangan dunia bisnis yang sangat pesat, mendorong terbentuknya suatu timbunan data-data yang berukuran sangat besar. Data-data tersebut pada umumnya berasal dari *data entry* dan *customer service*, kemudian oleh komputer data tersebut disimpan ke dalam server selanjutnya data diubah menjadi informasi yang disimpan dalam bentuk tabel [4]. *Data mining* mempunyai tujuan untuk mengekstraksi pengetahuan yang masih tersembunyi dari data yang sangat besar yang hasilnya tidak hanya akurat tetapi harus dapat dipahami oleh pengguna [3]. Bila pencarian pengetahuan yang tidak dapat dipahami oleh pengguna, maka tidak akan dapat diinterpretasikan secara benar. Hal ini kemungkinan pengguna tidak akan cukup yakin dengan pengetahuan untuk pengambilan keputusan, karena akan menyebabkan keputusan yang salah [6].

Dalam sebuah media *online* edisi 15 Oktober 2012 dengan judul “PDAM Dianggap Lamban Perluas Cakupan” [5], pelanggan mengeluhkan pelayanan PDAM Surabaya. Perusahaan Daerah Air Minum (PDAM) Surya

Sembada Kota Surabaya dinilai lamban dalam memperluas cakupan jumlah pelanggannya. Kawasan yang sampai sekarang belum tersentuh cakupan pelanggan PDAM di antaranya di Kelurahan Sambisari dan Lontar Kecamatan Sambikerep. Ketika pelanggan baru ingin mengajukan pemasangan saluran air, dimana yang menangani masalah pemasangan saluran air di Surabaya adalah Perusahaan Daerah Air Minum (PDAM) Surabaya, pelanggan menunggu lama realisasinya. Hal ini disebabkan karena data-data lokasi pelanggan baru harus disesuaikan dengan data-data lokasi dari pipa saluran air utama membutuhkan waktu yang lama sehingga waktu tunggu untuk pemasangan saluran air sangat lama sekitar lebih dari 1 bulan. Berdasarkan data dari PDAM Surabaya, jenis pelanggannya terdiri dari perumahan, pemerintah, perdagangan, industri, sosial umum, sosial khusus, dan pelabuhan. Dari tahun 2008 sampai dengan 2012 ada kenaikan jumlah pelanggan PDAM, tetapi sampai pada tahun 2012 PDAM mencakup layanan 86,88% [7].

Untuk mendapatkan informasi tentang data-data pelanggan maka digunakan metode-metode yang ada dalam *data mining*. Salah satu fungsional dalam *data mining* adalah pengelompokan (*clustering*). Pengelompokan ini bertujuan untuk mengelompokkan data berdasarkan tingkat kesamaan dari *record* yang dikelompokkan. Persamaan antar data dilihat dengan menggunakan kedekatan jarak antar *record* pada grafik. Salah satu algoritma yang dapat digunakan untuk mengelompokkan data adalah algoritma *Ant Colony* (algoritma semut) yang dikenal dengan nama algoritma *Ant Colony Optimization* (ACO). Algoritma ACO ini merupakan algoritma yang biasa digunakan untuk kasus optimasi, misalnya dalam kasus *Travelling Salesman Problem* (TSP). Algoritma ini merepresentasikan aksi yang dilakukan oleh koloni semut dalam mencari dan mengangkut bahan makanan. *Node* dianalogikan sebagai bahan makanan sedangkan semut merupakan *agent* dari algoritma yang bertugas mencari makanan dengan jarak tempuh seminimal mungkin, tiap *node* harus dilewati dengan jarak minimal.

Dengan melihat permasalahan tersebut diatas, maka dilakukan penelitian tentang pengelompokan data pelanggan PDAM Surabaya dengan menggunakan metode algoritma *Ant Colony Optimization*. Dengan menggunakan metode tersebut PDAM Surabaya dapat meningkatkan pelayanan pemasangan saluran air di Surabaya dengan lebih baik dan cepat.

2. TINJAUAN PUSTAKA

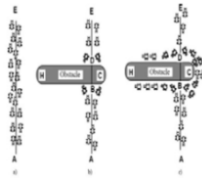
2.1 Algoritma Optimasi

Optimasi adalah suatu proses untuk mencapai hasil yang ideal atau optimal (nilai efektif yang dapat dicapai). Untuk mencapai nilai optimal baik minimal atau maksimal secara sistematis dilakukan pemilihan nilai variabel integer atau nyata yang akan memberikan solusi optimal. Nilai optimal adalah nilai yang didapat dengan melalui suatu proses dan dianggap menjadi suatu solusi jawaban yang paling baik dari semua solusi yang ada. Nilai optimal dapat dicari dengan 2 cara yaitu (1) cara konvensional, mencoba semua kemungkinan yang ada dengan mencatat nilai yang didapat; dan (2) cara dengan menggunakan suatu rumus atau gambar sehingga nilai optimal dapat diperkirakan dengan cepat dan tepat [1].

Algoritma optimasi merupakan suatu algoritma atau metode numerik untuk menemukan nilai x sedemikian menghasilkan $f(x)$ yang bernilai sekecil (atau sebesar) mungkin untuk suatu fungsi f yang diberikan, yang mungkin disertai dengan beberapa batasan pada x [8]. Nilai x berupa skalar atau vektor dari nilai-nilai kontinyu maupun diskrit. Hasil dari suatu proses optimasi adalah suatu himpunan masukan yang membuat fungsi-fungsi objektif menghasilkan nilai-nilai optimal (maksimal atau minimal). Algoritma optimasi sedikit berbeda dengan algoritma pencarian (*search algorithm*). Pada algoritma pencarian terdapat kriteria tertentu yang menyatakan apakah suatu elemen x merupakan solusi atau bukan. Pada algoritma optimasi berupa fungsi-fungsi objektif yang menggambarkan bagus atau tidaknya suatu konfigurasi yang diberikan. Karena fungsi-fungsi objektif tersebut bisa memberikan definisi masalah yang lebih umum, maka algoritma optimasi dikatakan sebagai generalisasi dari algoritma pencarian.

2.2 Algoritma Ant Colony Optimization

Ant Colony Optimization (ACO) termasuk teknik pencarian *multi agent* yang sering digunakan untuk permasalahan optimasi (khususnya kombinatorial). Algoritma ini terinspirasi oleh tingkah laku semut dalam suatu koloni. Algoritma ACO pertama kali diperkenalkan oleh Marco Dorigo pada tahun 1991 kemudian dipublikasikan dengan nama *Ant System* [1][2]. Ketika semut-semut mencari makan dapat menemukan jalur terpendek antara sumber makanan dan sarangnya. Semut berjalan dari sumber makanan ke sarang dan sebaliknya, semut meletakkan suatu zat (*pheromone*) di sepanjang jalur yang semut lalui. Ketika zat tersebut disekresikan sebagai isyarat seekor semut maka semut yang lain dapat mengenalinya. *Pheromone* ini akan membimbing semut lain untuk menemukan sumber makanan. Jumlah *pheromone* yang ditinggalkan oleh semut bergantung pada jumlah makanan yang ditemukan. Semakin banyak makanan yang didapat, semakin banyak pula jumlah *pheromone* yang ditinggalkan. Sehingga semakin banyak semut yang melewati suatu jalur semakin kuat pula jejak *pheromone* yang terkumpul di jalur tersebut.



Gambar 1 Mekanisme pergerakan koloni semut

Pada Gambar 1 menunjukkan perjalanan semut yang berjalan dari titik A ke titik E. Pada awalnya ketika belum diberikan pembatas, maka semut akan berjalan dengan jumlah yang sama di sebelah kiri dan kanan garis pembatas. Ketika diberikan penghalang maka semut pada awalnya akan sama. Akan tetapi, lama kelamaan semut akan cenderung melewati sebelah kanan garis pembatas karena jarak yang ditempuh lebih pendek. Hal ini dikarenakan pengaruh *pheromone* tadi.

3. HASIL PERCOBAAN, ANALISA DAN EVALUASI SISTEM

ACO termasuk teknik pencarian *multi agent* untuk menyelesaikan permasalahan optimasi yang terinspirasi tingkah laku semut dalam suatu koloni. Algoritma ini terinspirasi oleh tingkah laku koloni semut, hewan yang hampir buta dengan kemampuan individu yang sederhana namun dapat menemukan jalan terpendek dari sarang menuju sumber makanan, ketika bekerjasama dengan semut-semut lainnya dalam suatu koloni. Algoritma ACO secara umum dapat ditunjukkan pada Gambar 2.

Algoritma Ant Colony Optimization (ACO)

1. Inisialisasi
 $t=0$ {t adalah penghitung waktu}
 $NC=0$ {NC adalah penghitung *cycle*}
 Untuk setiap busur (i,j), tentukan nilai awal untuk intensitas jejak, $\tau_{ij}(t)=c$ dan $\Delta\tau_{ij}(t)=0$
 Letakkan m semut pada n nodes (kota)
 $s=1$ {s adalah indeks untuk tabu list}
 for $k=1$ to m do
 Letakkan kota awal untuk semut ke-k di dalam tabuk (s)
 end
2. repeat {langkah ini diulang (n-1) kali}
 $s=s+1$
 for $k=1$ to m do
 pilih kota j sebagai kota berikutnya yang akan dikunjungi dengan probabilitas $p_{ij}^k(t)$ {pada waktu t, semut ke-k berada pada kota $i=tabuk(s-1)$ }
 pindahkan semut ke-k ke kota j
 masukkan kota j ke dalam tabuk (s)
 end
 until tabu list penuh
3. for $k=1$ to m do
 pindahkan semut ke-k dari tabuk (n) ke tabuk (1)
 hitung L_k =panjang *tour* yang dihasilkan oleh semut ke-k
 update *tour* terpendek {jika L_k saat ini lebih pendek dari L_k sebelumnya}
 end
 untuk setiap busur (i,j) lakukan
 for $k=1$ to m do

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{jika } (i,j) \in \text{tour yang ada di tabuk}_k \\ 0 & \text{untuk lainnya} \end{cases}$$

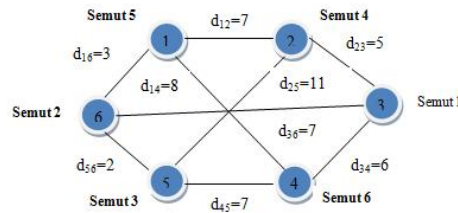
$$\Delta\tau_{ij} = \Delta\tau_{ij} + \Delta\tau_{ij}^k$$
 end
4. untuk setiap busur (i,j) hitung $\tau_{ij}(t+n)$ berdasarkan persamaan $\tau_{ij}(t+n) = \rho\tau_{ij}(t) + \Delta\tau_{ij}$
 $t=t+1$
 $NC=NC+1$
 Untuk setiap busur (i,j) set $\Delta\tau_{ij} = 0$
5. if (NC < NC_{MAX}) and (bukan tingkah laku yang stagnan) then
 kosongkan semua tabu list
 kembali ke langkah 2
 else
 return *tour* terpendek
 berhenti {semua proses iterasi telah selesai}
 end

Gambar 2 Algoritma Ant Colony Optimization (ACO)

Dalam masalah ini digunakan bantuan *graph* simetris dengan 6 simpul (data sebagai kelurahan) seperti tampak pada Gambar 3. Pada *graph* simetris tak berarah, biaya (jarak) dari simpul 1 ke simpul 2 adalah sama dengan biaya simpul dari simpul 2 ke simpul 1 yang dinyatakan dengan simbol d_{12} . Datanya adalah sebagai berikut:

- | | |
|--|--|
| (i) Simpul 1: kelurahan Kendangsari Desa | (iv) Simpul 4: kelurahan Kendangsari Taman |
| (ii) Simpul 2: kelurahan Tenggilis | (v) Simpul 5: kelurahan Jemur Andayani |
| (iii) Simpul 3: kelurahan Jemursari | (vi) Simpul 6: kelurahan Wonocolo |

Pada inialisasi, misalkan sejumlah $m = 6$ ekor semut diletakkan secara acak pada $n = 6$ simpul pada *graph* tersebut. Masukkan setiap simpul awal dari semua semut ke dalam tabu list masing-masing semut. Dalam hal ini $\text{tabu}_1 = \{3\}$, $\text{tabu}_2 = \{6\}$, $\text{tabu}_3 = \{5\}$, $\text{tabu}_4 = \{2\}$, $\text{tabu}_5 = \{1\}$, dan $\text{tabu}_6 = \{4\}$. Intensitas jejak *pheromone* untuk semua busur diisi dengan konstanta, $\tau_{ij}(t) = c$, artinya semua busur memiliki intensitas sama karena memang belum ada seekor semutpun yang melewati busur.



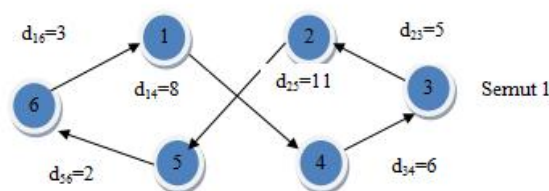
Gambar 3 Inialisasi 6 ekor semut diletakkan secara acak

Selanjutnya setiap semut membuat suatu tour. Semut 1 pada posisi awalnya berada di simpul 3 akan bergerak ke simpul berikutnya. Terdapat 3 pilihan simpul yang bias dikunjungi yaitu simpul 2, 4 dan 6. Semut akan memilih salah satu dari ketiga simpul tersebut dengan probabilitas $p_{ij}^k(t)$ yaitu:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [n_{ij}]^\beta}{\sum_{k \in \text{lewat}} [\tau_{ik}(t)]^\alpha [n_{ik}]^\beta} & , \text{jika } j \in \text{lewat } k \\ 0 & , \text{untuk lainnya} \end{cases} \quad (1)$$

Karena belum ada seekor semutpun yang melewati busur-busur yang ada, maka intensitas jejak *pheromone* masih sama untuk semua busur. Sehingga probabilitas $p_{ij}^k(t)$ masih didominasi oleh *visibility* n_{ij} (misalkan $n_{ij} = (1/d_{ij})$) dimana d_{ij} adalah jarak simpul i ke simpul j). diantara ketiga simpul tersebut, simpul 2 memiliki probabilitas paling besar karena jarak dari simpul 3 ke simpul 2 paling kecil, hal ini belum pasti terpilih. Setelah dihitung probabilitas ketiga simpul tersebut, proses pemilihan simpul bias dilakukan dengan cara membangkitkan sebuah bilangan acak. Misal yang terpilih simpul 2, semut bergerak ke simpul 2 sehingga tabu_1 di-update menjadi $\{3,2\}$. Pada saat berada di simpul 2, semut akan melanjutkan ke kunjungan ke simpul berikutnya. Terdapat 2 pilihan simpul yaitu simpul 1 atau 5 (simpul 3 tidak boleh dikunjungi) karena sudah berada di tabu_1 . Misal semut memilih dan bergerak ke simpul 5 sehingga tabu_1 di-update menjadi $\{3,2,5\}$. Pada simpul 5, semut memiliki 2 pilihan simpul yaitu simpul 4 atau 6. Semut memilih dan bergerak ke simpul 6 sehingga tabu_1 di-update menjadi $\{3,2,5,6\}$. Saat berada di simpul 6, semut hanya memiliki 1 pilihan simpul berikutnya yaitu simpul 1. Semut bergerak ke simpul 1 dan tabu_1 di-update menjadi $\{3,2,5,6,1\}$. dari simpul 1 tersebut, semut bergerak ke simpul 4 sehingga tabu_1 di-update menjadi $\{3,2,5,6,1,4\}$. Dengan demikian tabu_1 menjadi penuh.

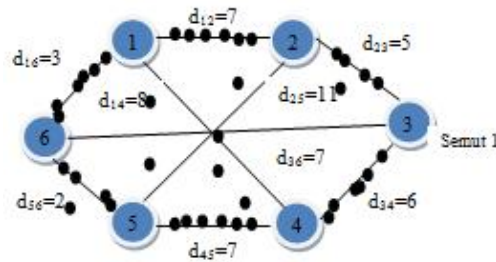
Selanjutnya semut bergerak dari $\text{tabu}_1(n)$ ke $\text{tabu}_1(1)$ dimana $n = \text{jumlah simpul}$. Dari simpul 4 semut bergerak ke simpul 3 sehingga semut tersebut menghasilkan *tour* yang lengkap yaitu 3-2-5-6-1-4-3 dengan total jarak $5+11+2+3+8+6=35$. Jadi semut 1 menghasilkan 1 solusi berupa *tour* 3-2-5-6-1-4-3 dengan total jarak 35. Gambar 5 menunjukkan hasil dari *tour* semut 1 yang menghasilkan rute 3-2-5-6-1-4-3 dengan jarak 35.



Gambar 4 Hasil tour dari semut 1, $\text{tabu}_1 = \{3,2,5,6,1,4\}$

Selama pembuatan suatu *tour*, semut menaruh jejak *pheromone* pada semua busur yang dilewatinya. Pada Gambar 5 mengilustrasikan *pheromone* yang diletakkan oleh Semut 1 selama pembuatan suatu *tour*. Hal yang sama juga dilakukan oleh semua semut yang lain. Intensitas *pheromone* di semua busur di-update menggunakan persamaan:

$$\tau_{ij}(t+n) = \rho \tau_{ij}(t) + \Delta \tau_{ij} \quad (2)$$



Gambar 5 Intensitas jejak pheromone yang sering dilewati

Sehingga jejak-jejak *pheromone* yang ditinggalkan oleh semut tersebut akan lebih sering memilih busur-busur tersebut. Pada akhirnya dihasilkan sebuah *tour* dengan total jarak terpendek 30 untuk rute 1-2-3-4-5-6-1. Dengan cara yang sama dilakukan untuk semut-semut yang lain dapat dilihat pada Tabel 1 di bawah ini. Karena terdapat 6 semut maka dihasilkan 6 *tour* lengkap. Selanjutnya dari 6 *tour* yang ada dipilih *tour* yang total jaraknya paling kecil. Dari tabel 1, jarak terpendek dengan menggunakan algoritma ACO adalah 30.

Tabel 1 Hasil percobaan Algoritma Ant Colony Optimization

Tabu list	Isi tabu list	Rute	Jarak	Tabu list	Isi tabu list	Rute	Jarak
Tabu ₁	{3,2,5,6,1,4}	3-2-5-6-1-4-3	35	Tabu ₄	{2,1,6,5,4,3}	2-1-6-5-4-3-2	30
	{3,2,1,6,5,4}	3-2-1-6-5-4-3	30		{2,1,4,3,6,5}	2-1-4-3-6-5-2	43
	{3,2,5,4,1,6}	3-2-5-4-1-6-3	43		{2,1,4,5,6,3}	2-1-4-5-6-3-2	38
	{3,2,1,4,5,6}	3-2-1-4-5-6-3	38		{2,1,6,3,4,5}	2-1-6-3-4-5-2	43
	{3,4,1,2,5,6}	3-4-1-2-5-6-3	43		{2,3,4,5,6,1}	2-3-4-5-6-1-2	30
	{3,4,5,2,1,6}	3-4-5-2-1-6-3	43		{2,3,4,1,6,5}	2-3-4-1-6-5-2	35
	{3,4,1,6,5,2}	3-4-1-6-5-2-3	35		{2,3,6,5,4,1}	2-3-6-5-4-1-2	38
	{3,4,5,6,1,2}	3-4-5-6-1-2-3	30		{2,3,6,5,4,1}	2-3-6-5-4-1-2	43
	{3,6,1,2,5,4}	3-6-1-2-5-4-3	43		{2,5,4,3,6,1}	2-5-4-3-6-1-2	43
	{3,6,1,4,5,2}	3-6-1-4-5-2-3	43		{2,5,4,1,6,3}	2-5-4-1-6-3-2	43
	{3,6,5,4,1,2}	3-6-5-4-1-2-3	38		{2,5,6,3,4,1}	2-5-6-3-4-1-2	43
	{3,6,5,2,1,4}	3-6-5-2-1-4-3	46		{2,5,6,1,4,3}	2-5-6-1-4-3-2	35
Tabu ₂	{6,1,2,3,4,5}	6-1-2-3-4-5-6	30	Tabu ₅	{1,2,3,4,5,6}	1-2-3-4-5-6-1	30
	{6,1,4,3,2,5}	6-1-4-3-2-5-6	35		{1,2,5,4,3,6}	1-2-5-4-3-6-1	43
	{6,1,2,5,4,3}	6-1-2-5-4-3-6	43		{1,2,3,6,5,4}	1-2-3-6-5-4-1	38
	{6,1,4,5,2,3}	6-1-4-5-2-3-6	43		{1,2,5,6,3,4}	1-2-5-6-3-4-1	43
	{6,5,4,3,2,1}	6-5-4-3-2-1-6	30		{1,6,5,4,3,2}	1-6-5-4-3-2-1	30
	{6,5,2,3,4,1}	6-5-2-3-4-1-6	35		{1,6,5,2,3,4}	1-6-5-2-3-4-1	35
	{6,5,2,1,4,3}	6-5-2-1-4-3-6	38		{1,4,3,2,5,6}	1-4-3-2-5-6-1	35
	{6,5,4,1,2,3}	6-5-4-1-2-3-6	43		{1,4,3,6,5,2}	1-4-3-6-5-2-1	43
	{6,3,2,5,4,1}	6-3-2-5-4-1-6	46		{1,4,5,2,3,6}	1-4-5-2-3-6-1	43
	{6,3,4,1,2,5}	6-3-4-1-2-5-6	43		{1,4,5,6,3,2}	1-4-5-6-3-2-1	38
	{6,3,2,1,4,5}	6-3-2-1-4-5-6	38		{1,6,3,4,5,2}	1-6-3-4-5-2-1	43
	{6,3,4,5,2,1}	6-3-4-5-2-1-6	43		{1,6,3,2,5,4}	1-6-3-2-5-4-1	43
Tabu ₃	{5,4,3,2,1,6}	5-4-3-2-1-6-5	30	Tabu ₆	{4,3,2,1,6,5}	4-3-2-1-6-5-4	30
	{5,4,1,2,3,6}	5-4-1-2-3-6-5	38		{4,3,2,5,6,1}	4-3-2-5-6-1-4	35
	{5,4,3,6,1,2}	5-4-3-6-1-2-5	43		{4,3,6,5,2,1}	4-3-6-5-2-1-4	43
	{5,4,1,6,3,2}	5-4-1-6-3-2-5	43		{4,3,6,1,2,5}	4-3-6-1-2-5-4	35
	{5,6,1,2,3,4}	5-6-1-2-3-4-5	30		{4,1,2,3,6,5}	4-1-2-3-6-5-4	38
	{5,6,1,4,3,2}	5-6-1-4-3-2-5	35		{4,1,2,5,6,3}	4-1-2-5-6-3-4	43
	{5,6,3,2,1,4}	5-6-3-2-1-4-5	36		{4,1,6,5,2,3}	4-1-6-5-2-3-4	35
	{5,6,3,4,1,2}	5-6-3-4-1-2-5	43		{4,1,6,3,2,5}	4-1-6-3-2-5-4	43
	{5,2,3,6,1,4}	5-2-3-6-1-4-5	30		{4,5,6,1,2,3}	4-5-6-1-2-3-4	30
	{5,2,3,4,1,6}	5-2-3-4-1-6-5	38		{4,5,2,3,6,1}	4-5-2-3-6-1-4	43
	{5,2,1,4,3,6}	5-2-1-4-3-6-5	43		{4,5,2,1,6,3}	4-5-2-1-6-3-4	43
	{5,2,1,6,3,4}	5-2-1-6-3-4-5	35		{4,5,6,3,2,1}	4-5-6-3-2-1-4	38

4. SIMPULAN DAN SARAN

4.1 Simpulan

Algoritma *Ant Colony Optimization* (ACO) merupakan suatu algoritma dengan pendekatan yang lebih baik untuk mencari solusi yang mendekati optimal. Nilai optimal yang digunakan pada algoritma ACO menggunakan bantuan graf atau rumus sehingga nilai optimal dapat diperkirakan dengan cepat dan tepat. Dibandingkan dengan

pendekatan-pendekatan lainnya, ACO memiliki keunggulan waktu pemrosesan yang sangat cepat dan kemampuan adaptasi dengan graf yang berubah secara dinamis. Sehingga algoritma ACO dapat digunakan untuk pengelompokan data-data pelanggan PDAM Surabaya.

4.2 Saran

1. Perlu dikembangkan lagi dengan menggunakan salah satu metode pengelompokan yang lain yaitu berbasis metode statistik (Single Linkage, Complete Linkage, Average Linkage, K-Means dan lain-lain), berbasis fuzzy (Fuzzy C-Means), berbasis *neural network* (Kohonen SOM atau LVQ) atau metode lain untuk optimasi centroid atau lebar *cluster* (Genetik Algoritma).
2. Disamping itu juga perlu dilengkapi dengan adanya aplikasi Sistem Informasi Geografis (SIG) yang menampilkan peta-peta Surabaya sebagai simulasi dengan menggunakan aplikasi *watercad*.

5. DAFTAR RUJUKAN

- [1] Anil, Jain, 1998, *Algorithm for Clustering Data*, Prentice Hall, New Jersey
- [2] Dorigo, M and Stutzle, Thomas, 2004, *Ant Colony Optimization*, The MIT Press, Massachusets
- [3] Kapiudin, Maulani, 2007, *Datamining untuk Klasifikasi Pelanggan dengan Ant Colony Optimization*, Jurnal Informatika Vol. 8 No. 1, pp. 68-73
- [4] Laudon., K.C., Laudon, J.P., 2005, *Sistem Informasi Manajemen*, ANDI, Yogyakarta
- [5] Post, Surabaya, 2012, *PDAM Dianggap Lamban Perluas Cakupan*, [Online] (Update 25 April 2013)
- [6] Sharma, S., 1996, *Applied Multivariate Technique*, John Wiley & Sons Inc., New York
- [7] Surabaya, Surya Sembada PDAM, 2012, *Jumlah Pelanggan Tahunan*, [Online] (Update 25 April 2013)
- [8] Suyanto, 2010, *Algoritma Optimasi : Deterministik atau Probabilitik*, Graha Ilmu, Yogyakarta