

Algoritma ant colony optimization (ACO) dalam optimalisasi rute pada vehicle routing problem with time window

Dian Fitrotin Septiashri¹, Purwanto², Darmawan Satyananda³

JURUSAN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS NEGERI MALANG

E-mail: chan_daient@yahoo.co.id

Abstrak: *Vehicle Routing Problem with Time Window* (VRPTW) merupakan permasalahan bagaimana suatu depot distribusi barang, dengan sejumlah kendaraan berkapasitas tertentu melayani sejumlah *customer* pada lokasi yang terpisah, dengan permintaan dan batasan *time window* tertentu dengan tujuan meminimalkan total biaya perjalanan, tanpa mengabaikan batasan kapasitas kendaraan dan *time window* depot. Algoritma *Ant Colony Optimization* terinspirasi dari perilaku semut, yaitu perjalanan semut dalam mencari makan. Semut mempunyai zat khusus yang disebut *pheromone*. Terdapat tiga langkah pada algoritma *Ant Colony Optimization* yaitu langkah inisialisasi awal jarak *pheromone*, *update pheromone lokal* dan *update pheromone global*. Dari ketiga langkah tersebut terdapat iterasi yaitu perbaikan *pheromone* agar menghasilkan solusi terbaik dari rute yang terbentuk oleh semut. Oleh sebab itu, untuk mempermudah pencarian rute, Algoritma ACO tersebut diimplementasikan ke dalam program komputer menggunakan Delphi 7.

Kata Kunci: *Vehicle Routing Problem with Time Window* (VRPTW), Algoritma *Ant Colony Optimization*,

Abstract: *Vehicle Routing Problem with Time Window* (VRPTW) is problem about how a depot of goods, with a certain number of vehicles with a specified capacity serves some customers at separate locations, which have certain demand and limited time window, with the aim is minimizing the total cost of the trip, without ignoring the limits of the vehicle and depot time window. *Ant Colony Optimization* algorithm inspired by ant behavior, that is ant journey in foraging. Ant have special compound called *pheromone*. There are three steps in *Ant Colony Optimization* algorithm, those are initialization distance *pheromone*, local *pheromone update* and global *pheromone update*. ACO uses iteration to *pheromone* in order to produce the best solution of routes formed by ants. Therefore, to simplify the search of route, the ACO algorithm is implemented into a computer program using Delphi 7.

Keywords: *Vehicle Routing Problem with Time Window* (VRPTW), *Ant Colony Optimization Algorithm*.

Salah satu cabang matematika yang banyak membantu permasalahan dalam kehidupan sehari-hari adalah teori *graph*. Teori *graph* dapat diperlihatkan

-
1. Dian Fitrotin Septiashri adalah mahasiswa jurusan Matematika FMIPA Universitas Negeri Malang
 2. Purwanto adalah dosen jurusan matematika FMIPA Univeritas Negeri Malang
 3. Darmawan Satyananda adalah dosen jurusan matematika FMIPA Univeritas Negeri Malang

peranan dan kegunaannya dalam memecahkan berbagai permasalahan dengan mengkaji dan menganalisa model atau uraian (Purwanto, 1998). Salah satu model yang banyak dipakai adalah *Vehicle Routing Problem* (VRP). Pendiskripsian VRP merupakan permasalahan dalam menentukan sejumlah rute untuk sekumpulan kendaraan identik yang harus melayani sejumlah *customer* dari depot pusat. Tujuan dari permasalahan VRP adalah untuk melayani *customer* sesuai dengan permintaan dengan meminimalkan biaya angkut dan jumlah kendaraan yang dimulai dan berakhir di depot pusat. Salah satu varian dari VRP adalah *Vehicle Routing Problem with Time Window* (VRPTW) yang merupakan permasalahan bagaimana suatu depot dengan sejumlah kendaraan berkapasitas tertentu dalam melayani sejumlah *customer* pada titik-titik lokasi yang terpisah, dengan permintaan dan batasan *time window* tertentu dengan tujuan meminimalkan total biaya perjalanan tanpa mengabaikan batasan kapasitas kendaraan dan *time window* depot pusat.

Banyak metode optimasi heuristik telah muncul. Salah satu metode optimasi heuristik yaitu didasarkan atas analogi jalan semut. Metode ini dikenal dengan metode *Ant Colony Optimization* (ACO), termasuk suatu kelompok teknik heuristik yang secara kolektif dikenal dengan *Swarm Intelligence*. *Ant Colony Optimization* diadopsi dari perilaku koloni semut yang dikenal sebagai sistem semut (Dorigo, 1996). Koloni semut dapat menemukan rute terpendek antara sarang dan sumber makanan berdasarkan jejak kaki pada lintasan yang telah dilalui. Semakin banyak semut yang melalui suatu lintasan, maka akan semakin jelas bekas jejak kakinya. Hal ini akan menyebabkan lintasan yang dilalui semut dalam jumlah sedikit, semakin lama akan semakin berkurang kepadatan semut yang melewatinya, atau bahkan akan tidak dilewati sama sekali. Sebaliknya, lintasan yang dilalui semut dalam jumlah banyak, semakin lama akan semakin bertambah kepadatan semut yang melewatinya, atau bahkan semua semut akan melalui lintasan tersebut (Dorigo, 1996).

Berdasarkan uraian di atas, akan dibahas mengenai algoritma *Ant Colony Optimization* (ACO) pada permasalahan *Vehicle Routing problem with Time Window* (VRPTW).

HASIL YANG DIHARAPKAN

1. Menjelaskan Algoritma *Ant Colony Optimization* (ACO) pada *Vehicle Routing Problem with Time Window*.
2. Menerapkan Algoritma *Ant Colony Optimization* (ACO) pada *Vehicle Routing Problem with Time Window* dengan contoh-contoh khusus.
3. Implementasi permasalahan *Vehicle Routing Problem with Time Window* dengan Algoritma *Ant Colony Optimization* menggunakan *Delphi 7*.
4. Menjelaskan simulasi aplikasi Algoritma *Ant Colony Optimization* (ACO) menggunakan Bahasa *Delphi 7*.

HASIL DAN PEMBAHASAN

☐ Vehicle Routing Problem with Time Window (VRPTW)

Vehicle Routing Problem with Time Windows (VRPTW) merupakan permasalahan bagaimana suatu depot dengan sejumlah kendaraan berkapasitas

tertentu untuk melayani sejumlah *customer* pada lokasi yang terpisah, dengan permintaan dan *time window* tertentu dengan tujuan meminimalkan total biaya perjalanan tanpa mengabaikan batasan kapasitas kendaraan dan *time window* depot pusat.

Permasalahan VRPTW dapat dideskripsikan sebagai berikut:

1. Terdapat satu depot pusat distribusi barang dan sejumlah kendaraan pada depot dengan kapasitas tertentu yang melayani permintaan *customer*.
2. Setiap kendaraan harus memulai dan mengakhiri rutenya pada depot pusat.
3. Jumlah total permintaan yang akan dilayani setiap kendaraan tidak boleh melebihi kapasitas kendaraan
4. Tiap *customer* i j k l m n p q r s t u v w x y z aa ab ac ad ae af ag ah ai aj ak al am an ao ap aq ar as at au av aw ax ay az ba bb bc bd be bf bg bh bi bj bk bl bm bn bo bp bq br bs bt bu bv bw bx by bz ca cb cc cd ce cf cg ch ci cj ck cl cm cn co cp cq cr cs ct cu cv cw cx cy cz da db dc dd de df dg dh di dj dk dl dm dn do dp dq dr ds dt du dv dw dx dy dz ea eb ec ed ee ef eg eh ei ej ek el em en eo ep eq er es et eu ev ew ex ey ez fa fb fc fd fe ff fg fh fi fj fk fl fm fn fo fp fq fr fs ft fu fv fw fx fy fz ga gb gc gd ge gf gg gh gi gj gk gl gm gn go gp gq gr gs gt gu gv gw gx gy gz ha hb hc hd he hf hg hh hi hj hk hl hm hn ho hp hq hr hs ht hu hv hw hx hy hz ia ib ic id ie if ig ih ii ij ik il im in io ip iq ir is it iu iv iw ix iy iz ja jb jc jd je jf jj jk jl jm jn jo jp jq jr js jt ju jv jw jx jy jz ka kb kc kd ke kf kg kh ki kj kl km kn ko kp kq kr ks kt ku kv kw kx ky kz la lb lc ld le lf lg lh li lj lk ll lm ln lo lp lq lr ls lt lu lv lw lx ly lz ma mb mc md me mf mg mh mi mj mk ml mn mo mp mq mr ms mt mu mv mw mx my mz na nb nc nd ne nf ng nh ni nj nk nl nm nn no np nq nr ns nt nu nv nw nx ny nz oa ob oc od oe of og oh oi oj ok ol om on oo op oq or os ot ou ov ow ox oy oz pa pb pc pd pe pf pg ph pi pj pk pl pm pn po pp pq pr ps pt pu pv pw px py pz qa qb qc qd qe qf qg qh qi qj qk ql qm qn qo qp qq qr qs qt qu qv qw qx qy qz ra rb rc rd re rf rg rh ri rj rk rl rm rn ro rp rq rr rs rt ru rv rw rx ry rz sa sb sc sd se sf sg sh si sj sk sl sm sn so sp sq sr ss st su sv sw sx sy sz ta tb tc td te tf tg th ti tj tk tl tm tn to tp tq tr ts tt tu tv tw tx ty tz ua ub uc ud ue uf ug uh ui uj uk ul um un uo up uq ur us ut uu uv uw ux uy uz va vb vc vd ve vf vg vh vi vj vk vl vm vn vo vp vq vr vs vt vu vv vw vx vy vz wa wb wc wd we wf wg wh wi wj wk wl wm wn wo wp wq wr ws wt wu wv ww wx wy wz xa xb xc xd xe xf xg xh xi xj xk xl xm xn xo xp xq xr xs xt xu xv xw xx xy xz ya yb yc yd ye yf yg yh yi yj yk yl ym yn yo yp yq yr ys yt yu yv yw yx yy yz za zb zc zd ze zf zg zh zi zj zk zl zm zn zo zp zq zr zs zt zu zv zw zx zy zz aa ab ac ad ae af ag ah ai aj ak al am an ao ap aq ar as at au av aw ax ay az ba bb bc bd be bf bg bh bi bj bk bl bm bn bo bp bq br bs bt bu bv bw bx by bz ca cb cc cd ce cf cg ch ci cj ck cl cm cn co cp cq cr cs ct cu cv cw cx cy cz da db dc dd de df dg dh di dj dk dl dm dn do dp dq dr ds dt du dv dw dx dy dz ea eb ec ed ee ef eg eh ei ej ek el em en eo ep eq er es et eu ev ew ex ey ez fa fb fc fd fe ff fg fh fi fj fk fl fm fn fo fp fq fr fs ft fu fv fw fx fy fz ga gb gc gd ge gf gg gh gi gj gk gl gm gn go gp gq gr gs gt gu gv gw gx gy gz ha hb hc hd he hf hg hh hi hj hk hl hm hn ho hp hq hr hs ht hu hv hw hx hy hz ia ib ic id ie if ig ih ii ij ik il im in io ip iq ir is it iu iv iw ix iy iz ja jb jc jd je jf jj jk jl jm jn jo jp jq jr js jt ju jv jw jx jy jz ka kb kc kd ke kf kg kh ki kj kl km kn ko kp kq kr ks kt ku kv kw kx ky kz la lb lc ld le lf lg lh li lj lk ll lm ln lo lp lq lr ls lt lu lv lw lx ly lz ma mb mc md me mf mg mh mi mj mk ml mn mo mp mq mr ms mt mu mv mw mx my mz na nb nc nd ne nf ng nh ni nj nk nl nm nn no np nq nr ns nt nu nv nw nx ny nz oa ob oc od oe of og oh oi oj ok ol om on oo op oq or os ot ou ov ow ox oy oz pa pb pc pd pe pf pg ph pi pj pk pl pm pn po pp pq pr ps pt pu pv pw px py pz qa qb qc qd qe qf qg qh qi qj qk ql qm qn qo qp qq qr qs qt qu qv qw qx qy qz ra rb rc rd re rf rg rh ri rj rk rl rm rn ro rp rq rr rs rt ru rv rw rx ry rz sa sb sc sd se sf sg sh si sj sk sl sm sn so sp sq sr ss st su sv sw sx sy sz ta tb tc td te tf tg th ti tj tk tl tm tn to tp tq tr ts tt tu tv tw tx ty tz ua ub uc ud ue uf ug uh ui uj uk ul um un uo up uq ur us ut uu uv uw ux uy uz va vb vc vd ve vf vg vh vi vj vk vl vm vn vo vp vq vr vs vt vu vv vw vx vy vz wa wb wc wd we wf wg wh wi wj wk wl wm wn wo wp wq wr ws wt wu wv ww wx wy wz xa xb xc xd xe xf xg xh xi xj xk xl xm xn xo xp xq xr xs xt xu xv xw xx xy xz ya yb yc yd ye yf yg yh yi yj yk yl ym yn yo yp yq yr ys yt yu yv yw yx yy yz za zb zc zd ze zf zg zh zi zj zk zl zm zn zo zp zq zr zs zt zu zv zw zx zy zz

Fungsi tujuan VRPTW yaitu untuk meminimalkan total biaya perjalanan semua kendaraan. Sedangkan semua batasannya juga sama dengan VRP persamaan tetapi perlu dengan batasan *time window*. Batasan-batasan untuk VRPTW adalah sebagai berikut:

Batasan 1: Setiap kendaraan k memulai pelayanan pada titik i dengan memenuhi

$$\sum_{j \in N} x_{ij}^k \geq 1 \quad (k \in K)$$

Dengan

Batasan 2: Pelayanan harus dilakukan berdasarkan batasan waktu tiap-tiap titik

Batasan 3: Setiap kendaraan harus kembali sebelum waktu akhir depot pusat

$$\sum_{i \in N} x_{ik}^k \leq 1 \quad (k \in K)$$

Keterangan:

t_i adalah waktu kendaraan k tiba pada waktu

t_i^* adalah awal waktu untuk melayani titik

t_i^* adalah akhir waktu untuk melayani titik

t_{ij} adalah waktu perjalanan dari i ke j

t_{ij}^* adalah waktu tunggu sebelum melayani di titik

t_{ij}^* adalah waktu pelayanan di titik

☐ **Algoritma Ant Colony Optimization (ACO) pada Vehicle Routing Problem with Time Window (VRPTW)**

Algoritma *Ant Colony Optimization* (ACO) adalah algoritma baru yang dikembangkan sebagai alternatif untuk menyelesaikan optimasi (Dorigo,

1996). Dalam penyelesaian VRPTW untuk algoritma *Ant Colony Optimization* (ACO), setiap semut melalui rute perjalanan dari depot dan mengunjungi semua *customer* dan hanya dapat dikunjungi satu kali. Jika jumlah total muatan *customer* semut melebihi batasan kapasitas kendaraan atau batasan *time window* dari depot pusat maka *customer* tidak bisa dikunjungi.

Menurut Bell (2004) Algoritma *Ant Colony Optimization* (ACO) terdiri dari beberapa tahap yaitu:

1. Konstruksi Rute

Semut diibaratkan suatu kendaraan dan rute yang terbentuk dilakukan dengan menyeleksi *customer* sampai semua *customer* dikunjungi. Setiap semut memulai rute dari depot. Semut memilih *customer* selanjutnya untuk dikunjungi dengan mendaftar *customer* yang mungkin dan kendala kapasitas dari *customer* tersebut terpenuhi. Aturan menentukan *customer* selanjutnya dinamakan aturan *state transition rule* atau aturan transisi status (Dorigo, 1997). Algoritma ACO membentuk rute dari semut pertama sampai semut selanjutnya. Jarak *customer* didefinisikan dengan d_{ij} yang membentuk rute. Hal ini terus dilakukan hingga semut dapat membentuk rute yang mungkin. Untuk memilih *customer* menggunakan persamaan:

$$P_{ij} = \frac{\tau_{ij} \cdot \eta_{ij}^{\alpha}}{\sum_{k \in N_i} \tau_{ik} \cdot \eta_{ik}^{\alpha}}$$

Dimana τ_{ij} adalah jumlah *pheromone* yang terdapat pada sisi antara *customer* dengan kemungkinan *customer* j . Nilai η_{ij} didefinisikan sebagai invers dari jarak antara 2 *customer* dan parameter α merupakan parameter pengendali jarak d_{ij} . *Customer* yang telah dikunjungi oleh semut disimpan pada N_i . Nilai α adalah bilangan pecahan acak, α adalah parameter. Ketika seleksi dilakukan, semut memilih nilai yang paling tinggi dari persamaan (1) kecuali jika $\alpha < \alpha_0$. Dalam hal ini, semut memilih variabel acak q ke *customer* selanjutnya pada probabilitas dari semut pada titik i yang menuju titik j , dimana persamaannya:

$$q \leq \frac{P_{ij}}{\sum_{k \in N_i} P_{ik}}$$

dengan menggunakan persamaan (1) dan (2) setiap semut mengikuti sisi-sisi secara acak berdasarkan jarak terpendek dan jumlah *pheromone* yang terbesar. Jika kendala kapasitas belum terpenuhi, semut akan kembali ke depot hingga semua *customer* telah dikunjungi.

2. Update pheromone trail

Setelah semua semut menyelesaikan rutenya masing-masing maka *pheromone* diupdate. *Update pheromone trail* termasuk *local updating* setelah solusi rute sudah dibangkitkan dan *global updating* untuk rute terbaik setelah solusi m telah ditetapkan sebelumnya.

Local updating dibentuk dengan mengurangi *pheromone* pada *customer* yang telah dikunjungi dengan menirukan penguapan *pheromone* dan menjamin tidak ada lintasan yang berpengaruh, dimana persamaannya:

dimana τ_{ij} adalah parameter penguapan *pheromone* dan τ_{ij}^0 adalah nilai *pheromone* awal pada suatu *customer*.

Setelah solusi τ_{ij} ditetapkan untuk membentuk kemungkinan rute yang mungkin. *Global trail updating* dilakukan dengan menambah *pheromone* pada semua titik termasuk rute terbaik yang dibentuk oleh semut. *Global trail updating* berdasarkan persamaan:

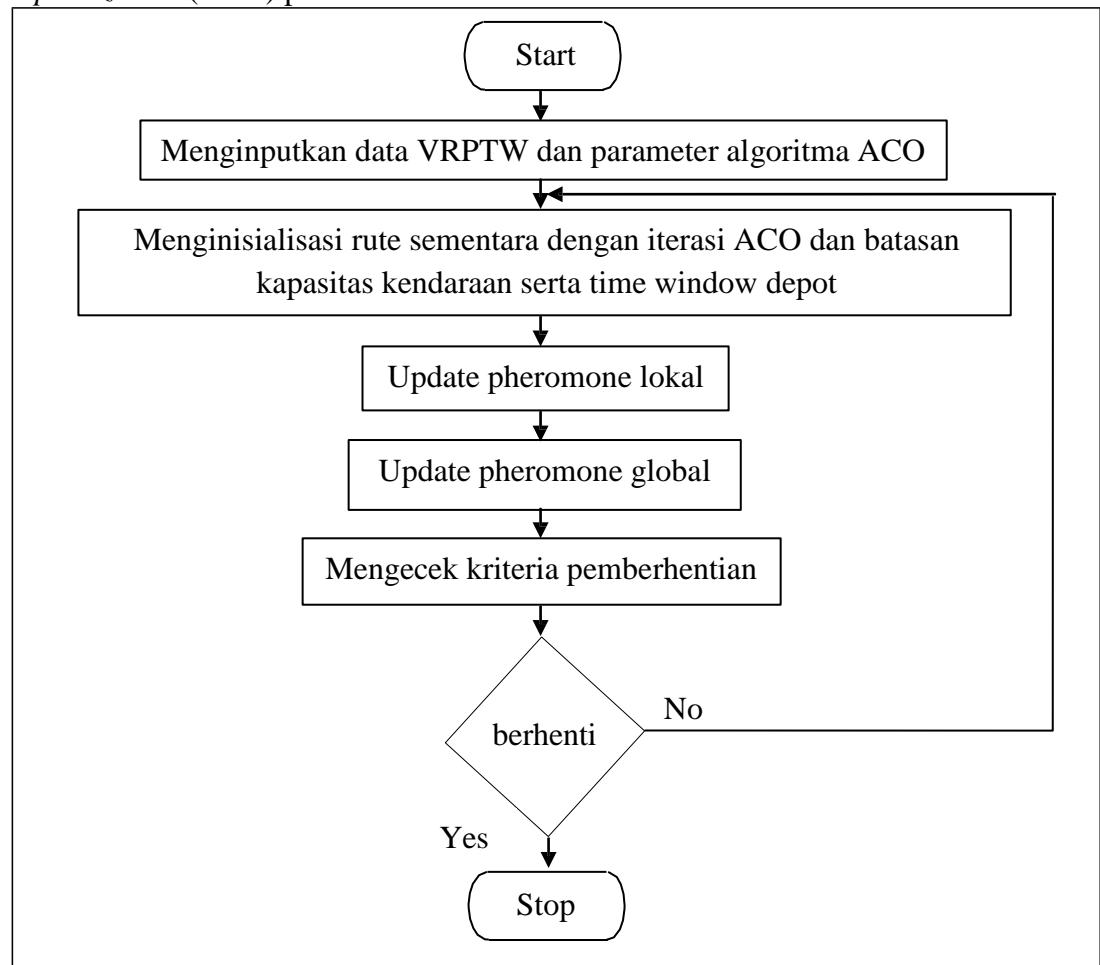
Global updating melanjutkan rute terpendek dan meningkatkan kemungkinan rute yang berisi solusi terbaik. Proses ini diulangi hingga ditentukan bilangan iterasi $iter$ dan solusi terbaik dari semua iterasi ditemukan.

Dari penjelasan Bell (2004) dapat diperoleh langkah-langkah Algoritma *Ant Colony Optimization* pada VRPTW adalah sebagai berikut:

1. Inialisasi awal jarak *pheromone* dan memilih semut untuk bergerak ke *customer* selanjutnya.
2. *Update pheromone lokal* dari masing-masing rute yang dibentuk semut.
3. *Update pheromone global* dari solusi terbaik yang dibentuk semut.

Dari ketiga langkah tersebut, terdapat iterasi untuk perbaikan sehingga kemungkinan rute yang berisi solusi terbaik bisa ditentukan.

Berikut ini adalah *Block Diagram* dari Algoritma *Ant Colony Optimization* (ACO) pada VRPTW



Gambar 1 : Block Diagram

☐ Contoh Penerapan

Suatu Agen Dipo Bangunan akan mendistribusikan keramik lantai ke 5 pelanggan. Permasalahannya adalah bagaimana mendesain rute yang paling optimal dengan tidak mengabaikan batasan yaitu tidak boleh melebihi kapasitas kendaraan dan batas *time window* sehingga dapat ditentukan total jarak tempuh minimum. Adapun data dari permasalahan tersebut adalah sebagai berikut:

Tabel 1. Data Depot

Data Depot	Time Window		Kecepatan rata-rata (km/jam)	Kapasitas kendaraan (dos)
	08.30	16.30	50	18

Tabel 2. Parameter yang digunakan

			Jumlah semut	Σ
0,9	1	0,5	3	2

Tabel 3. Jarak antar depot ke *customer* dan antar *customer* ke *customer*

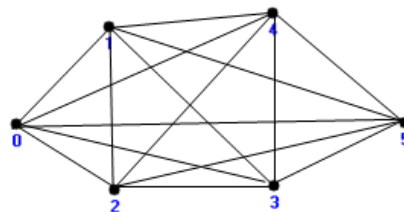
	0	1	2	3	4	5
0	0	4 km	3 km	7 km	5 km	9 km
1	4 km	0	5 km	2 km	6 km	4 km
2	3 km	5 km	0	7 km	6 km	3 km
3	7 km	2 km	7 km	0	4 km	5 km
4	5 km	6 km	6 km	4 km	0	8 km
5	9 km	4 km	3 km	5 km	8 km	0

Keterangan:

Untuk kolom pertama dan baris pertama, 0 menyatakan depot dan 1,2,...5 merupakan customer. Jarak dari 0 (depot) ke customer 1 adalah 4 km dan seterusnya.

Tabel 4. Data Customer

Customer	Kapasitas Permintaan (dos)	Service Time (menit)	Service Time (jam)
1	10	50	0,833
2	4	20	0,333
3	7	35	0,5833
4	3	15	0,25
5	10	50	0,833



Gambar 2. Model Graph

Pada Gambar 2, lokasi depot dan *customer-customer* diwakili oleh titik dan jalan yang dilalui dari depot menuju *customer* atau dari suatu *customer* ke *customer* lainnya diwakili oleh sisi.

Untuk menghitung nilai *visibility* dari depot ke *customer* dan antar *customer* digunakan rumus berikut:

Dimana v_{ij} adalah nilai *visibility* dari titik i ke titik j
 d_{ij} adalah jarak dari titik i ke titik j

Tabel 5. Nilai Visibility

	0	1	2	3	4	5
0	0	0,25	0,33	0,14	0,2	0,11
1	0,25	0	0,2	0,5	0,17	0,25
2	0,33	0,2	0	0,14	0,17	0,33
3	0,14	0,5	0,14	0	0,25	0,2
4	0,2	0,17	0,17	0,25	0	0,13
5	0,11	0,25	0,33	0,2	0,13	0

Keterangan:

Untuk kolom pertama dan baris pertama, 0 menyatakan depot dan 1,2,...5 merupakan *customer*. Nilai *visibility* dari 0 (depot) ke *customer* 1 adalah 0,25 dan seterusnya.

Berikut ini adalah penyelesaian permasalahan distribusi tersebut dengan menggunakan algoritma *Ant Colony Optimization* (ACO).

Tabel 6. pheromone awal

pheromone	0	1	2	3	4	5
0	0					
1		0				
2			0			
3				0		
4					0	
5						0

Keterangan:

Untuk kolom pertama dan baris pertama, 0 menyatakan depot dan 1,2,...5 merupakan *customer*. Jumlah *pheromone* dari 0 (depot) ke *customer* 1 adalah dan seterusnya.

Tabel 7. Iterasi 1

Rute awal	Customer	Rute Terpendek	Rute Terpilih

			(karena kapasitas kendaraan tidak memenuhi maka kembali ke)
			(karena kapasitas kendaraan tidak memenuhi maka kembali ke)
		Rute akhir :	

Tabel 8. Tabel nilai pheromone iterasi 1

pheromone	0	1	2	3	4	5
0	0					
1		0				
2			0			
3				0		
4					0	
5						0

Keterangan:

Untuk kolom pertama dan baris pertama, 0 menyatakan depot dan 1,2,...5 merupakan customer. Jumlah pheromone dari 0 (depot) ke customer 1 adalah dan seterusnya.

Tabel 9. Iterasi 2

Rute awal	Customer	Rute Terpendek	Rute Terpilih
			(karena kapasitas

			kendaraan tidak memenuhi maka kembali ke)
			(karena kapasitas kendaraan tidak memenuhi maka kembali ke)
		Rute akhir :	

Tabel 10. Tabel nilai pheromone akhir iterasi 2

pheromone	0	1	2	3	4	5
0	0					
1		0				
2			0			
3				0		
4					0	
5						0

Keterangan:

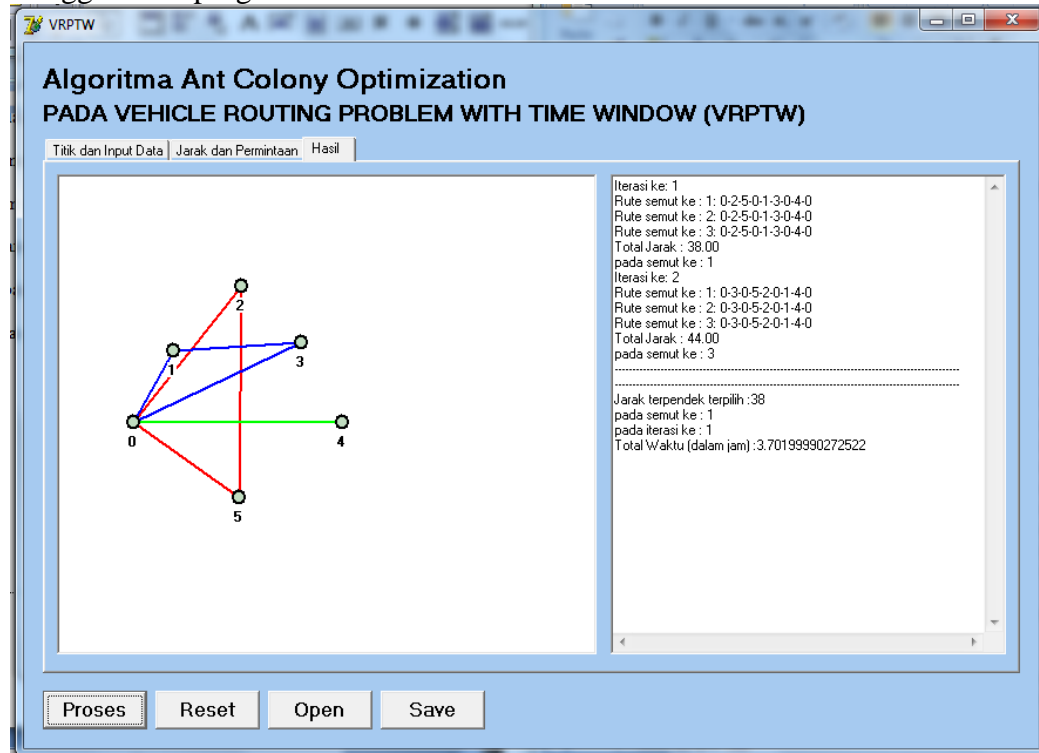
Untuk kolom pertama dan baris pertama, 0 menyatakan depot dan 1,2,...5 merupakan customer. Jumlah pheromone dari 0 (depot) ke customer 1 adalah dan seterusnya.

Pada permasalahan pendistribusian keramik lantai yang diselesaikan dengan algoritma *Ant Colony Optimization* (ACO) menghasilkan rute kendaraan = dengan jarak total = 38 km dengan time window jam. Dengan demikian pada permasalahan tersebut hanya membutuhkan 1 kendaraan yang akan digunakan untuk mendistribusikan keramik lantai ke 5 customernya karena masih memenuhi *time window* yang ditentukan.

Algoritma ACO diimplementasikan ke dalam program komputer. Untuk menjalankannya, diawali dengan menginputkan data seperti banyaknya titik dengan titik 0 sebagai depot dan titik lainnya sebagai *customer*, jarak antar titik, permintaan tiap *customer*, kapasitas kendaraan, waktu buka depot, waktu pelayanan tiap item, kecepatan rata-rata kendaraan dan parameter-parameter algoritma ACO. Data tersebut di atas akan diproses sesuai dengan tahapan algoritma ACO yaitu identifikasi masalah, identifikasi parameter-parameter algoritma ACO, inialisasi dengan iterasi ACO dan batasan kapasitas kendaraan serta *time window* depot, *update pheromone lokal*, *update pheromone global* dan mengecek kriteria pemberhentian. Output dari proses tersebut adalah hasil rute dari masing-masing semut pada iterasi, total jarak

tempuh dan total waktu tempuh yang terpilih berupa teks tulisan serta urutan rute berupa visualisasi gambar.

Berikut ini adalah penyelesaian permasalahan distribusi tersebut dengan menggunakan program.



Gambar 3 : Tampilan hasil perhitungan program

Berdasarkan Gambar 3 tersebut, diperoleh solusi rute jarak sebagai berikut:

1. Iterasi 1
 - Semut ke 1 :
 - Semut ke 2 :
 - Semut ke 3 :
 Dari masing-masing semut tersebut, rute yang dipilih pada semut ke 1 iterasi ke 1, dengan total jarak 38 km

2. Iterasi 2
 - Semut ke 1 :
 - Semut ke 2 :
 - Semut ke 3 :
 Dari masing-masing semut tersebut, rute yang dipilih pada semut ke 3 iterasi ke 2, dengan total jarak 44 km

Sehingga dari 2 iterasi, hasil rute terpendek yaitu pada semut ke 1 pada iterasi 1 dengan total waktu 3,07199990272522 jam. Pada gambar 3 juga dijelaskan gambar rute yang diperoleh pada permasalahan tersebut.

Program Ant Colony Optimization (ACO) ini telah diuji coba dengan menggunakan 6, 20, 50 dan 105 titik.

KESIMPULAN

Berdasarkan hasil dan pembahasan, berikut ini adalah beberapa kesimpulan yang bisa diberikan:

1. Pencarian solusi untuk permasalahan *Vehicle Routing Problem with Time Window* (VRPTW) menggunakan Algoritma *Ant Colony Optimization* dapat menggunakan pemrograman *Borland Delphi 7.0*. Inputan pada program berupa menggambar titik depot dan titik *customer*, kapasitas kendaraan, waktu buka depot, waktu pelayanan, kecepatan rata-rata kendaraan, jumlah semut, jumlah iterasi , , jarak serta permintaan setiap *customer*. Proses dimulai dengan menyeleksi *customer* sampai semua *customer* dikunjungi dengan menghitung nilai *pheromone*, kemudian mengupdate nilai *pheromone* dengan menggunakan *update pheromone lokal* dan *update pheromone global* untuk solusi terbaik serta dilakukan iterasi untuk pengoptimalan pencarian rute terbaik. Output yang ditampilkan pada program adalah berupa gambar solusi dan keterangan hasil rute serta total waktu yang dibutuhkan untuk melayani semua *customer*.
2. Dari simulasi program *Vehicle Routing Problem with Time Window* (VRPTW) menggunakan Algoritma *Ant Colony Optimization* terlihat bahwa jika parameter , , jumlah iterasi, dan jumlah semut berbeda maka rute dan total jarak yang terbentuk juga berbeda, selama nilai parameter untuk dan tetap pada interval yang ditentukan. Parameter , , dan merupakan parameter yang bersifat *random* (acak) sehingga mempengaruhi rute dan total jarak yang terbentuk. Pada Penyelesaian Algoritma ACO terdapat kaitan antara kapasitas kendaraan besar dan permintaan *customer* kecil maka penyesuaian depot untuk rute yang terbentuk lebih sedikit, sedangkan jika kapasitas kendaraan kecil dan permintaan *customer* besar maka penyesuaian depot untuk rute yang terbentuk lebih banyak sehingga total jarak yang dihasilkan lebih panjang. Jika banyaknya semut dan banyaknya iterasi besar kemungkinan untuk penelusuran rute untuk mencari kemungkinan solusi yang minimum lebih besar sehingga bisa mencapai solusi optimum. Jika *time window* (waktu buka depot) kurang dari total waktu maka penyelesaiannya maka ada *customer* yang tidak layak untuk dilayani sehingga harus menambah waktu pada *time window* (waktu buka depot) agar setiap *customer* bisa dilayani.

SARAN

Program yang telah dibuat ini dapat digunakan untuk menyelesaikan permasalahan VRPTW pada kegiatan Praktik Kerja Lapangan (PKL), karena telah diuji hingga 105 titik. Agar diperoleh rute dengan total jarak tempuh yang minimal, disarankan agar jumlah iterasi yang diinputkan lebih dari atau sama dengan faktorial dari jumlah *customer*.

Pada penelitian selanjutnya disarankan untuk menambahkan teknik-teknik yang dapat memberikan kestabilan hasil dalam setiap iterasi dan memperkuat nilai solusinya dengan mempertimbangkan tingkat kemampuan komputer atau dengan menambahkan *procedure* untuk menghitung kecepatan komputasi program dari masing-masing algoritma, sehingga dapat dibandingkan program dengan algoritma manakah yang memiliki waktu komputasi lebih cepat.

DAFTAR PUSTAKA

- Bell, E. John, & McMullen, R. Patrick. 2004. *Ant Colony Optimization Techniques for the Vehicle Routing Problem*. Department of Operational Sciences, Air Force Institute of Technology Wright-Patterson AFB, OH, USA. Elsevier Advanced Engineering Informatics 18 (2004) 41–48.
- Dorigo, M., & Gambardella. 1996. *Ant Colony System: A Cooperative learning Approach to the Traveling Salesman Problem*. Tech.Rep/IRIDIA/1996/-003, Universite Libre de Bruxelles, Belgium.
- Purwanto, 1998. *Matematika Diskrit*. Malang : Institut Keguruan dan Ilmu Pengetahuan Malang.
- Satyananda, Darmawan. 2012. *Panduan Praktikum Struktur Data*. Malang: Fakultas Matematika dan Ilmu Alam Universitas Negeri Malang.
- Socha, K., Dorigo, M., 2006. *Ant Colony Optimization for Countinous Domain*. European Journal of Operational Research.