

Ant Colony Optimization

Budi Santosa, PhD

Dosen Teknik Industri ITS, Surabaya

Lab Komputasi dan Optimasi Industri

Email : budi_s@ie.its.ac.id

1.1 Pendahuluan

Dalam dua dekade terakhir ini, banyak penelitian yang berhubungan dengan komputasi menggunakan perilaku alam sebagai dasar penelitiannya. Salah satunya adalah ant colony. Penelitian mengenai ant colony telah menyumbangkan pengetahuan yang sangat besar terhadap bidang bioinformation, yang merupakan observasi bagaimana suatu kejadian yang ada di alam dapat digunakan untuk mengatasi permasalahan yang ada pada dunia industri. **Ant-Colony Optimization (ACO)** termasuk dalam kelompok Swarm Intelligence, yang merupakan salah satu jenis pengembangan paradigma yang digunakan untuk menyelesaikan masalah optimasi dimana inspirasi yang digunakan untuk memecahkan masalah tersebut berasal dari perilaku kumpulan atau kawanan (swarm) serangga.

ACO biasanya digunakan untuk menyelesaikan discrete optimization problems dan persoalan yang kompleks dimana terdapat banyak variabel. Hasil yang diperoleh dengan menggunakan ACO, walaupun tidak optimal namun mendekati optimal. ACO sudah diterapkan di berbagai masalah seperti VRP, penjadwalan proyek dengan sumberdaya terbatas, data mining, penjadwalan pekerjaan (job scheduling) dan beberapa masalah kombinatorial ayang lain. Ant-based techniques pertama kali digunakan oleh Dorigo et. al Dorigo et al. [1996] dengan menggunakan ACO untuk menyelesaikan Traveling Salesman Problem (TSP). Dalam ACO, setiap semut dalam kawanan yang berjalan akan meninggalkan pheromone (semacam zat kimia) pada jalur yang dilaluinya. Pheromone ini menjadi semacam sinyal bagi sesama semut. Jalur yang pendek akan menyisakan sinyal yang lebih kuat. Semut berikutnya, pada saat memutuskan jalur mana yang harus dipilih,

biasanya akan cenderung memilih untuk mengikuti jalur dengan sinyal yang paling kuat, sehingga jalur terpendek akan ditemui karena lebih banyak semut yang akan melewati jalur tersebut. Semakin banyak semut yang lewat suatu jalur, semakin kuat sinyal di jalur itu. Hal ini dapat dilihat pada Gambar 8.1, dimana ditunjukkan bagaimana jalur yang dilalui oleh semut pada saat mencari makan dari sarang sampai ke sumber makanan.

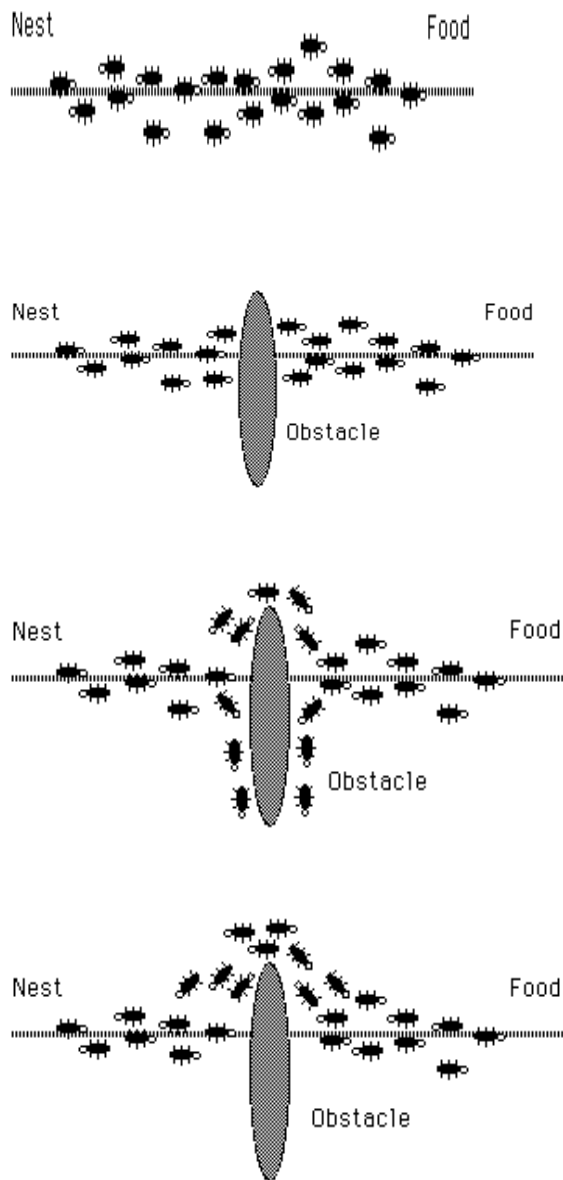
1.2 Konsep ACO

Konsep *Ant System* sendiri terinspirasi dari pengamatan terhadap tingkah laku semut. Secara sederhana, perilaku semut dapat digambarkan sebagai berikut. Semut merupakan serangga sosial yang hidup dalam koloni-koloni dan berperilaku berdasarkan kepentingan koloni. Salah satu hal yang menarik dari perilaku semut adalah kemampuannya dalam menemukan jarak terpendek antara sarang mereka dan sumber makanan. Misalkan ada segerombolan semut yang berada di depan harus memilih lintasan tertentu untuk dilewati. Pada saat semut pertama berjalan, semut tersebut meninggalkan hormon pheromone yang dapat dicium oleh semut berikutnya, sehingga semut-semut berikutnya tahu apakah tempat tersebut sudah dilewati atau belum. Dengan demikian, setiap semut yang berjalan akan meninggalkan pheromone pada jalur yang dilaluinya.

ACO meniru perilaku koloni semut dalam mencari sumber makanan bermula dan kembali ke sarangnya yang ternyata secara alami mencari rute terpendek

Semut yang melewati lintasan yang pendek akan meninggalkan aroma pheromone yang lebih tajam daripada yang menempuh lintasan yang lebih panjang. Hal ini terjadi karena pheromone yang ditinggalkan dapat menguap. Saat semut-semut yang di belakang mengikuti semut-semut yang di depannya, semut-semut tersebut akan memilih lintasan berdasarkan kuatnya aroma pheromone dan jarak lintasan. Semakin banyak semut yang menempuh suatu lintasan tertentu, maka aroma pheromone pada lintasan tersebut akan semakin kuat sehingga semut-semut berikutnya akan mengikuti lintasan yang sama, sehingga jalur terpendek akan ditemui karena lebih banyak semut yang akan melewati jalur tersebut. Gambar 8.1 menunjukkan bagaimana koloni semut berjalan dari sarang menuju sumber makanan.

Pada awalnya semut akan terdistribusi merata melalui dua lintasan. Setelah beberapa saat semut-semut mulai memilih lintasan terpendek karena pheromone yang ditinggalkan semut akan mengumpul di lintasan yang lebih pendek.



Gambar 8.1: Koloni semut akhirnya akan memilih lintasan terpendek menuju sumber makanan

Salah satu hal yang penting pada ant-inspired technologies adalah pada saat sistem telah menemukan solusi yang optimal, ACO akan dapat beradaptasi dengan cepat terhadap perubahan yang terjadi di sekitar. Adaptasi ini didasarkan pada pheromone yang merupakan dasar dari ant-system. Pheromone yang lebih kuat akan dimiliki oleh solusi dengan jalur yang lebih optimal pada akhir suatu algoritma.

Ant colony optimization (ACO) didasarkan pada perilaku bekerjasama dari koloni semut yang dapat menemukan lintasan terpendek dari sarangnya menuju ke sumber makanan.

Proses dalam ACO bisa dijelaskan sebagai berikut. Misalkan ada N semut dalam satu koloni. Semut-semut itu akan memulai gerakan dari sarang mereka menuju tujuan akhir melalui beberapa simpul dan berakhir pada simpul tujuan di akhir setiap siklus atau iterasi. Kalau semua semut sudah menyelesaikan lintasannya, jumlah pheromone pada lintasan terbaik secara global akan diperbarui. Lintasan global terbaik artinya terbaik diantara semua semut. Pada awal proses, yaitu pada iterasi 1, semua ruas dari simpul awal akan diberi jumlah pheromone yang sama.

Sehingga pada iterasi 1, semua semut akan mulai dari simpul awal dan berakhir pada simpul tujuan dengan cara memilih simpul-simpul antara secara random. Proses optimasi berakhir jika jumlah iterasi maksimum sudah tercapai atau tidak ada lagi solusi yang lebih baik yang bisa didapat dalam beberapa iterasi yang berturut-turut.

1.3 Perilaku Semut

Seekor semut k pada simpul i akan memilih simpul j yang dituju pada layer berikutnya dengan probabilitas

$$p_{i,j} = \begin{cases} \frac{\tau_{i,j}^\alpha}{\sum_{j \in N_i^{(k)}} \tau_{i,j}^\alpha}, & \text{jika } j \in N_i^{(k)} \\ 0, & \text{jika } j \notin N_i^{(k)} \end{cases} \quad (8.1)$$

dimana α menunjukkan derajat kepentingan pheromone dan $N_i^{(k)}$ adalah pilihan yang dipunyai semut k (neighborhood) pada saat ia berada pada simpul i . Neighbor-

hood dari semut k pada simpul i akan mengandung semua simpul yang bisa dituju yang tersambung secara langsung ke simpul i , kecuali simpul yang sudah dikunjungi sebelumnya.

1.4 Penambahan dan Penguapan Pheromone

Seekor semut k ketika melewati ruas akan meninggalkan pheromone. Jumlah pheromone yang terdapat pada ruas ij setelah dilewati semut k diberikan dengan rumus

$$\tau_{i,j} \leftarrow \tau_{i,j} + \Delta\tau^k \quad (8.2)$$

Dengan meningkatnya nilai pheromone pada ruas $i-j$, maka kemungkinan ruas ini akan dipilih lagi pada iterasi berikutnya semakin besar. Setelah sejumlah simpul dilewati maka akan terjadi penguapan pheromone dengan aturan sebagai berikut

$$\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j}, j; \forall(i,j) \in A \quad (8.3)$$

dimana $\rho \in (0, 1]$ adalah parameter tingkat penguapan dan A menyatakan segmen atau ruas yang sudah dilalui oleh semut k sebagai bagian dari lintasan dari sarangnya menuju ke makanan. Penurunan jumlah pheromone memungkinkan semut untuk mengeksplorasi lintasan yang berbeda selama proses pencarian. Ini juga akan menghilangkan kemungkinan memilih lintasan yang kurang bagus. Selain itu, ini juga membantu membatasi nilai maksimum yang dicapai oleh suatu lintasan pheromone. Jumlah pheromone yang ditambahkan pada ruas $i - j$ oleh semut k diberikan sebagai

$$\Delta\tau_{i,j}^{(k)} = \frac{Q}{L_k} \quad (8.4)$$

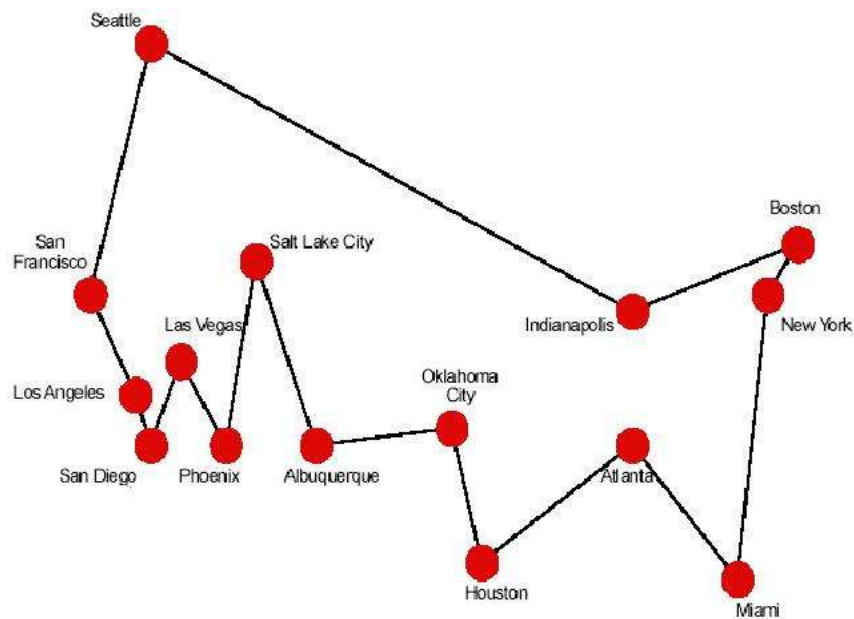
dimana Q adalah konstanta dan L_k adalah lintasan terpendek yang dilalui semut k (dalam kasus TSP adalah lintasan yang dilalui oleh salesman). Nilai Q biasanya ditentukan oleh user. Atau bisa juga diimplementasikan dengan cara berikut

$$\Delta\tau_{i,j}^{(k)} \begin{cases} \frac{Q}{L_k}, & \text{jika } (i,j) \in \text{lintasan terbaik global} \\ 0, & \text{untuk yang lain} \end{cases} \quad (8.5)$$

1.6 ACO untuk Traveling Salesman Problem

8.6. ACO UNTUK TRAVELING SALESMAN PROBLEM

Awalnya ACO diterapkan oleh penemunya untuk menyelesaikan persoalan Traveling Salesman Problem (TSP), seperti dijelaskan dalam sub bab 15. Pada bagian ini akan dijelaskan bagaimana ACO diterapkan pada Traveling Salesman Problem (TSP) berdasarkan Marco Dorigo [Dorigo et al., 1996], yang merupakan penemu algoritma ini. Dalam TSP, jika terdapat n kota akan ada $(n \times (n - 1) / 2)$ buah ruas, dan juga memiliki $(n - 1)!$ rute yang mungkin. Jarak yang digunakan dalam TSP standar merupakan jarak simetris, artinya jarak antara kota r ke kota s sama dengan jarak antara kota s ke kota r $d(r, s) = d(s, r)$.



Gambar 8.2, Ilustrasi TSP

Dalam TSP, setiap semut memulai rutennya melalui sebuah kota awal yang sama yaitu 1 dan akan kembali lagi ke kota 1. Secara berulang, setiap semut akan memilih kota berikutnya untuk dikunjungi sehingga setiap semut akan menemukan rute yang dilaluinya hingga kembali ke kota 1 lagi. Pemilihan kota-kota yang akan dilaluinya didasarkan pada suatu fungsi probabilitas, dengan mempertimbangkan visibility (invers dari jarak) kota tersebut dan jumlah pheromone yang terdapat pada ruas yang menghubungkan kota r ke kota s . probabilitas bergerak dari kota r ke kota s akan membesar jika jarak kota r ke kota s lebih kecil. Semut-semut akan memilih untuk bergerak menuju ke kota-kota yang dihubungkan dengan ruas yang pendek dan atau memiliki tingkat pheromone yang tinggi [Dorigo and Gambardella, 1997].

8.6. ACO UNTUK TRAVELING SALESMAN PROBLEM

Perlu diketahui bahwa setiap semut memiliki sebuah memori, dinamai tabu list, yang berisi semua kota yang telah dikunjunginya pada setiap tur. Tabu list ini mencegah semut untuk mengunjungi kota-kota yang sebelumnya telah dikunjungi selama rute tersebut berlangsung, sehingga tidak akan ada satu kota dikunjungi lebih dari sekali. Setelah satu iterasi selesai atau semua semut menyelesaikan rutenya, akan dilakukan updating pheromone secara global pada setiap semut. Penguapan pheromone terjadi pada semua ruas yang dilalui dan untuk setiap semut dihitung panjang rute yang telah ditempuh. Semakin pendek sebuah rute yang dihasilkan oleh seekor semut, semakin besar jumlah pheromone yang diletakkan pada ruas-ruas yang dilaluinya. Dengan kata lain, ruas-ruas yang merupakan bagian dari rute yang pendek adalah ruas-ruas yang menerima jumlah pheromone yang lebih besar. Hal ini menyebabkan ruas-ruas yang diberi pheromone lebih banyak akan lebih disukai pada iterasi selanjutnya sesudah semut sampai ke sarang dan kembali ke sumber makanan. Dan juga, jalur terpendek yang ditemukan oleh ants disimpan dan semua tabu list yang ada dikosongkan kembali.

Peranan utama dari penguapan pheromone tadi adalah untuk mencegah stagnasi, yaitu situasi dimana semua semut berakhir dengan melakukan rute yang sama. Proses di atas kemudian diulangi sampai rute-rute yang dilakukan mencapai jumlah iterasi maksimum yang ditentukan user) atau tidak terjadi lagi perubahan solusi. Aturan transisi status yang digunakan oleh Ant System dinamai random-roportional rule, yang ditunjukkan oleh persamaan 8.9,

$$p_k(r, s) = \begin{cases} \tau(r, s)\eta(r, s)^\beta / \sum_{u \in M_k} \tau(r, u)\eta(r, u)^\beta, & \text{jika } s \notin M_k \\ 0, & \text{lainnya} \end{cases} \quad (8.9)$$

dimana $p_k(r, s)$ adalah peluang pindah dari kota r ke kota s dari semut k , τ adalah tingkat pheromone, $\eta(r, s)$ kedekatan kota r dan s dan β adalah bobot yang mengontrol visibility terhadap tingkat pheromone τ . Pertama perlu dilakukan penghitungan matrik yang menyatakan kedekatan antar kota (visibility), $\eta(r, s)$, dimana setiap entrinya merupakan $1/\text{jarak}$ dari r ke s . Untuk setiap semut perlu dilakukan penempatan awal di kota mana akan memulai rute. Untuk TSP, setiap semut akan mulai dari kota 1. Dari persamaan 8.9 ini ruas yang lebih pendek dan memiliki jumlah pheromone yang lebih besar akan lebih besar probabilitasnya untuk dipilih dalam rute.

Tingkat pheromone (pheromone trail) berubah baik secara lokal dan global. Pembaruan pheromone secara global dimaksudkan untuk memberikan ruas-ruas yang tepat dalam rute terpendek. Setelah satu iterasi, semua semut

8.6. ACO UNTUK TRAVELING SALESMAN PROBLEM

menyelesaikan satu rute, setelah semua semut kembali ke sarangnya akan terjadi updating pheromone dengan aturan

$$\tau_{r,s} \leftarrow (1 - \rho)\tau_{r,s} + \sum_{k=1}^N \Delta\tau_{r,s}^k \quad (8.10)$$

dimana $\rho \in (0, 1]$ adalah tingkat penguapan atau sering juga disebut faktor *decay* dan $\Delta\tau(k)$ *ij* adalah jumlah pheromone yang ditambahkan pada ruas $r-s$ oleh semut terbaik k . Tujuan dari update pheromone adalah meningkatkan nilai pheromone yang berkaitan dengan lintasan yang menjanjikan atau bagus.

Setelah semut-semut *artificial* menyelesaikan satu rute mereka maka semut terbaik akan menambahkan pheromone pada ruas-ruas yang sudah dikunjungi; yaitu ruas-ruas yang masuk dalam rutenya. Sementara ruas yang lain tidak berubah tingkat pheromonenya. Jumlah pheromone $\Delta\tau(r, s)$ yang ditambahkan dalam setiap ruas yang dikunjungi (r, s) oleh semut terbaik berbanding terbalik dengan jarak total rute: semakin pendek rute semakin tinggi jumlah pheromone yang ditambahkan pada ruas-ruasnya. cara ini memang dimaksudkan untuk meniru bagaimana pheromone ini diakumulasi di dalam kejadian nyata dimana dipengaruhi oleh panjang lintasan dan kontinuitas waktu semut menjalani lintasan. Pembaruan global diformulasikan sebagai $\tau(r, s) \leftarrow (1 - \rho)\tau(r, s) + \Delta\tau(r, s)$, dimana $\Delta(r, s) = (\text{rute} - \text{terpendek})^{-1}$ atau $1/L_k$, dimana L_k adalah lintasan terpendek yang ditempuh semut k .

Aturan pembaruan pheromone lokal ini berguna untuk mencegah agar ruas-ruas yang bagus tidak dipilih oleh semua semut: setiap kali suatu ruas dipilih oleh seekor semut jumlah pheromonena berubah dengan menerapkan aturan pembaruan lokal: $\tau(r, s) \leftarrow (1 - \rho)\tau(r, s) + \Delta\tau(r, s)$, dimana τ_0 adalah suatu parameter. Pembaruan lokal ini juga dipengaruhi oleh penguapan seperti yang terjadi pada semut nyata. Dengan kata lain, pengaruh dari pembaruan lokal ini adalah untuk membuat tingkat kemenarikan ruas-ruas yang ada berubah secara dinamis: setiap kali seekor semut menggunakan sebuah ruas maka ruas ini dengan segera akan berkurang tingkat ketertarikannya (karena pada ruas tersebut jumlah pheromone berkurang akibat penguapan). Ini secara tidak langsung akan membuat semut yang lain akan memilih ruas-ruas lain yang belum dikunjungi. Sehingga semut-semut tidak akan memiliki kecenderungan untuk memilih jalur yang sama. Dengan cara membuat rute yang berbeda maka akan terdapat kemungkinan yang lebih tinggi untuk mendapatkan salah satu rute yang lebih baik daripada jika

8.6. ACO UNTUK TRAVELING SALESMAN PROBLEM

mereka semua menempuh rute yang sama. Algoritma ACO untuk TSP diberikan dalam Algoritma 12 berikut.

ACO telah diterapkan diberbagai bidang, diantaranya ACO untuk *vehicle routing problem* (VRP) [Bell and McMullen, 2004], capacitated vehicle routing problem (CVRP) [Mazzeo and Loiseau, 2004], penjadwalan proyek dengan resource ter-

Algorithm 12: Algoritma ACO untuk TSP

Input: jumlah semut, matrik jarak, itmax

Output: rute terbaik, total jarak

% Tahap Inisialisasi

foreach $i=1$:jumlah semut **do**

$\tau(i, j) = \tau_0$

 Hitung nilai visibility $h(i, j) = 1/\text{jarak}(i, j)$

end

while $it < itmax$ **do**

 Tempatkan semua semut pada kota 1

foreach $i=1$:jumlah semut **do**

 Mulailah dari kota r

 Hitung tingkat pheromone untuk setiap ruas yang terhubung dari r

 Hitung probabilitas transisi dari kota r ke kota lain

 Bangkitkan bilangan random

 Gunakan roulette wheel selection untuk memilih kota berikutnya

end

foreach $i=1$:jumlah semut **do**

 Hitung jarak rute untuk setiap semut

 Update nilai τ menggunakan persamaan (8.10)

% Untuk $\Delta\tau$ bisa digunakan $1/f$ dari setiap semut

% dimana f adalah panjang rute dikurangi min panjang rute

end

end

8.6. ACO UNTUK TRAVELING SALESMAN PROBLEM

batas (RCPSP) [Merkle et al., 2002]. ACO juga diterapkan dalam data mining seperti klustering [Lee et al., 2008].

1.7 Lampiran Matlab Code untuk ACO-TSP

```
function [bestroute,mincost]=acs_budi(d,iter,n_ants);
%input:
%d - matrik jarak ukuran n x n
%iter -jumlah iterasi
%n_ants- jumlah semut
m=n_ants;%jumlah semut
n=length(d);%jumlah kota.
e=.5;%evaporation coefficient.
alpha=1;%pangkat untuk ants' sight.
beta=3;%pangkat untuk trace's effect.
for i=1:n%generating sight matrix.
    for j=1:n
        if d(i,j)==0
            h(i,j)=0;
        else
            h(i,j)=1/d(i,j);%inverse distance
        end
    end
end
end
t=0.01*ones(n);%tho awal,
el=.96;%coefficient of common cost elimination.
for i=1:iter

    for i=1:m
        app(i,1)=1;%semua semut mulai dari kota 1
    end
    %rute semut
    for i=1:m %untuk semua semut
        mh=h;%matriks invers jarak
        for j=1:n-1 %simpul berikutnya
            c=app(i,j); %memilih satu kota
            mh(:,c)=0; %jika sdh dipilih maka inv distance=0
            temp=(t(c,:).^beta).*(mh(c,:).^alpha)
```

8.7. LAMPIRAN

```
%menghitung tho pheromone
s=(sum(temp))%jumlah tho
p=(1/s).*temp;%probabilitas
r=rand;
s=0;
for k=1:n %banyaknya kota
    s=s+p(k);
    if r<=s
        app(i,j+1)=k;
        %penempatan semut i di simpul berikutnya
        break
    end
end
end
end
at=app;% hasil rute
at=horzcat(at,at(:,1));
% diakhir rute kembali ke kota 1
% menghitung ongkos
for i=1:m
    s=0;
    for j=1:n
        s=s+d(at(i,j),at(i,j+1));
        %menghitung jarak rute utk semua semut
    end
    f(i)=s;
end
cost=f;
f=f-el*min(f);%eliminasi ongkos .
%update ruas dari rute terbaik
for i=1:m
    for j=1:n
        dt=1/f(i);%inverse jarak total
        t(at(i,j),at(i,j+1))=(1-e)*t(at(i,j),at(i,j+1))+dt;
        %updating tau, dimana rute terbaik otomatis
        %mendapat tambahan pheromone terbesar.
    end
end
end
[mincost(i),number]=min(cost);
bestroute(i,:)=at(number,:);
iteration(i)=i;
end
bestroute=bestroute(end,:);
```

8.7. *LAMPIRAN*
