

DAFTAR PUSTAKA

- Alfriany, L. N. (2008). Implementasi Algoritma Ant Untuk Menentukan Jalur Terpendek Dalam Proses Pengiriman Surat Pos Dengan Menggunakan Visual Basic 6.0". *Jurnal Mitra Tahun XIV*, 2
- Budayasa, I. K. (2007). *Teori Graf dan Aplikasinya*. Surabaya: Unesa University Press.
- Cahyono, B. (2016). Penggunaan Software Matrix Laboratory (Matlab) Dalam Pembelajaran Aljabar Linier. *Phenomenon : Jurnal Pendidikan MIPA*, 3(1), 45–62. <https://doi.org/10.21580/phen.2013.3.1.174>
- Davendra, D. (2010). *Travelling Salesman Problem: Theory and Applications*. Rijeka: InTech.
- Delima. (2020). *Konsep dasar algoritma pemrograman dan bahasa pemrograman*.
- Dian. (2013). Algoritma Optimasi Untuk Penyelesaian Travelling Salesman Problem (Optimization Algorithm For Solving Travelling Salesman Problem). *Jurnal Transformatika. Teknologi Informasi Fakultas Teknologi Informasi dan Komunikasi, Universitas Semarang*, 11(1)
- Fikri, M. Shahabudin (2010). Studi dan Impementasi Mengenal Algoritma Semut. *Jurnal Teknik Informatika dan Komputer*, 3
- Helene, I. et al. (2008). Penyelesaian Masalah Travelling Salesman Problem Menggunakan Ant Colony System. *Jurnal Media Informatika*, 6
- Hotang, C. B. (2016). Implementasi Algoritma Ant Colony Optimization (ACO) Pada Masalah Pendistribusian Barang di Kantor Pos. Universitas Sriwijaya, Palembang.

- Irfan, M. (2018). Penyelesaian Travelling Salesman Problem (TSP) Menggunakan Algoritma Hill Climbing dan MATLAB. *Matematika*, 17(1), 13–20. <https://doi.org/10.29313/jmtm.v17i1.3090>
- Kusrini, & Istiyanto, J. E. (2007). Penyelesaian Travelling Salesman Problem Dengan Algoritma Cheapest Insertion Heuristics Dan Basis Data. *Jurnal Informatika*, 8(2), 109-114–114. <https://doi.org/10.9744/informatika.8.2.pp.109-114>
- Leksono, A. (2009). *Ant colony optimization*. Universitas Diponegoro.
- Mangkunegara, A. P. (2001). *Manajemen Sumber Daya Perusahaan*. Bandung: PT. Remaja Rosdakarya
- Munir, R. (2007). *Matematika Diskrit*. Bandung: Informatika ITB
- Megalina, Y. (2010). Universitas Sumatera Utara Poliklinik Universitas Sumatera Utara. In *Universitas Sumatera Utara*.
- Mutakhirroh, I., et al. (2007). Pencarian Jalur Terpendek Menggunakan Algoritma Semut. *Seminar Nasional*. Yogyakarta: FTI UII. Hlmn B81-B85.
- Noviansyah, M. (2019). *Pengenalan Dasar Matlab*. Universitas Bina Sarana Informatika.
- Nurlaelasari, E., Supriyadi, S., & Lenggana, U. T. (2018). Penerapan Algoritma Ant Colony Optimization Menentukan Nilai Optimal Dalam Memilih Objek Wisata Berbasis Android. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 9(1), 287–298. <https://doi.org/10.24176/simet.v9i1.1914>
- Paillin, D. B., & Tupan, J. M. (2018). Pemecahan Traveling Salesman Problem Menggunakan Teknik Branch and Bound dan Cheapest Insertion Heuristic

- (Studi Kasus : PT. Paris Jaya Mandiri – Ambon). *Seminar Dan Konferensi Nasional IDEC, 2014*, 7–8.
- Permana, A. Y., & Romadlon, P. (2019). Perancangan Sistem Informasi Penjualan Perumahan Menggunakan Metode SDLC Pada PT. Mandiri Land Prosperous Berbasis Mobile. *SIGMA – Jurnal Teknologi Pelita Bangsa*, 10 (Volume 10 Nomor 2 Desember 2019), 153–167.
- Refianti. R. (2005). Solusi Optimal Travelling Salesman Problem dengan Algoritma Ant Colony System. *Journal of Informatics and Computer*
- Robitoh. (2018). Perancangan Sistem Administrasi Pada Puskesmas Sarolangun Berbasis Web. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699.
- Sanggala, E., Dimiyati, T. T., & Yogaswara, Y. (2021). Genetic Algorithm Untuk Memperbaiki Rute Travelling Salesman. *Jurnal Logistik Bisnis*, 11(2), 14–18.
- Santosa, B. (1992). *Tutorial on Ant Colony Optimization*.
- Septima, U (2008), Implementasi Algoritma Ant Colony System untuk Penyelesaian Travelling Salesman Problem. *Jurnal Percikan*, 92
- Siang, J. J. (2009). *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer*. Edisi keempat. Yogyakarta
- Tisen, M. (2013). *Penentuan Rute Optimal Pengantaran Koran Menggunakan Travelling Salesman Problem*. 27–32.
- Tjiptono, F. dan Chandra, G. (2008). *Pemasaran Strategik*. Yogyakarta: Andi Offset.
- Wardy, I. S. (2007). Penggunaan graph dalam algoritma semut untuk melakukan

optimisasi, Program studi Teknik Informatika, ITB, Bandung.

Wiyanti, D. T. (2013). Algoritma Optimasi Untuk Penyelesaian Travelling
Salesman Problem. *Jurnal Transformatika*, 11(1), 1.
<https://doi.org/10.26623/transformatika.v11i1.76>

LAMPIRAN

Matlab Code

```
function [besttour,mincost]=aco(d,iter,n_ants)
tic
%d is the n x n distance matrix.
%iter is the number of iterations.
%n_ants is the number of ants.

m=n_ants;%number of ants.
n=length(d);%number of nodes.

alpha=2;%order of ants' sight effect.
beta=1;%order of trace's effect.
gamma=0.1;%evaporation coefficient.

%% ### Generate visibility matrix ###

h=zeros(n);
for i=1:n
    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

rute=zeros(m,n); % rute for m ants
thoinit=1000*ones(n);%initial tho.
fprintf('iter,distance\n');
for x=1:iter
    %% ### Generate ants route ###

    for i=1:m %ants initial placing.
        rute(i,1)=1;%ants start from node 1
    end

    for i=1:m %for all ants
        mh=h; %copy visibility matrix
        for j=1:n %next node
            c=rute(i,j); %current city, choosing the next node
            mh(:,c)=0; %invalidate visited node
            tho=(mh(c,:).^alpha).*(thoinit(c,:).^beta); %calculate
pheromone level
            s=(sum(tho)); %sum of tho
            p=(1/s).*tho; %probability marix of (1,n) dimension

            r=rand; %random value between 0 and 1
            s=0;
            for k=1:n %number of nodes
```

```

        s=s+p(k);%calculate cumulative number of node
        if r<=s
            rute(i,j+1)=k; %choose node k as the next stop
            break
        end
    end
end
end

% make route cyclic
rute_c=horzcat(rute,rute(:,1)); %add first city to the end of
ant's route

%% ### Compute Distances and Keep The Best Route ###

%compute total distance of path traversed by each ant
f=zeros(m,1);
for i=1:m %number of ant
    s=0;
    for j=1:n
        s=s+d(rute_c(i,j),rute_c(i,j+1));
    end
    f(i)=s;
end

[minf,idk]=min(f);
fprintf('%d,%f\n',x, minf);
if x == 1 || minf < best_distance
    % update best route on iteration 1 or if we get a better
distance
    best_distance=minf;
    best_iteration=x;
    best=rute_c(idk,:);
end

%% ### Update Pheromone ###

% Evaporate thoinit
thoinit=(1-gamma)*thoinit;
for i=1:m % for all ant
    for j=1:n % for all nodes
        dt=1/f(i);%inverse total distance, delta pheromone

thoinit(rute_c(i,j),rute_c(i,j+1))=thoinit(rute_c(i,j),rute_c(i,j+
1))+dt; %updating traces
    end
end
end
disp('The best route is first found on iteration:');
disp(best_iteration);
besttour=best;
mincost=best_distance;
toc

```