

## Lampiran 1 Data Hasil Produksi Perusahaan Filter Rokok

Produk	Hasil Produksi Tahun Juli 2016 – Juni 2017 (batang)					
	Juli	Ags	Sep	Okt	Nov	Des
Menthol	71.300.000	118.900.000	122.900.000	87.300.000	108.200.000	151.700.000
Dual Shape	84.200.000	64.600.000	79.300.000	66.100.000	74.900.000	67.800.000
ROA	507.200.000	535.400.000	657.900.000	479.200.000	420.700.000	641.000.000
Cavitek	1.500.000	2.700.000	100.000	2.500.000	0	1.500.000
CPA/CPS	64.000.000	76.000.000	142.600.000	94.100.000	19.700.000	35.500.000
NWA	146.300.000	110.100.000	195.900.000	266.000.000	133.300.000	224.300.000

Produk	Hasil Produksi Tahun Juli 2016 – Juni 2017 (batang)					
	Jan	Feb	Mar	Apr	Mei	Jun
Menthol	56.400.000	104.900.000	132.700.000	135.600.000	148.000.000	94.800.000
Dual Shape	36.900.000	56.400.000	99.900.000	112.300.000	84.400.000	57.200.000
ROA	475.800.000	453.900.000	714.900.000	742.000.000	600.400.000	674.700.000
Cavitek	1.900.000	3.000.000	38.200.000	49.500.000	68.600.000	70.500.000
CPA/CPS	21.200.000	30.000.000	42.200.000	36.000.000	66.800.000	15.300.000
NWA	136.400.000	175.500.000	291.400.000	221.500.000	162.700.000	253.700.000

Produk	Hasil Produksi Tahun Juli 2016 – Juni 2017 (tray, 1 tray= 40.000 batang)											
	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
Menthol	1782,5	2972,5	3072,5	2182,5	2705	3792,5	1410	2622,5	3317,5	3390	3700	2370
Dual Shape	2105	1615	1982,5	1652,5	1872,5	1695	922,5	1410	2497,5	2807,5	2110	1430
ROA	12680	13385	16447,5	11980	10517,5	16025	11895	11347,5	17872,5	18550	15010	16867,5
Cavitek	37,5	67,5	2,5	62,5	0	37,5	47,5	75	955	1237,5	1715	1762,5
CPA/CPS	1600	1900	3565	2352,5	492,5	887,5	530	750	1055	900	1670	382,5
NWA	3657,5	2752,5	4897,5	6650	3332,5	5607,5	3410	4387,5	7285	5537,5	4067,5	6342,5

Produk	Hasil Produksi Tahun Juli 2015 – Juni 2016 (tray/hari/shift)											
	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun
Menthol	23	25	20	18	24	31	18	24	23	22	20	30
Dual Shape	21	23	16	22	24	23	12	19	18	22	29	18
ROA	27	26	21	24	20	24	17	26	29	17	21	19
Cavitek	1	1	1	1	0	1	1	1	11	14	23	22
CPA/CPS	20	17	16	21	7	12	7	10	12	10	23	5
NWA	16	19	23	26	22	25	23	17	21	22	25	14

## Lampiran 2 Koding *Simulated Annealing* pada MATLAB

### 1. Program Utama

```

clear all
close all
clc

%% INPUT

% TMI=xlsread('data SA.xlsx',1);
% Konstanta=xlsread('data SA.xlsx',2);
% Presedence=xlsread('data SA.xlsx',3);

TMIload=load('TMI.mat');
TMIread=TMIload.TMI;
TMIProduct=TMIread;
Konstantaload=load('Konstanta.mat');
Konstanta=Konstantaload.Konstanta;
PresedenceSWAPload=load('PresedenceSWAP.mat');
PresedenceSWAP=PresedenceSWAPload.Presedence;
PresedenceTRANSFERload=load('PresedenceTRANSFER.mat');
PresedenceTRANSFER=PresedenceTRANSFERload.Presedence;
Constrainload=load('Constrain.mat');
Constrain=Constrainload.Constrain;
JMLPRODUK=6;

%input manual
SP=1;
Tabulis=[1,2,3,39];

% dari excel
C=Konstanta(1);
Cr=Konstanta(2);
qm1=Konstanta(3);
qm2=Konstanta(4);
qm3=Konstanta(5);
qm4=Konstanta(6);
qm5=Konstanta(7);
qm6=Konstanta(8);
qm=[qm1,qm2,qm3,qm4,qm5,qm6];
%T=Konstanta(9);
Tin=100;
MAXP=Konstanta(10);

[Mtmi,Ntmi]=size(TMIread);
TMI=struct('Kelompok',[],'Data',[],'Mean',[],'Meantotal',[]);
TMI(1).Kelompok=TMIread(1:5,1);
TMI(2).Kelompok=TMIread(6:8,1);

```

```
TMI (3) .Kelompok=TMIread(9:10,1);
TMI (4) .Kelompok=TMIread(11:12,1);
TMI (5) .Kelompok=TMIread(13:15,1);
TMI (6) .Kelompok=TMIread(16:19,1);
TMI (7) .Kelompok=TMIread(20:26,1);
TMI (8) .Kelompok=TMIread(27:36,1);
TMI (9) .Kelompok=TMIread(37:39,1);
```

```
TMI (1) .Data=TMIread(1:5,2:Ntmi-1);
TMI (2) .Data=TMIread(6:8,2:Ntmi-1);
TMI (3) .Data=TMIread(9:10,2:Ntmi-1);
TMI (4) .Data=TMIread(11:12,2:Ntmi-1);
TMI (5) .Data=TMIread(13:15,2:Ntmi-1);
TMI (6) .Data=TMIread(16:19,2:Ntmi-1);
TMI (7) .Data=TMIread(20:26,2:Ntmi-1);
TMI (8) .Data=TMIread(27:36,2:Ntmi-1);
TMI (9) .Data=TMIread(37:39,2:Ntmi-1);
```

```
TMI (1) .Mean=TMIread(1:5,Ntmi);
TMI (2) .Mean=TMIread(6:8,Ntmi);
TMI (3) .Mean=TMIread(9:10,Ntmi);
TMI (4) .Mean=TMIread(11:12,Ntmi);
TMI (5) .Mean=TMIread(13:15,Ntmi);
TMI (6) .Mean=TMIread(16:19,Ntmi);
TMI (7) .Mean=TMIread(20:26,Ntmi);
TMI (8) .Mean=TMIread(27:36,Ntmi);
TMI (9) .Mean=TMIread(37:39,Ntmi);
```

```
TMI (1) .Meantotal=sum(TMIread(1:5,Ntmi));
TMI (2) .Meantotal=sum(TMIread(6:8,Ntmi));
TMI (3) .Meantotal=sum(TMIread(9:10,Ntmi));
TMI (4) .Meantotal=sum(TMIread(11:12,Ntmi));
TMI (5) .Meantotal=sum(TMIread(13:15,Ntmi));
TMI (6) .Meantotal=sum(TMIread(16:19,Ntmi));
TMI (7) .Meantotal=sum(TMIread(20:26,Ntmi));
TMI (8) .Meantotal=sum(TMIread(27:36,Ntmi));
TMI (9) .Meantotal=sum(TMIread(37:39,Ntmi));
```

```
%%
```

```
=====
=====
```

```
%
```

```
=====
=====
```

```
% STAGE 1 STAGE 1 STAGE 1 STAGE 1 STAGE 1 STAGE 1 STAGE 1
STAGE 1 STAGE 1
```

```
%
```

```
=====
=====
```

```

%
=====
=====

disp('=== SA STAGE 1 == SA STAGE 1 == SA STAGE 1 == SA STAGE
1 ===')

%% BDT
[~,M]=size(TMI(1).Data);
K=length(TMI);

for m=1:M
    for ws=1:K
        BDT(ws,m)=(qm(m)*(SP*C-sum(TMI(ws).Data(:,m))));
        IDT(ws,m)=(C-sum(TMI(ws).Data(:,m)))*qm(m);
    end
end
%BDT=sum(BDTcount,2);
%BDT(1)=sum(BDTcount2);

%% Simulated Annealing STAGE 1

TMIiter=struct('tmi',TMI);
T=Tin;
k=2;
while T>10
    for s=1:39
        r=randsp(T);
        TMIsebelum=TMIiter(k-1).tmi;

        [TMIBaru,K,keterangan,WS,PRODUK]=swaptransferfcn(r,TMIsebelum
        ,IDT,K,M,Tabulis,PresedenceTRANSFER,PresedenceSWAP,C,Constrai
        n);

        TMIsiklus=struct('TS',[]);
        TMIsiklus(s).TS=TMIBaru;
        disp(keterangan)
        % HITUNG WAKTU STASIUN KERJA BARU
        for i=1:K
            MEANTOTALGET(i,1)=TMIBaru(i).Meantotal;
        end
        MEANTOTALBARU=sum(MEANTOTALGET);
        [~,M]=size(TMIBaru(1).Data);

        for m=1:M
            for ws=1:K
                BDTbaru(ws,m)=(qm(m)*(SP*C-
sum(TMIBaru(ws).Data(:,m))));
                IDT(ws,m)=(C-
sum(TMIBaru(ws).Data(:,m)))*qm(m);
            end
        end
    end
end

```

```

end

if BDTbaru(WS, PRODUK) < BDT(WS, PRODUK)
    BDT=BDTbaru;
    TMIiter(k).tmi=TMIBaru;
    %disp('TMIiter terupdate')
else
    diff=( (BDTbaru(WS, PRODUK) -
BDT(WS, PRODUK)) /BDTbaru(WS, PRODUK)) *100;
    R=rand(1);
    if R<exp(-diff/T);
        BDT=BDTbaru;
        TMIiter(k).tmi=TMIBaru;
        %disp('TMIiter terupdate')
    else
        TMIiter(k).tmi=TMIsbelum;
        %disp('TMIiter tidak terupdate')
    end
end
end
k=k+1;
end
T=T*Cr;
end

%%
=====
=====
%
=====
=====
% STAGE 2 STAGE 2 STAGE 2 STAGE 2 STAGE 2 STAGE 2 STAGE 2
STAGE 2 STAGE 2
%
=====
=====
%
=====
=====
%%
disp('=== SA STAGE 2 == SA STAGE 2 == SA STAGE 2 == SA STAGE
2 ===')
TMI=TMIiter(end).tmi;

for m=1:M
    for ws=1:K
        IDT(ws,m)=(C-sum(TMI(ws).Data(:,m))).*qm(m);
        Skcount(ws,m)=IDT(ws,m)*qm(m);
    end
end
end
Sk=sum(Skcount(:));

```

```

Sim=sum(IDT(:));
IDTperSimminseperK=(IDT/Sim - 1/K).^2;
sumIDTperSimminseperK_times_qm=sum(IDTperSimminseperK,1)*qm';
BWS=sumIDTperSimminseperK_times_qm*(K/(K-1));

for m=1:M
    for ws=1:K
        qmSkmpersk(ws,m)=(qm(m)*IDT(ws,m)/Sk) - (1/M)^2;
    end
end
BS=(M/(K*(M-1)))*sum(qmSkmpersk(:));

BSawal=BS;
BWSawal=BWS;

%% Simulated Annealing STAGE 2

TMIiter2=struct('tmi',TMI);
T=Tin;
k=2;

BSiter(1)=BS;
BWSiter(1)=BWS;
while T>10
    for s=1:39
        r=randsp2(T);
        TMIsebelum=TMIiter2(k-1).tmi;

[TMiBaru,K,keterangan,WS,PRODUK]=swaptransferfcn(r,TMIsebelum
, IDT,K,M,Tabulis,PresedenceTRANSFER,PresedenceSWAP,C,Constrai
n);

        TMIsiklus=struct('TS',[]);
        TMIsiklus(s).TS=TMiBaru;
        disp(keterangan)
        % HITUNG WAKTU STASIUN KERJA BARU
        for i=1:K
            MEANTOTALGET(i,1)=TMiBaru(i).Meantotal;
        end
        MEANTOTALBARU=sum(MEANTOTALGET);
        [~,M]=size(TMiBaru(1).Data);
        BSsebelum=BS;
        BWSsebelum=BWS;

%-----
%-----
        for m=1:M
            for ws=1:K
                IDT(ws,m)=(C-
sum(TMiBaru(ws).Data(:,m)))*qm(m);
                Skcount(ws,m)=IDT(ws,m)*qm(m);
            end
        end
end

```

```

Sk=sum(Skcount(:));
Sim=sum(IDT(:));
IDTperSimminseperK=(IDT/Sim - 1/K).^2;

sumIDTperSimminseperK_times_qm=sum(IDTperSimminseperK,1)*qm';
BWSbaru=sumIDTperSimminseperK_times_qm*(K/(K-1));

for m=1:M
    for ws=1:K
        qmSkmpersk(ws,m)=(qm(m)*IDT(ws,m)/Sk) -
(1/M))^2;
    end
end
BSbaru=(M/(K*(M-1)))*sum(qmSkmpersk(:));

%-----
-----

if BSbaru<BS && BWSbaru<BWS
    BS=BSbaru;
    BWS=BWSbaru;
    BSiter(k)=BSbaru;
    BWSiter(k)=BWSbaru;
    TMIiter2(k).tmi=TMIBaru;
    %disp('TMIiter terupdate')
else
    diff=((BSbaru-BS)+(BWSbaru-
BWS))/(BSbaru+BWSbaru)*100;
    R=rand(1);
    if R<exp(-diff/T);
        BS=BSbaru;
        BWS=BWSbaru;
        BSiter(k)=BSbaru;
        BWSiter(k)=BWSbaru;
        TMIiter2(k).tmi=TMIBaru;
        %disp('TMIiter terupdate')
    else
        TMIiter2(k).tmi=TMIsbelum;
        BS=BSsbelum;
        BWS=BWSsbelum;
        BSiter(k)=BSsbelum;
        BWSiter(k)=BWSsbelum;
        %disp('TMIiter tidak terupdate')
    end
end

%-----
-----

end
k=k+1;
end
T=T*Cr;

```

end

```
TMIFinal=TMIter2(end).tmi
```

```
format long
```

```
disp('====Sebelum SA====')  
disp(BSawal);  
disp(BWSawal);
```

```
disp('====Sesudah SA====')  
disp(BSbaru);  
disp(BWSbaru);
```

```
format short
```

```
subplot(2,1,1)  
plot(BSiter)  
xlabel('iterasi');  
ylabel('BS');
```

```
subplot(2,1,2)  
plot(BWSiter)  
xlabel('iterasi');  
ylabel('BWS');
```



## 2. Fungsi Tukar dan Pindah Aktivitas

```

function
[TMIbaru,K,keterangan,WS,PRODUK]=swaptransferfcn(r,TMI,IDT,K,
M,Tabulis,PresedenceTRANSFER,PresedenceSWAP,C,Constrain)
TMIbaru=[];
%keterangan='no process';
if r<0.25
    %=====
    %== SWAP SWAP SWAP SWAP SWAP SWAP SWAP SWAP SWAP ==
    %=====
    [devIDT,IDTmeank]=idtfcn(IDT,M,K);
    ws1=randi(K,1);

    [ws1,SP,kelompokws1,TMIws1sp,Meanws1,kelompokws1del,TMIws1spdel,
TMIws1del,Meanws1del]=evaluasiwaktuproduk1(ws1,devIDT,TMI,
Tabulis,K);
    Szx=IDT(ws1,SP);
    SZpM=IDTmeank(ws1);
    WS=ws1;
    PRODUK=SP;
    %assignin('base','Szx',Szx)
    %assignin('base','SZpM',SZpM)

    %=====
    %=====

    if Szx> SZpM
        saws1=(find(TMIws1spdel==min(TMIws1spdel)));
%selected activity dari ws1
        aktivitas1=kelompokws1del(saws1);
        ws2=randi(K,1);
        while ws2==ws1
            ws2=randi(K,1);
        end

        [ws2,TMIws2sp,kelompokws2,Meanws2,kelompokws2del,TMIws2spdel,
TMIws2del,Meanws2del]=evaluasiwaktuproduk2(ws2,ws1,SP,devIDT,TM
I,Tabulis,K);
        saws2=(find(TMIws2spdel==max(TMIws2spdel)));
        aktivitas2=kelompokws2del(saws2);

        %assignin('base','aktivitas1',aktivitas1);
        %assignin('base','aktivitas2',aktivitas2);
        %assignin('base','saws2',saws2)
        %assignin('base','TMIws2spdel',TMIws2spdel);

    %=====
    %=====
    %=====
else

```

```

%=====
=====
===
    saws1=(find(TMIws1spdel==max(TMIws1spdel)));
%selected activity dari ws1
    aktivitas1=kelompokws1del(saws1);
    ws2=randi(K,1);
    while ws2==ws1
        ws2=randi(K,1);
    end

[ws2,TMIws2sp, kelompokws2, Meanws2, kelompokws2del, TMIws2spdel,
TMIwsdel, Meanwsdel]=evaluasiwaktuproduk2(ws2,ws1,SP,devIDT,TM
I,Tabulis,K);
    saws2buf=(find(TMIws2spdel>0));
    TMIws2spdel=TMIws2spdel(saws2buf);
    saws2=(find(TMIws2spdel==min(TMIws2spdel)));
    aktivitas2=kelompokws2del(saws2);
end

Pre=PresedenceSWAP(aktivitas1,aktivitas2);
if Pre==1
    keterangan='Presedence - no swap';
    TMIbaru=TMI;
elseif Pre==0
    keterangan='Precedence for Swap OK';
    TMI1mean=TMI(ws1).Mean;
    TMI2mean=TMI(ws2).Mean;

    Swapadd1=find(kelompokws1==kelompokws1del(saws1));
    Swapadd2=find(kelompokws2==kelompokws2del(saws2));
    %assignin('base','ws1',ws1);
    %assignin('base','ws2',ws2);

    assignin('base','kelompokws2',kelompokws2)
    assignin('base','kelompokws2del',kelompokws2del)
    TMI1meannew=TMI1mean;
    TMI1meannew(Swapadd1)=TMI2mean(Swapadd2); %matriks
Mean baru yg sudah diswap

    TMI2meannew=TMI2mean;
    TMI2meannew(Swapadd2)=TMI1mean(Swapadd1); %matriks
Mean baru yg sudah diswap
    assignin('base','TMI1meannew',TMI1meannew)
    assignin('base','TMI2meannew',TMI2meannew)
    sumTMI1meannew=sum(TMI1meannew); %mean total dari TMI
pada ws1 dijumlah
    sumTMI2meannew=sum(TMI2meannew); %mean total dari TMI
pada ws1 dijumlah

```

```

        if sumTMI1meannew>C || sumTMI2meannew>C %kapasitas
tidak memenuhi
            keterangan='Capacity - no swap';
            TMIBaru=TMI;
        else
            keterangan='Capacity for Swap OK';
            TMI1kelompok=TMI (ws1) .Kelompok;
            TMI2kelompok=TMI (ws2) .Kelompok;

            TMI1kelompoknew=TMI1kelompok;
            TMI1kelompoknew (Swapadd1)=TMI2kelompok (Swapadd2);

            TMI2kelompoknew=TMI2kelompok;
            TMI2kelompoknew (Swapadd2)=TMI1kelompok (Swapadd1);

            %Constrain
            for i=1:length(TMI1kelompoknew)
                for j=1:length(TMI1kelompoknew)

ZCws1cek(i,j)=Constrain(TMI1kelompoknew(i),TMI1kelompoknew(j)
);

                    end
                end

                for i=1:length(TMI2kelompoknew)
                    for j=1:length(TMI2kelompoknew)

ZCws2cek(i,j)=Constrain(TMI2kelompoknew(i),TMI2kelompoknew(j)
);

                            end
                        end

            ZCws1cekfinal=sum(ZCws1cek(:));
            ZCws2cekfinal=sum(ZCws2cek(:));
            if ZCws1cekfinal~=0 || ZCws2cekfinal~=0
                keterangan='Zoning Constrain - no swap';
                TMIBaru=TMI;
            else
                assignin('base','Swapadd1',Swapadd1)
                assignin('base','Swapadd2',Swapadd2)

                assignin('base','ws1',ws1)
                assignin('base','ws2',ws2)
                BUF1_Kelompok=TMI (ws1) .Kelompok (Swapadd1);
                BUF1_Data=TMI (ws1) .Data (Swapadd1,:);
                BUF1_Mean=TMI (ws1) .Mean (Swapadd1);

                BUF2_Kelompok=TMI (ws2) .Kelompok (Swapadd2);
                BUF2_Data=TMI (ws2) .Data (Swapadd2,:);
            end
        end
    end
end

```

```

BUF2_Mean=TMI (ws2) .Mean (Swapadd2) ;

% SWAP
TMI (ws1) .Kelompok (Swapadd1)=BUF2_Kelompok;
TMI (ws1) .Data (Swapadd1, :)=BUF2_Data;
TMI (ws1) .Mean (Swapadd1)=BUF2_Mean;
TMI (ws1) .Meantotal=sum (TMI (ws1) .Mean) ;

TMI (ws2) .Kelompok (Swapadd2)=BUF1_Kelompok;
TMI (ws2) .Data (Swapadd2, :)=BUF1_Data;
TMI (ws2) .Mean (Swapadd2)=BUF1_Mean;
TMI (ws2) .Meantotal=sum (TMI (ws2) .Mean) ;
assignin ('base', 'TMIterswap', TMI)
TMIterswap=TMI;
TMIBaru=TMIterswap;
keterangan=strcat ('SWAP at[' , ...
    num2str (ws1) , ' ] to [' , num2str (ws2) , ' ] --
activity [' , ...
    num2str (TMI (ws1) .Kelompok (Swapadd1)) , ' ]
to [' , num2str (TMI (ws2) .Kelompok (Swapadd2)) , ' ]' ) ;
    end
end
end
else
%=====
%== TRANSFER TRANSFER TRANSFER TRANSFER TRANSFER =====
%=====
[devIDT, IDTmeank]=idtfcn (IDT, M, K) ;
% Workstation ke-k
ws1=randi (K, 1) ;
%disp (strcat ('ws1=', num2str (ws1)))

[ws1, SP, kelompokws1, TMIws1sp, Meanws1, kelompokws1del, TMIws1spd
el, TMIws1del, Meanws1del]=evaluasiwaktuproduk1 (ws1, devIDT, TMI,
Tabulis, K) ;
%disp (strcat ('ws1 updated =', num2str (ws1)))
if isempty (TMI (ws1) .Kelompok) ==1
    TMI (ws1) =[] ;
    K=K-1 ;
    ws1=randi (K, 1) ;

[ws1, SP, kelompokws1, TMIws1sp, Meanws1, kelompokws1del, TMIws1spd
el, TMIws1del, Meanws1del]=evaluasiwaktuproduk1 (ws1, devIDT, TMI,
Tabulis, K) ;
end

Szx=IDT (ws1, SP) ;
SZpM=IDTmeank (ws1) ;

```

```

=====
if Szx> SZpM
    WS=ws1;
    PRODUK=SP;
    Cekws=1:K;
    Cekws(ws1)=[];
    TRFfindall=[];
    for i=1:length(Cekws)
        cekmax=[];
        cekmaxempty=[];
        TMIcekdata=TMI(Cekws(i)).Data;
        kelompokcek=TMI(Cekws(i)).Kelompok;
        TMIcek=TMIcekdata(:,SP);
        for j=1:length(TMIcek)
            cekmax(j,1)=(TMIcek(j)==max(TMIcekdata(j,:)));
        end
        cekmaxempty=find(cekmax==1);
        if isempty(cekmaxempty)==0
            TRFfind=kelompokcek(cekmaxempty);
            TRFfindall=[TRFfindall;TRFfind];
        end
    end

    %TRFfindall=tabulisignorTRF(Tabulis,TRFfindall);

[TRFfindall,TMIcekdel]=tabulisignorTRF(Tabulis,TRFfindall,TMI
cek);
    aktivitasTRF=max(TRFfindall);

    cariws=[];
    for i=1:K;
        assignin('base','TMI_i_Kelompok',TMI(i).Kelompok);
        assignin('base','aktivitasTRF',aktivitasTRF);
        cariws=find(TMI(i).Kelompok==aktivitasTRF);
        if isempty(cariws)~=1
            WSORIGIN=i;
            break;
        end
    end

    %TMI=struct('Kelompok',[],'Data',[],'Mean',[],'Meantotal',[])
;

    TMIwsKELOMPOKnew=TMI(ws1).Kelompok;
    TMIwsDATAnew=TMI(ws1).Data;
    TMIwsMEANnew=TMI(ws1).Mean;

    Aadd=find(TMI(WSORIGIN).Kelompok==aktivitasTRF);

```

```

TMIwsKELOMPOKnew=[TMIwsKELOMPOKnew;aktivitasTRF];
TMIwsDATAnew=[TMIwsDATAnew;TMI(WSORIGIN).Data(Aadd,:)];
TMIwsMEANnew=[TMIwsMEANnew;TMI(WSORIGIN).Mean(Aadd,:)];
TMIwsMEANTOTALnew=sum(TMIwsMEANnew);

for c=1:length(TMI(ws1).Kelompok)

cekPre(c,1)=PresedenceTRANSFER(aktivitasTRF,TMI(ws1).Kelompok
(c));
end
PreTRF=sum(cekPre==0); %banyaknya precedence bernilai 0
dalam ws1

if PreTRF==0
    keterangan='Presedence - no transfer';
    TMiBaru=TMI;
elseif PreTRF~=0
    if TMIwsMEANTOTALnew>C %kapasitas tidak memenuhi
        keterangan='Capacity - no transfer';
        TMiBaru=TMI;
    else
        for i=1:length(TMI(ws1).Kelompok)

ZCcek(i,1)=Constrain(aktivitasTRF,TMI(ws1).Kelompok(i));
end
ZCcekfinal=sum(ZCcek(:));
if ZCcekfinal~=0
    keterangan='Zoning Constrain - no
transfer';

    TMiBaru=TMI;
else
    keterangan=strcat('TRANSFER at[',...
        num2str(WSORIGIN),'] to [' ,num2str(ws1))
;

    TMI(ws1).Kelompok=TMIwsKELOMPOKnew;
    TMI(ws1).Data=TMIwsDATAnew;
    TMI(ws1).Mean=TMIwsMEANnew;
    TMI(ws1).Meantotal=TMIwsMEANTOTALnew;

    TMI(WSORIGIN).Kelompok(Aadd,:)=[];
    TMI(WSORIGIN).Data(Aadd,:)=[];
    TMI(WSORIGIN).Mean(Aadd,:)=[];

TMI(WSORIGIN).Meantotal=sum(TMI(WSORIGIN).Mean);
TMiBaru=TMI;

end
end
end
end

```

```

%=====
=====
===
else
%=====
=====
=====
===
    % ws1 = WS ORIGIN
    TMIcekdata=TMI (ws1) .Data;
    TMIcek=TMIcekdata (:, SP);
    kelompokcek=TMI (ws1) .Kelompok;

[kelompokcek, TMIcekdel]=tabulisignorTRF (Tabulis, kelompokcek, T
MIcek);
    aktivitasTRF=kelompokcek (find (TMIcekdel==max (TMIcekdel) ));

    devIDTcek=devIDT (:, SP);
    devIDTcek (ws1, :)=[];
    IDTmeankcek=IDTmeank;
    IDTmeankcek (ws1, :)=[];
    WScek=1:K;
    WScek (ws1)=[];
    ceklowfind=[];
    for i=1:K-1
        ceklow (i, 1)=(devIDTcek (i)<IDTmeankcek (i) );
    end
    ceklowfind=find (ceklow==1);

    if isempty (ceklowfind)==0

WSESTINATION=WScek (ceklowfind (randi (length (ceklowfind), 1) ));
end

    TMIwsKELOMPOKnew=TMI (WSESTINATION) .Kelompok;
    TMIwsDATAnew=TMI (WSESTINATION) .Data;
    TMIwsMEANnew=TMI (WSESTINATION) .Mean;
    WS=WSESTINATION;
    PRODUK=SP;
    Aadd=find (TMI (ws1) .Kelompok==aktivitasTRF);

    TMIwsKELOMPOKnew=[TMIwsKELOMPOKnew;aktivitasTRF];
    TMIwsDATAnew=[TMIwsDATAnew;TMI (ws1) .Data (Aadd, :)];
    TMIwsMEANnew=[TMIwsMEANnew;TMI (ws1) .Mean (Aadd, :)];
    TMIwsMEANTOTALnew=sum (TMIwsMEANnew);

    % ===== PECEDEENCE - CAPACITY - ZONING CONSTRAIN
=====
    % paling tidak harus ada sepasang aktivitas dalam
WSESTINATION yang
    % Presedence-nya sama dengan 0

```

```

for c=1:length(TMI(WSDESTINATION).Kelompok)
    assignin('base','c',c)

cekPre(c,1)=PrecedenceTRANSFER(aktivitasTRF,TMI(WSDESTINATION)
).Kelompok(c));
    end
    PreTRF=sum(cekPre==0); %banyaknya precedence bernilai 0
dalam WSDESTINATION

    if PreTRF==0 %tidak ada sepasang aktivitas bernilai 0
        keterangan='Presedence - no transfer';
        TMIBaru=TMI;
    elseif PreTRF~=0
        if TMIwsMEANTOTALnew>C %kapasitas tidak memenuhi
            keterangan='Capacity - no transfer';
            TMIBaru=TMI;
        else
            for i=1:length(TMI(WSDESTINATION).Kelompok)

ZCcek(i,1)=Constrain(aktivitasTRF,TMI(WSDESTINATION).Kelompok
(i));
                end
                ZCcekfinal=sum(ZCcek(:));
                if ZCcekfinal~=0
                    keterangan='Zoning Constrain - no transfer';
                    TMIBaru=TMI;
                else
                    TMI(WSDESTINATION).Kelompok=TMIwsKELOMPOKnew;
                    TMI(WSDESTINATION).Data=TMIwsDATANew;
                    TMI(WSDESTINATION).Mean=TMIwsMEANnew;

TMI(WSDESTINATION).Meantotal=TMIwsMEANTOTALnew;
                    TMI(ws1).Kelompok(Aadd,:)=[];
                    TMI(ws1).Data(Aadd,:)=[];
                    TMI(ws1).Mean(Aadd,:)=[];
                    TMI(ws1).Meantotal=sum(TMI(ws1).Mean);
                    TMIBaru=TMI;
                    keterangan=strcat('TRANSFER at[',...
                        num2str(ws1),'] to
[' ,num2str(WSDESTINATION)) ;
                        end
                    end
                end
            end
        end
    end
end

```



### 3. Fungsi Mencari Nilai Waktu *Idle*

```
function [devIDT, IDTmeank]=idtfcn (IDT, M, K)

IDTmeank=mean (IDT, 2) ;
for m=1:M
    for k=1:K
        devIDT (k, m)=abs (IDT (k, m) -IDTmeank (k)) ;
    end
end
end
```

### 4. Fungsi Nilai Random Spesial Tahap 1

```
function [r]=randsp (T)

%if T>19 R>=0.25
if T>19
r = 0.25 + (1-0.25).*rand(1);

else
%if T<=19 R<0.25
r = 0.001 + (0.249-0.001).*rand(1);
end
end
```

### 5. Fungsi Nilai Random Spesial Tahap 2

```
function [r]=randsp2 (T)

%if T>19 R>=0.25
if T>19

r = 0.001 + (0.249-0.001).*rand(1);
else
%if T<=19 R<0.25

r = 0.25 + (1-0.25).*rand(1);
end
end
```

## 6. Fungsi Menghitung Waktu Aktivitas Terpilih Pertama

```

function
[ws,SP, kelompokws, TMIwssp, Meanws, kelompokwsdel, TMIwsspdel, TMI
wsdel, Meanwsdel]=evaluasiwaktuproduk1(ws, devIDT, TMI, Tabulis, K
)

[~, SP]=find(devIDT==max(devIDT(ws, :))); %SP = selected
product

TMIwssp=TMI(ws).Data(:, SP); %time
TMIws=TMI(ws).Data;
kelompokws=TMI(ws).Kelompok; %aktivitas
Meanws=TMI(ws).Mean;

% abaikan yang tabulis
%[TMIwsspdel, kelompokdel]=tabulisignor(Tabulis, kelompokws, TMI
wssp, TMIws, Meanws);
[kelompokwsdel, TMIwsspdel, TMIwsdel, Meanwsdel]=tabulisignor(Ta
bulis, kelompokws, TMIwssp, TMIws, Meanws);

s0find=[];
s0find=(find(TMIwsspdel~=0));
while isempty(s0find)==1
    %reselect ws ws2
    ws=randi(K, 1);

    [~, SP]=find(devIDT==max(devIDT(ws, :)));

    TMIwssp=TMI(ws).Data(:, SP); %time
    TMIws=TMI(ws).Data;
    kelompokws=TMI(ws).Kelompok; %aktivitas
    Meanws=TMI(ws).Mean;

    % abaikan yang tabulis

[kelompokwsdel, TMIwsspdel, TMIwsdel, Meanwsdel]=tabulisignor(Ta
bulis, kelompokws, TMIwssp, TMIws, Meanws);

    s0find=[];
    s0find=(find(TMIwsspdel~=0));
end

```

## 7. Fungsi Menghitung Waktu Aktivitas Terpilih 2

```

function
[ws2, TMIwssp, kelompokws, Meanws, kelompokwsdel, TMIwsspdel, TMIws
del, Meanwsdel]=evaluasiwaktuproduk2 (ws2, ws1, SP, devIDT, TMI, Tab
ulis, K)

TMIwssp=TMI (ws2) .Data (:, SP); %time
TMIws=TMI (ws2) .Data;
kelompokws=TMI (ws2) .Kelompok; %aktivitas
Meanws=TMI (ws2) .Mean;
% abaikan yang tabulis
%[TMIwsspdel, kelompokwsdel]=tabulisignor (Tabulis, kelompokws, T
MIwssp);
[kelompokwsdel, TMIwsspdel, TMIwsdel, Meanwsdel]=tabulisignor (Ta
bulis, kelompokws, TMIwssp, TMIws, Meanws);
assignin ('base', 'TMIwsspdel', TMIwsspdel)
assignin ('base', 'WS2', ws2)
assignin ('base', 'SP', SP)
s0find=[];
s0find=(find (TMIwsspdel~=0));
while isempty (s0find)==1
    %reselect ws ws2
    Knew=1:K;
    Knew (find (Knew==ws2))=[];
    ws2=Knew (randi (1));
    while ws2==ws1
        ws2=randi (K, 1);
    end

    [~, SP]=find (devIDT==max (devIDT (ws2, :)));

    TMIwssp=TMI (ws2) .Data (:, SP); %time
    TMIws=TMI (ws2) .Data;
    kelompokws=TMI (ws2) .Kelompok; %aktivitas
    Meanws=TMI (ws2) .Mean;

    % abaikan yang tabulis

[kelompokwsdel, TMIwsspdel, TMIwsdel, Meanwsdel]=tabulisignor (Ta
bulis, kelompokws, TMIwssp, TMIws, Meanws);

    s0find=[];
    s0find=(find (TMIwsspdel~=0));
end

```

8. Fungsi Eliminasi Aktivitas yang ada dalam *taboo list* untuk Penukaran Aktivitas

```

function
[kelompokdel, TMIwsspdel, TMIwsdel, Meanwsdel]=tabulisignor(Tabu
lis, kelompokws, TMIwssp, TMIws, Meanws) %wssp= workstation
selected product
E=[];
% abaikan yang tabulis
cek=1;
for i=1:length(kelompokws)
    for j=1:length(Tabulis)
        if kelompokws(i)==Tabulis(j)
            E(cek)=i;
            cek=cek+1;
        end
    end
end
end

if isempty(E)==0
    kelompokdel=kelompokws;
    kelompokdel(E)=[];

    TMIwsspdel=TMIwssp;
    TMIwsspdel(E)=[];

    TMIwsdel=TMIws;
    TMIwsdel(E,:)=[];

    Meanwsdel=Meanws;
    Meanwsdel(E)=[];
elseif isempty(E)~=0
    kelompokdel=kelompokws;
    TMIwsspdel=TMIwssp;
    TMIwsdel=TMIws;
    Meanwsdel=Meanws;
end

```

### 9. Fungsi Eliminasi Aktivitas yang ada dalam *taboo list* untuk Memindah Aktivitas

```
function
[kelompokdel, TMIcekdel]=tabulisignorTRF(Tabulis, kelompokws, TM
Icek) %wssp= workstation selected product
E=[];
% abaikan yang tabulis
cek=1;
for i=1:length(kelompokws)
    for j=1:length(Tabulis)
        if kelompokws(i)==Tabulis(j)
            E(cek)=i;
            cek=cek+1;
        end
    end
end

if isempty(E)==0
    kelompokdel=kelompokws;
    kelompokdel(E)=[];
    TMIcekdel=TMIcek;
    TMIcekdel(E)=[];
elseif isempty(E)~=0
    kelompokdel=kelompokws;
    TMIcekdel=TMIcek;
end
```





















































































