

Lampiran 6 : List Program Matlab Modifikasi dari program matlab (Rudhito, 2003)

```
(file. matrikinput.m)
A=[20 -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -
Inf -Inf -Inf ; -Inf 20 -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf
-Inf -Inf -Inf -Inf -Inf; 42 -Inf 120 -Inf -Inf -Inf -Inf -Inf -Inf
-Inf -Inf -Inf -Inf -Inf -Inf -Inf; -Inf 44 -Inf 28 -Inf -Inf -
Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf; 165 -Inf 143 -Inf
30 -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf; -Inf 74 -
Inf 58 -Inf 20 -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -
Inf; 199 -Inf 177 -Inf 64 -Inf 30 -Inf -Inf -Inf -Inf -Inf -Inf
-Inf -Inf; -Inf 96 -Inf 70 -Inf 42 -Inf 25 -Inf -Inf -Inf -Inf
-Inf -Inf -Inf; 232 -Inf 210 -Inf 97 -Inf 63 -Inf 30 -Inf -Inf
-Inf -Inf -Inf -Inf; 264 -Inf 242 -Inf 129 -Inf 95 -Inf 62 30 -
Inf -Inf -Inf -Inf -Inf; 298 -Inf 276 -Inf 163 -Inf 129 -Inf 96
64 15 -Inf -Inf -Inf -Inf -Inf; 315 -Inf 292 -Inf 180 -Inf 146 -Inf
112 81 32 5 -Inf -Inf -Inf -Inf; 323 124 300 98 188 60 154 53 120 89
40 13 15 -Inf -Inf; 341 142 318 116 206 78 172 71 138 107 58
31 33 15 -Inf; 360 161 337 135 225 97 191 90 157 126 77 50 52
34 20]
B = [3; 2; 25; 24; 148; 54; 182; 76; 215; 247; 281; 298; 306;
324; 343]
C = [-Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -Inf -
Inf -Inf -Inf 20]
x0 = [0; 1; -Inf; -Inf; -Inf; -Inf; -Inf; -Inf; -Inf; -Inf; -
Inf; -Inf; -Inf; -Inf; -Inf]
```

```
(file: maxio.m)
% Program Matlab Menghitung Lama Waktu Proses Tiap Mesin dan
Penyelesaian Akhir Produk Bakpia Pathok Jaya "25"
%Input Matriks A B C x0
matrikinput;
% u = barisan input
% output: x(k) = barisan keadaan sistem
% y(k) = barisan output sistem
% function io_SIMI = maxio
% Memasukkan input
disp(' ')
disp(' Menghitung Lama Waktu Proses Tiap Mesin dan
Penyelesaian Akhir Produk Bakpia Pathok Jaya "25" ')

disp('-----')
disp(' ')
u = input('Masukkan barisan input (dalam vektor kolom) u = ');
disp(' ')
q = length(u);
[a1, a2] = size(A);
L = zeros(a1,q);
M = zeros(1,q);
L(:,1)= x0;
% Menghitung x(1) = Ax(0) + Bu(1)
[x01, x02] = size(x0);
```

```

for i = 1 : a1
for j = 1 : x02
Ax0(i, j) = -Inf;
for p = 1: a2
Ax0(i, j) = max(Ax0(i, j) , A(i, p) + x0(p, j));
end;
end;
end;
x = max(Ax0, B+u(1));
% Menghitung y(1) = Cx(1)
[c1, c2] = size(C);
[x1, x2] = size(x);
for i = 1 : c1
for j = 1 : x2
Cx(i, j) = -Inf;
for p = 1: c2
Cx(i, j) = max(Cx(i, j) , C(i, p) + x(p, j));
end;
end;
end;
L(:,2)= x;
M(1,1)= Cx;
% Menghitung x(k+1) = Ax(k) + Bu(k+1) dan Menghitung y(k) =
Cx(k) utk k=1,2,...,p
% Menghitung x(k+1) = Ax(k) + Bu(k+1)
[a1, a2] = size(A);
[x1, x2] = size(x);
for r = 1 : q-1;
for i = 1 : a1
for j = 1 : x2
Ax(i, j) = -Inf;
for p = 1: a2
Ax(i, j) = max(Ax(i, j) , A(i, p) + x(p, j));
end;
end;
end;
x = max(Ax, B+u(r+1));
% Menghitung y(k) = Cx(k)
[c1, c2] = size(C);
[x1, x2] = size(x);
for i = 1 : c1
for j = 1 : x2
Cx(i, j) = -Inf;
for p = 1: c2
Cx(i, j) = max(Cx(i, j) , C(i, p) + x(p, j));
end;
end;
end;
L(:,r+2)= x;
M(1,r+1)= Cx;
end;
% Menampilkan hasil perhitungan
disp(' HASIL PERHITUNGAN :')
disp(' =====')
disp(' Matriks A = '),disp(A)

```

```

disp(' Matriks B = '),disp(B)
disp(' Matriks C = '),disp(C)
disp(' Kondisi awal x0 = '),disp(x0)
disp(' Barisan input u = '),disp(u')
disp(' Barisan vektor keadaan sistem x(k) utk k = 0,1, 2, ...
: '), disp(L)
disp(' Barisan output sistem y(k) utk k = 1, 2, ... : '),
disp(M)

```

(file. maxiopt.m)

```

% Program Matlab Menghitung OPTIMISASI INPUT-OUTPUT Sistem
Produksi Bakpia Pathok Jaya 25
%Input Matriks A B C x0
matrikinput;
% y = barisan output
% output:u_topi = barisan input paling lambat
% y_topi = barisan output untuk u_topi
% u_tilde = barisan input minimum simpangan
% y_tilde = barisan output untuk u_tilde
% function opt_input_output = optio
% Memasukkan input
disp(' ')
disp(' OPTIMASI INPUT-OUTPUT Sistem Produksi Bakpia Pathok
Jaya 25 ')
disp(' -----')
disp(' ')
y = input(' Masukkan barisan output (dalam vektor kolom) y =
');
disp(' ')
q = length(y);
% Menghitung C*B = CB
[c1, c2] = size(C);
[b1, b2] = size(B);
for i = 1 : c1
for j = 1 : b2
CB(i, j) = -Inf;
for p = 1: c2
CB(i, j) = max(CB(i, j) , C(i, p) + B(p, j));
end;
end;
end;
% Menghitung C*A = CA
[c1, c2] = size(C);
[a1, a2] = size(A);
for i = 1:c1
for j = 1: a2
CA(i, j) = -Inf;
for p = 1: c2
CA(i, j) = max(CA(i, j) , C(i, p) + A(p, j));
end;
end;
end;
L = zeros(q,a2);
L(1,:) = CA;

```

```

% Menghitung (C*A)*B
[ca1, ca2] = size(CA);
[b1, b2] = size(B);
for i = 1:ca1
for j = 1: b2
CAB(i, j) = -Inf;
for p = 1: ca2
CAB(i, j) = max(CAB(i, j) , CA(i, p) + B(p, j));
end;
end;
end;
% Menghitung A^k = Ak
[a1, a2]= size(A);
D = A;
for r = 1 : q-1
r+1;
for i = 1 : a1
for j = 1 : a2
Ak(i, j) = -Inf;
for p = 1: a2
Ak(i, j) = max(Ak(i, j) , A(i, p) + D(p, j));
end;
end;
end;
% Menghitung C*A^k = CAk
[c1, c2] = size(C);
[ak1, ak2] = size(Ak);
for i = 1 : c1
for j = 1: ak2
CAk(i, j) = -Inf;
for p = 1: c2
CAk(i, j) = max(CAk(i, j) , C(i, p) + Ak(p, j));
end;
end;
end;
L(r+1, :)=CAk;
% Menghitung CAkB
[cak1, cak2] = size(CAk);
[b1, b2] = size(B);
for i = 1:cak1
for j = 1: b2
CAkB(i, j) = -Inf;
for p = 1: cak2
CAkB(i, j) = max(CAkB(i, j) , CAk(i, p) + B(p, j));
end;
end;
end;
% Menyusun matriks H
for i = 1 : q
for j = 1 : q
if i < j
H(i,j) = -Inf;
end;
if i == j
H(i,j) = CB;

```

```

end;
if (i-j) ==1
H(i,j)= CAB;
end;
if (i-j) == r+1
H(i,j)= CAkB;
end;
if (i-j) > q
H(i,j)=[];
end;
end;
end;
D = Ak;
end;
% Menghitung K*x0
[l1, l2] = size(L);
[x01, x02] = size(x0);
for i = 1 : l1
for j = 1 : x02
Kx0(i, j) = -Inf;
for p = 1: l2
Kx0(i, j) = max(Kx0(i, j) , L(i, p) + x0(p, j));
end;
end;
end;
if max(Kx0 - y)<=0
% Menghitung input paling lambat u1 (H*(-y))
Ht=H';
my = -y;
[ht1, ht2] = size(Ht);
[my1, my2] = size(my);
for i = 1 : ht1
for j = 1 : my2
Htmy(i, j) = -Inf;
for p = 1: ht2
Htmy(i, j) = max(Htmy(i, j) , Ht(i, p) + my(p, j));
end;
end;
end;
u_topi = -Htmy;
% Mengitung H*u_topi
[h1, h2] = size(H);
[utp1, utp2] = size(u_topi);
for i = 1 : h1
for j = 1 : utp2
Hutp(i, j) = -Inf;
for p = 1: h2
Hutp(i, j) = max(Hutp(i, j) , H(i, p) + u_topi(p, j));
end;
end;
end;
Hutp;
% Menghitung barisan output y untuk u_topi
y_topi = max(Kx0, Hutp);
% Menghitung input minimum simpangan

```

```

delta = max(abs(y - y_topi));
u_tilde = u_topi + delta/2;
% Mengitung H*u_tilde
[h1, h2] = size(H);
[utd1, utd2] = size(u_tilde);
for i = 1 : h1
for j = 1 : utd2
Hutd(i, j) = -Inf;
for p = 1: h2
Hutd(i, j) = max(Hutd(i, j) , H(i, p) + u_tilde(p, j));
end;
end;
end;
Hutd;
% Menampilkan hasil perhitungan
disp(' HASIL PERHITUNGAN :')
disp(' =====')
disp('Matriks A = '),disp(A)
disp('Matriks B = '),disp(B)
disp('Matriks C = '),disp(C)
disp('Kondisi awal x0 = '),disp(x0)
disp('Barisan output y = '),disp(y)
disp('Barisan input paling lambat u_topi = '), disp((u_topi)')
disp('Barisan output y untuk u_topi = '), disp((y_topi)')
disp('Barisan input minimum simpangan u_tilde = '),
disp((u_tilde)')
disp('Barisan output y untuk u_tilde = '), disp((Hutd)')
else
disp('Input Minimum Simpangan tidak dapat dikerjakan (Kx0 >
y)')
end;

```

(file. Hu.m)

```

% Program Matlab Menghitung Hu
%Input Matriks A B C x0
matrikinput;
% u = barisan output
% output:H_u = barisan Hu
% function Hu = Hau
% Memasukkan input
disp(' ')
disp(' Menghitung Hu')
disp(' -----')
disp(' ')
u = input(' Masukkan barisan input (dalam vektor kolom) u =
');
disp(' ')
q = length(u);
% Menghitung C*B = CB
[c1, c2] = size(C);
[b1, b2] = size(B);
for i = 1 : c1
for j = 1 : b2
CB(i, j) = -Inf;

```

```

for p = 1: c2
CB(i, j) = max(CB(i, j) , C(i, p) + B(p, j));
end;
end;
end;
% Menghitung C*A = CA
[c1, c2] = size(C);
[a1, a2] = size(A);
for i = 1:c1
for j = 1: a2
CA(i, j) = -Inf;
for p = 1: c2
CA(i, j) = max(CA(i, j) , C(i, p) + A(p, j));
end;
end;
end;
L = zeros(q,a2);
L(1,:)= CA;
% Menghitung (C*A)*B
[ca1, ca2] = size(CA);
[b1, b2] = size(B);
for i = 1:ca1
for j = 1: b2
CAB(i, j) = -Inf;
for p = 1: ca2
CAB(i, j) = max(CAB(i, j) , CA(i, p) + B(p, j));
end;
end;
end;
% Menghitung A^k = Ak
[a1, a2]= size(A);
D = A;
for r = 1 : q-1
r+1;
for i = 1 : a1
for j = 1 : a2
Ak(i, j) = -Inf;
for p = 1: a2
Ak(i, j) = max(Ak(i, j) , A(i, p) + D(p, j));
end;
end;
end;
% Menghitung C*A^k = CAk
[c1, c2] = size(C);
[ak1, ak2] = size(Ak);
for i = 1 : c1
for j = 1: ak2
CAk(i, j) = -Inf;
for p = 1: c2
CAk(i, j) = max(CAk(i, j) , C(i, p) + Ak(p, j));
end;
end;
end;
L(r+1,:)=CAk;
% Menghitung CAkB

```

```

[cak1, cak2] = size(CAk);
[b1, b2] = size(B);
for i = 1:cak1
for j = 1: b2
CAkB(i, j) = -Inf;
for p = 1: cak2
CAkB(i, j) = max(CAkB(i, j) , CAk(i, p) + B(p, j));
end;
end;
end;
% Menyusun matriks H
for i = 1 : q
for j = 1 : q
if i < j
H(i,j) = -Inf;
end;
if i == j
H(i,j) = CB;
end;
if (i-j) ==1
H(i,j)= CAB;
end;
if (i-j) == r+1
H(i,j)= CAkB;
end;
if (i-j) > q
H(i,j)=[];
end;
end;
end;
D = Ak;
end;
% Mengitung H*u
[h1, h2] = size(H);
[u1, u2] = size(u);
for i = 1 : h1
for j = 1 : u2
Hu(i, j) = -Inf;
for p = 1: h2
Hu(i, j) = max(Hu(i, j) , H(i, p) + u(p, j));
end;
end;
end;
Hu;
% Menampilkan hasil perhitungan
disp(' HASIL PERHITUNGAN :')
disp(' =====')
disp('Barisan input u = '),disp(u)
disp('Barisan Hu = '), disp((Hu)')

                                (file kx0.m)
% Program Matlab Menghitung Kx0
%Input Matriks A B C x0
matrikinput;

```



```

% y = barisan output
% output:Kx0 = barisan input paling lambat
% function kx0 = kxnol
% Memasukkan input
disp(' ')
disp(' Menghitung Kx0')
disp(' -----')
disp(' ')
q = length(x0);
% Menghitung C*A = CA
[c1, c2] = size(C);
[a1, a2] = size(A);
for i = 1:c1
for j = 1: a2
CA(i, j) = -Inf;
for p = 1: c2
CA(i, j) = max(CA(i, j) , C(i, p) + A(p, j));
end;
end;
end;
L = zeros(q,a2);
L(1,:)= CA;
% Menghitung (C*A)*B
[ca1, ca2] = size(CA);
[b1, b2] = size(B);
for i = 1:ca1
for j = 1: b2
CAB(i, j) = -Inf;
for p = 1: ca2
CAB(i, j) = max(CAB(i, j) , CA(i, p) + B(p, j));
end;
end;
end;
% Menghitung A^k = Ak
[a1, a2]= size(A);
D = A;
for r = 1 : q-1
r+1;
for i = 1 : a1
for j = 1 : a2
Ak(i, j) = -Inf;
for p = 1: a2
Ak(i, j) = max(Ak(i, j) , A(i, p) + D(p, j));
end;
end;
end;
% Menghitung C*A^k = CAk
[c1, c2] = size(C);
[ak1, ak2] = size(Ak);
for i = 1 : c1
for j = 1: ak2
CAk(i, j) = -Inf;
for p = 1: c2
CAk(i, j) = max(CAk(i, j) , C(i, p) + Ak(p, j));
end;

```

```
end;
end;
L(r+1,:)=CAk;
D = Ak;
end;
% Menghitung K*x0
[l1, l2] = size(L);
[x01, x02] = size(x0);
for i = 1 : l1
for j = 1 : x02
Kx0(i, j) = -Inf;
for p = 1: l2
Kx0(i, j) = max(Kx0(i, j) , L(i, p) + x0(p, j));
end;
end;
end;
% Menampilkan hasil perhitungan
disp(' HASIL PERHITUNGAN :')
disp(' =====')
disp('Barisan Kx0 = '), disp((Kx0)')
```