DAFTAR PUSTAKA

Alman, Kusuma J., Amiruddin. (2013). Penyelesaian Numerik persamaan Adveksi-Difusi 2-D untuk Transfer Polutan dengan Menggunakan Metode Beda Hingga DuFort-Frankel. *ejournal Pascasarjana Unhas,* Makassar.

Biringen, S., & Chow, C. (2011). *An Introduction to Computational Fluid Mechanics by Example*, Hoboken, New Jersey: John Wiley & Sons, Inc.

Callister, W. D. (2001). *Fundamnetals of Materials Science and Engineering.* Department of Metallurgical Engineering The University of Utah.

Chern, I Liang. (2013). *Finite Difference Method for Solving Differential Equation*. Taiwan: Department of Mathematics National Taiwan University.

Cushman-Roisin, B., & Becker, J.M., (2009). *Introduction to Geophysical Fluid Dynamics: Physical and Numerical Aspects*. Academic Press.

Davidson, W. (2016). *Diffusive Gradients in Thin-Films for Environmental Measurements*. Cambridge University Press.

Fardiaz, S. (1992). *Polusi Air dan Udara.* Yogyakarta: Kanisius.

Gilberto E. Urroz. (2004). *Convergence, stability, and consistency of finite difference schemes in the solution of partial differential equations*. Diakses 25 September 2022. Available from : https://studylib.net/doc/10473146/convergence--stability--and-consistency-of-finite.

Hoffman & Chiang. (2000). *Computational Fluid Dynamics For Engineers, Volume 1*. USA: Wichita Kansas.

Hutomo, G.D., Kusuma, J., Zulfitri, W.F. (2019). Solusi Numerik Menggunakan Metode DuFort Frankel Pada Persamaan Adveksi-Difusi 2D dengan Koefisien Variabel. *Axiomath: Jurnal Matematika dan Aplikasinya*, 1(2), 6–13. doi:10.46918/axiomath.v1i2

Kusuma, J., Ribal, A., Mahie, A.G., Aris, N. (2017). On Pollution Distribution on Unhas Lake Using Two Dimension Advection-Diffusion Equation. *Far East Journal of Mathematical Science (FJMS),* 101(8), 1721–1729. doi:10.17654/MS101081721.

Kusuma, J., Khaeruddin., Toaha, S., Aris, N., Alman. (2014). Suatu Tinjauan Numerik Persamaan Adveksi Difusi 2-D Transfer Polutan dengan Menggunakan Metode Beda Hingga Du-Fort Frankel. *KNM XVII,* Surabaya.

Luknanto, Djoko. (1992). Angkutan Limbah. Universitas Gadjah Mada, Yogyakarta: Pusat Antar Universitas, Ilmu Teknik.

Morton, K.W., & Mayers, D.F. (2005). *Numerical Solution of Partial Differential Equaitons Second Edition.* New York: Cambridge University Press.

Mukhtasor. (2007). *Pencemaran Pesisir dan Laut*. Jakarta: PT. Pradnya Paramita.

Mustafa, A., Hasnawi., Tarunamulia., Selamat, M.B., Samawi, M.F. (2019). Distribusi Polutan Logam Berat di Perairan Pantai yang digunakan untuk Memasok Tambang Udang Terdekat dan Mitigasinya di Kecamatan Jabon Provinsi Jawa Timur. *Jurnal Riset Akuakultur*, 14(2), 127–138. doi:10.15578/jra.14.2.2019.127-138.

Noviyani, D., Yundari, Yudhi (2019). Solusi Persamaan Difusi Pada Larutan Gula dengan Metode Beda Hingga. *Buletin Ilmiah Math. Stat. dan Terapannya (Bimaster).* 8(3), 573-578, doi: 10.26418/bbimst.v8i3.34026.

Sampera, H., & Apriansyah. (2016). Aplikasi Metode Beda Hingga Carnk-Nicolson Implisit Untuk Menentukan Kasus Adveksi-Difusi 2D pada Sebaran Polutan di Suatu Perairan. *Prisma Fisika,* 4(2), 56-63. Doi: 10.26418/pf.v4i2.15851.

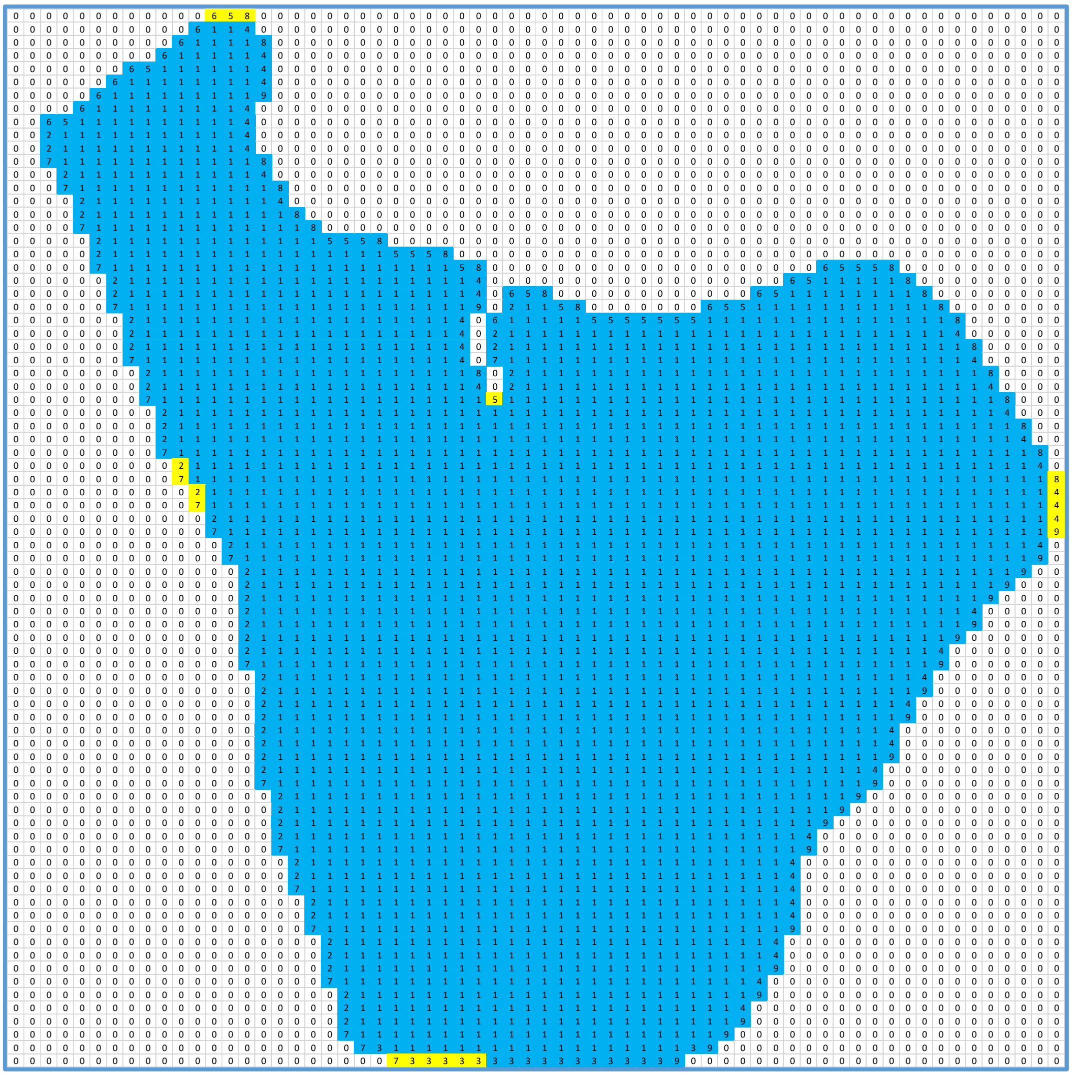
Taylor, M.E., (2011). *Partial Differential Equations I, Basic Theory, 2nd Edition*. USA: Springer.

Yaqin, K., Karim, Y., Fachruddin, L. (2018). Kualitas Air dan Kandungan Beberapa Logam di Danau Unhas. *Jurnal Pengelolaan Perairan*. 1(1), 1-13.

Zuhair. (2008). *Metode Numerik – Deret Taylor dan Deret MacLaurin*. Jakarta: Universitas Mercubuana.

LAMPIRAN

Lampiran 1. File Excel grid komputasi danau unhas yang berisikan kode dari 1 sampai 9



Lampiran 2. Program MATLAB penyebaran logam berat pada danau unhas dengan menggunakan persamaan adveksi-difusi 2-dimensi

```
% Persamaan Adveksi Difusi 2 Dimensi dengan Metode Beda Hingga
%                   Irreguler Boundary Problem
%  Semua Irregular Boundary harus diletakan pada Reguler Grid
dengan kode seperti berikut :
%
%   0-Di Luar                  5-Di Kiri     FSx CSy
%   1-Di Dalam CSx CSy         6-Kiri Bawah  FSx FSy   |_
%   2-Di Bawah CSx CFy         7-Kanan Bawah BSx FSy    _     _|
%   3-Di Kanan BSx CSy         8-Kiri Atas   FSx BSy   |      _
%   4-Di Atas  CSx BSy         9-Kanan Atas  BSx BSy           |

clear all; clc;
% Matriks Danau Unhas
CD = xlsread('D:\ISEKAI\Tugas Akhir\OTW
HASIL\KodeGridDanauUnhas.xlsx','Model 1','B2:BM81');
Nx = 80;               % Jumlah Grid Sumbu x
Ny = 64;               % Jumlah Grid Sumbu y
Delx = 0.5;            % Lebar Grid Sumbu x
Dely = 0.5;            % Lebar Grid Sumbu y
% Difusi Logam Arsen
Dx = 11.11*10^(-8);    % Koefisien Difusi arah x
Dy = 11.11*10^(-8);    % Koefisien Difusi arah y
% Difusi Logam Krom
Dx = 7.3*10^(-8);
Dy = 7.3*10^(-8);
% Difusi Logam Timbal
Dx = 11.6*10^(-8);
Dy = 11.6*10^(-8);
% Kasus 1
Vx = 0;                % Koefisien Adveksi arah x
Vy = 0;                % Koefisien Adveksi arah y
% Kasus 2
Vx = -0.159*10^(-8);
Vy = -0.159*10^(-8);
% Nilai Konstanta Skema Beda Hingga
Ax = Vx/Delx;
Ay = Vy/Dely;
Bx = Dx/(Delx^2);
By = Dy/(Dely^2);
Ax2 = Ax/2.;
Ay2 = Ay/2.;
% Pembagi Skema Beda Hingga
Denom1=2.*Bx+2*By;      % Untuk Kode 1
Denom2=-Ay+2.*Bx-By;    % Untuk Kode 2
Denom3=Ax-Bx+2.*By;     % Untuk Kode 3
Denom4=Ay+2.*Bx-By;     % Untuk Kode 4
Denom5=-Ax-Bx+2.*By;    % Untuk Kode 5
Denom6=-Ax-Ay-Bx-By;    % Untuk Kode 6
Denom7=Ax-Ay-Bx-By;     % Untuk Kode 7
Denom8=-Ax+Ay-Bx-By;    % Untuk Kode 8
Denom9=Ax+Ay-Bx-By;     % Untuk Kode 9

% Pemisalan Nilai Awal Grid
for i = 1:Nx
 for j = 1:Ny
```

```
   if CD(i,j) ~= 0
    CA(i,j) = 1;
   end
  end
end

% Pembagian antara Daratan dan Perairan
for i = 1:Nx
 Rub(i) = 1;
 for j = 2:Ny
  if CA(i,j)-CA(i,j-1) ~= 0
   Rub(i) = Rub(i)+1;
  end
 end
end

% Perhitungan Jumlah Grid yang dibagi antara Daratan dan Perairan
for i = 1:Nx
 r = 1;
 Nilai = 1;
 for j = 2:Ny
  if CA(i,j)-CA(i,j-1) == 0
   Nilai = Nilai+1;
   CDR(i,r) = Nilai;
  else
   r = r+1;
   Nilai = 1;
  end
 end
end

% Modifikasi Jumlah Grid di Daratan dan Perairan
for i=22:30
 CDR(i,3)=1;
end
for i=24:30
 CDR(i,5)=0;
end
CDR(20,5)=0;
CDR(21,5)=0;
CDR(23,7)=0;
CDR(24,7)=0;
CDR(31,3)=0;
CDR(32,3)=0;

TTP = 0;
for i = 1:Nx
 has = 0;
 for j = 1:Ny
  if CD(i,j) ~= 0
   has = has+1;
  end
 end
 CDP(i) = has;     % Total Jumlah Grid di Perairan
 if i == 1
  CDK(i) = CDP(i);
 else
  CDK(i) = CDK(i-1)+CDP(i);
```

```
 end
 TTP = TTP+has;     % Jumlah Solusi di Dalam Matriks
end

% Nilai Awal Konsentrasi pada Masing Masing Statiun
Robin(1:TTP)=0.;

% Logam Arsen
Robin(1039)=0.011;      % St1
Robin(1092)=0.011;
Robin(1145)=0.011;
Robin(1197)=0.011;
Robin(1249)=0.011;
Robin(2653)=0.385;      % St2
Robin(2654)=0.385;
Robin(2655)=0.385;
Robin(2656)=0.385;
Robin(2657)=0.385;
Robin(2658)=0.385;
Robin(933)=0.454;       % St3
Robin(986)=0.454;
Robin(1040)=0.454;
Robin(1093)=0.454;
Robin(1)=0.463;         % St4
Robin(2)=0.463;
Robin(3)=0.463;
Robin(689)=0.48;        % St5

% Logam Krom
Robin(1039)=2.02;       % St1
Robin(1092)=2.02;
Robin(1145)=2.02;
Robin(1197)=2.02;
Robin(1249)=2.02;
Robin(2653)=10^(-8);    % St2
Robin(2654)=10^(-8);
Robin(2655)=10^(-8);
Robin(2656)=10^(-8);
Robin(2657)=10^(-8);
Robin(2658)=10^(-8);
Robin(933)=10^(-8);     % St3
Robin(986)=10^(-8);
Robin(1040)=10^(-8);
Robin(1093)=10^(-8);
Robin(1)=10^(-8);       % St4
Robin(2)=10^(-8);
Robin(3)=10^(-8);
Robin(689)=10^(-8);     % St5

% Logam Timbal
Robin(1039)=6.1;        % St1
Robin(1092)=6.1;
Robin(1145)=6.1;
Robin(1197)=6.1;
Robin(1249)=6.1;
Robin(2653)=5.53;      % St2
Robin(2654)=5.53;
Robin(2655)=5.53;
```

```
Robin(2656)=5.53;
Robin(2657)=5.53;
Robin(2658)=5.53;
Robin(933)=0.24;      % St3
Robin(986)=0.24;
Robin(1040)=0.24;
Robin(1093)=0.24;
Robin(1)=0.46;        % St4
Robin(2)=0.46;
Robin(3)=0.46;
Robin(689)=7.2;       % St5

TBC=0;
for i=1:TTP
    if Robin(i)~=0
        TBC=TBC+1;    % Banyaknya Nilai Awal
    end
end

% Menyusun Matriks AX=B dimana A = MA(TTP,TTP), X = MX(TTP), dan B
= MB(TTP)
% Dengan Nilai awal Matriksnya 0
MA(1:TTP,1:TTP)=0.0;
MBB(1:TTP-TBC)=0;
MAA(1:TTP-TBC,1:TTP-TBC)=0;

% Menghitung Nilai pada Matriks A
% Skema Beda Hingga Adveksi Difusi 2-Dimensi
k=0;
for i = 1:Nx
 for j = 1:Ny
  if CD(i,j)~=0
  k=k+1;
  pilih=CD(i,j);
  switch pilih
   case 1  %Kode Grid 1 FIX
    MA(k,k)=1.;
    MA(k,k+1)=-(By-Ay2)/Denom1;
    MA(k,k-1)=-(By+Ay2)/Denom1;
    if Rub(i-1)<=3
     MA(k,k+CDR(i,1)-CDR(i-1,1)-CDP(i-1))=-(Bx+Ax2)/Denom1;
     MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i))=-(Bx-Ax2)/Denom1;
    else
     if j<=30
      MA(k,k+CDR(i,1)-CDR(i-1,1)-CDP(i-1))=-(Bx+Ax2)/Denom1;
      MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i))=-(Bx-Ax2)/Denom1;
     else
      if Rub(i)<=3
       MA(k,k+CDR(i,1)-CDR(i-1,1)-CDP(i-1)-CDR(i-1,3))=-
(Bx+Ax2)/Denom1;
       MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i))=-(Bx-Ax2)/Denom1;
      elseif Rub(i+1)<=3
       MA(k,k+CDR(i,1)-CDR(i-1,1)-CDP(i-1))=-(Bx+Ax2)/Denom1;
       MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i)+CDR(i,3))=-(Bx-
Ax2)/Denom1;
      else
       if j<=42
        MA(k,k+CDR(i,1)-CDR(i-1,1)-CDP(i-1))=-(Bx+Ax2)/Denom1;
```

```
        MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i))=-(Bx-Ax2)/Denom1;
       elseif i>=25
        MA(k,k+CDR(i,1)-CDR(i-1,1)-CDP(i-1))=-(Bx+Ax2)/Denom1;
        MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i))=-(Bx-Ax2)/Denom1;
       else
        MA(k,k+CDR(i,1)-CDR(i-1,1)-CDR(i-1,3)-CDR(i-1,5)-CDP(i-
1)+CDR(i,3)+CDR(i,5))=-(Bx+Ax2)/Denom1;
        MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i)+CDR(i,3)+CDR(i,5)-
CDR(i+1,3)-CDR(i+1,5))=-(Bx-Ax2)/Denom1;
       end
      end
     end
    end
   case 2  %Kode Grid 2 FIX
    MA(k,k)=1.;
    MA(k,k+1)=-(-Ay-2.*By)/Denom2;
    MA(k,k+2)=-By/Denom2;
    if Rub(i)<=3
     MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i))=-(-Ax2+Bx)/Denom2;
     MA(k,k+CDR(i,1)-CDR(i-1,1)-CDP(i-1))=-(Ax2+Bx)/Denom2;
    else
     if Rub(i+1)<=3
      MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i)+CDR(i,3))=-(-
Ax2+Bx)/Denom2;
     else
      MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i))=-(-Ax2+Bx)/Denom2;
     end
      MA(k,k+CDR(i,1)-CDR(i-1,1)-CDP(i-1))=-(Ax2+Bx)/Denom2;
    end
   case 3  %Kode Grid 3 FIX
    MA(k,k)=1.;
    MA(k,k+1)=-(-Ay2+By)/Denom3;
    MA(k,k-1)=-(Ay2+By)/Denom3;
    MA(k,k+CDR(i,1)-CDR(i-1,1)-CDP(i-1))=-(Ax-2.*Bx)/Denom3;
    MA(k,k+CDR(i,1)-CDP(i-1)-CDR(i-2,1)-CDP(i-2))=-Bx/Denom3;
   case 4  %Kode Grid 4 FIX
    MA(k,k)=1.;
    MA(k,k-1)=-(Ay-2.*By)/Denom4;
    MA(k,k-2)=-By/Denom4;
    if Rub(i)<=3
     if Rub(i-1)<=3
      MA(k,k+CDR(i,1)-CDR(i-1,1)-CDP(i-1))=-(Ax2+Bx)/Denom4;
     else
      MA(k,k+CDR(i,1)-CDR(i-1,1)-CDP(i-1)-CDR(i-1,3))=-
(Ax2+Bx)/Denom4;
     end
    else
     MA(k,k+CDR(i,1)-CDR(i-1,1)-CDP(i-1))=-(Ax2+Bx)/Denom4;
    end
    MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i))=-(-Ax2+Bx)/Denom4;
   case 5  %Kode Grid 5 FIX
    MA(k,k)=1.;
    MA(k,k+1)=-(-Ay2+By)/Denom5;
    MA(k,k-1)=-(Ay2+By)/Denom5;
    if Rub(i)<=3
     MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i))=-(-Ax-2.*Bx)/Denom5;
     MA(k,k+CDR(i,1)-CDR(i+2,1)+CDP(i)+CDP(i+1))=-Bx/Denom5;
    else
     if j<=42
```

```
      MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i))=-(-Ax-2.*Bx)/Denom5;
      MA(k,k+CDR(i,1)-CDR(i+2,1)+CDP(i)+CDP(i+1))=-Bx/Denom5;
     else
      if Rub(i)<6
       if Rub(i+1)<6
        MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i)+CDR(i,3)-CDR(i+1,3))=-(-
Ax-2.*Bx)/Denom5;
       else
        MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i)+CDR(i,3)-CDR(i+1,3)-
CDR(i+1,5))=-(-Ax-2.*Bx)/Denom5;
       end
        MA(k,k+CDR(i,1)-CDR(i+2,1)+CDP(i)+CDP(i+1)+CDR(i,3)-
CDR(i+2,3)-CDR(i+2,5))=-Bx/Denom5;
      else
       MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i)+CDR(i,3)+CDR(i,5)-
CDR(i+1,3)-CDR(i+1,5))=-(-Ax-2.*Bx)/Denom5;
       MA(k,k+CDR(i,1)-
CDR(i+2,1)+CDP(i)+CDP(i+1)+CDR(i,3)+CDR(i,5)-CDR(i+2,3)-
CDR(i+2,5))=-Bx/Denom5;
      end
     end
    end
   case 6  %Kode Grid 6 FIX
    MA(k,k)=1.;
    MA(k,k+1)=-(-Ay-2.*By)/Denom6;
    MA(k,k+2)=-By/Denom6;
    if Rub(i)<=3
     MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i))=-(-Ax-2.*Bx)/Denom6;
     MA(k,k+CDR(i,1)-CDR(i+2,1)+CDP(i)+CDP(i+1))=-Bx/Denom6;
    else
     if j<=35
      MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i))=-(-Ax-2.*Bx)/Denom6;
      MA(k,k+CDR(i,1)-CDR(i+2,1)+CDP(i)+CDP(i+1))=-Bx/Denom6;
     else
      if Rub(i)<6
       if Rub(i+1)<6
        MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i)+CDR(i,3)-CDR(i+1,3))=-(-
Ax-2.*Bx)/Denom6;
       else
        MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i)+CDR(i,3)-CDR(i+1,3)-
CDR(i+1,5))=-(-Ax-2.*Bx)/Denom6;
       end
       MA(k,k+CDR(i,1)-CDR(i+2,1)+CDP(i)+CDP(i+1)+CDR(i,3)-
CDR(i+2,3)-CDR(i+2,5))=-Bx/Denom6;
      else
       MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i)+CDR(i,3)+CDR(i,5)-
CDR(i+1,3)-CDR(i+1,5))=-(-Ax-2.*Bx)/Denom6;
       MA(k,k+CDR(i,1)-CDR(i+2,1)+CDP(i)+CDP(i+1)+CDR(i,3)+
CDR(i,5)-CDR(i+2,3)-CDR(i+2,5))=-Bx/Denom6;
      end
     end
    end
   case 7  %Kode Grid 7 FIX
    MA(k,k)=1.;
    MA(k,k+1)=-(-Ay-2.*By)/Denom7;
    MA(k,k+2)=-By/Denom7;
    MA(k,k+CDR(i,1)-CDR(i-1,1)-CDP(i-1))=-(Ax-2.*Bx)/Denom7;
    MA(k,k+CDR(i,1)-CDP(i-1)-CDR(i-2,1)-CDP(i-2))=-Bx/Denom7;
   case 8  %Kode Grid 8 FIX
```

```
    MA(k,k)=1.;
    MA(k,k-1)=-(Ay-2.*By)/Denom8;
    MA(k,k-2)=-By/Denom8;
    if Rub(i)<=3
     MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i))=-(-Ax-2.*Bx)/Denom8;
     MA(k,k+CDR(i,1)-CDR(i+2,1)+CDP(i)+CDP(i+1))=-Bx/Denom8;
    else
     if j<=35
      MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i))=-(-Ax-2.*Bx)/Denom8;
      MA(k,k+CDR(i,1)-CDR(i+2,1)+CDP(i)+CDP(i+1))=-Bx/Denom8;
     else
      if Rub(i)<6
       if Rub(i+1)>5
        MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i)+CDR(i,3)-CDR(i+1,3)-
CDR(i+1,5))=-(-Ax-2.*Bx)/Denom8;
        MA(k,k+CDR(i,1)-CDR(i+2,1)+CDP(i)+CDP(i+1)+CDR(i,3)-
CDR(i+2,3)-CDR(i+2,5))=-Bx/Denom8;
       elseif Rub(i+1)<=3
        MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i)+CDR(i,3))=-(-Ax-
2.*Bx)/Denom8;
        MA(k,k+CDR(i,1)-CDR(i+2,1)+CDP(i)+CDP(i+1)+CDR(i,3))=-
Bx/Denom8;
       else
        MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i)+CDR(i,3)-CDR(i+1,3))=-(-
Ax-2.*Bx)/Denom8;
        MA(k,k+CDR(i,1)-CDR(i+2,1)+CDP(i)+CDP(i+1)+CDR(i,3)-
CDR(i+2,3)-CDR(i+2,5))=-Bx/Denom8;
       end
      else
       MA(k,k+CDR(i,1)-CDR(i+1,1)+CDP(i)+CDR(i,3)+CDR(i,5)-
CDR(i+1,3)-CDR(i+1,5))=-(-Ax-2.*Bx)/Denom8;
       MA(k,k+CDR(i,1)-CDR(i+2,1)+CDP(i)+CDP(i+1)+CDR(i,3)+
CDR(i,5)-CDR(i+2,3)-CDR(i+2,5))=-Bx/Denom8;
      end
     end
    end
   case 9  %Kode Grid 9 FIX
    MA(k,k)=1.;
    MA(k,k-1)=-(Ay-2.*By)/Denom9;
    MA(k,k-2)=-By/Denom9;
    MA(k,k+CDR(i,1)-CDR(i-1,1)-CDP(i-1))=-(Ax-2.*Bx)/Denom9;
    MA(k,k+CDR(i,1)-CDR(i-2,1)-CDP(i-1)-CDP(i-2))=-Bx/Denom9;
   end
  end
 end
end

k=0;
for i=1:TTP
 if Robin(i)~=0
  k=k+1;
 else
  l=0;
  for j=1:TTP
   if Robin(j)~=0
    l=l+1;
    MBB(i-k)=MBB(i-k)-MA(i,j)*Robin(j); % Matriks MB tanpa Nilai
Konsentrasi Awal
   else
```

```
    MAA(i-k,j-l)=MA(i,j);                % Matriks MA tanpa Nilai
Konsentrasi Awal
   end
  end
 end
end

% Menghitung Nilai Matriks MX
MXX=pinv(MAA)*MBB';                                   % Matriks MX
tanpa Nilai Konsentrasi Awal
j=0;
for i=1:TTP
 if Robin(i)~=0
  j=j+1;
  MX(i)=Robin(i);
 else
  MX(i)=MXX(i-j);
 end
end

% Mengembalikan Nilai ke dalam Matriks Danau Unhas
k=0;
for i = 1:Nx
 for j = 1:Ny
  if CD(i,j)~= 0
   k=k+1;
   C(i,j)=MX(k);
  else
   C(i,j)=NaN;
  end
 end
end

% Membuat Plot
[X,Y] = meshgrid(1:Nx,1:Ny);
Z(:,:)=C(1:Nx,1:Ny);

% Plot dalam bentu 2-Dimensi
figure(1),
contourf(X,Y,Z',300,'linestyle','none');
set(gcf,'color','w');

% Lokasi teks stasiun
annotation('textbox',[0.400066408701109 0.849231950844855
0.0695764484417484 0.054761904761905],'String',{'St
1'},'FitBoxToText','off','BackgroundColor',[1 1 1]);
annotation('textbox',[0.696345952729553 0.375268817204301
0.0695764484417484 0.054761904761905],'String',{'St
2'},'FitBoxToText','off','BackgroundColor',[1 1 1]);
annotation('textbox',[0.358838109182179 0.252073732718895
0.0695764484417484 0.054761904761905],'String',{'St
3'},'FitBoxToText','off','BackgroundColor',[1 1 1]);
annotation('textbox',[0.152691382555951 0.228801843317972
0.0695764484417484 0.054761904761905],'String',{'St
4'},'FitBoxToText','off','BackgroundColor',[1 1 1]);
annotation('textbox',[0.383715226939971 0.455760368663595
0.0695764484417484 0.054761904761905],'String',{'St
5'},'FitBoxToText','off','BackgroundColor',[1 1 1]);
```

```
hold on
xlabel('\it\bf Lebar Danau','fontsize',12,'Color','b');
ylabel('\it\bf Panjang Danau','fontsize',12,'Color','b');
caxis([0 max(MX)]);
hold off

% Plot dalam bentuk 3-Dimensi
figure(2),
ZZ(:,:)=C(1:Nx,1:Ny);
surf(X,Y,ZZ','linestyle','none');
axis ([1 Nx 1 Ny 0 100]);
set(gcf,'color','w');
caxis([0 max(MX)]);

% Mengatur label axis 3-Dimensi
xlabel('\it\bf Lebar Danau','fontsize',12,'Color','b');
xh = get(gca,'XLabel');
set(xh, 'Units', 'Normalized');
pos = get(xh, 'Position');
set(xh, 'Position', pos.*[1,0,1],'Rotation',20);
ylabel('\it\bf Panjang Danau','fontsize',12,'Color','b');
yh = get(gca,'YLabel');
set(yh, 'Units', 'Normalized');
pos = get(yh, 'Position');
set(yh, 'Position', pos.*[1,-1,1],'Rotation',-30);
zlabel('\it\bf Konsentrasi Polutan','fontsize',12,'Color','b');
zh = get(gca,'ZLabel');
set(zh, 'Units', 'Normalized');
pos = get(zh, 'Position');
set(zh, 'Position', pos.*[1,1,1],'Rotation',90);
```