

### Daftar Pustaka

- Aryati, L., 2011, *Pengantar Persamaan Diferensial Parsial*, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada.
- Ayres, F., Ault, J. C, Ratna L., 1999, *Persamaan Diferensial*, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Airlangga.
- Budhi, W.B., 2001, *Kalkulus Peubah Banyak dan Penggunaannya*, Penerbit ITB, Bandung.
- Causon, M., D., Mingham, G.,C., 2010, *Introductory Finite Difference Methods For PDEs*, Department of Computing and Mathematics, Manchester Metropolitan University, UK
- DuChateau, P., and Zachmann, D.W., 1986, *Partial Differential Equations*, McGraw-Hill, New York
- Dutykh, Denys, 2010, *Mathematical modeling in the environment: from tsunamis to powder-snow avalanches*, Mathematics, Universite de Savoie.
- Evangelia, T. Flouri, dkk., 2013, *Application of a finite difference computational model to simulation of earthquake generated tsunamis*. Journal of Applied Numerical Mathematics 67 (2013) 111-125
- Flyer, N., Lehto, E., Blaise, S., Wright, B., G., St-Cyr, A., 2012, *A Guide to RBF-Generated Finite Differences for Nonlinear Transport: Shallow Water Simulations on a Sphere*. Journal of Department of Mathematics.
- Hafeez, H., 2008, *Inundation of tsunami Waves and It's Relation To The Tsunami Run-Up*, Pakistan Journal of Meteorology.
- Humi, M., and Miller, W.B., 1992, *Boundary Value Problems and Partial Differential Equations*. United States of America: PWS KENT Publishing Company, Boston.
- Imamura, F., 1997, *IUGG/IOC TIME PROJECT Numerical Method of Tsunami Simulation with the Leap-Frog Scheme*, Intergovernmental Oceanographic Commission, Tohoku University, Japan



- Kampf Jochen, 2009, *Ocean Modelling for Beginners, Using Open-Source Software*, Adelaide, Australia.
- Leveque, R.J., 1999, *Numerical Methods for Conservation Laws*, Second Edition, Basel, Boston, Berlin.
- Luknanto Djoko, 2003, *Model Matematika*, Fakultas Teknik, Universitas Gadjah Mada
- Munson, R.B., Young F. D., Okiishi H. T, 2004, *Mekanika Fluida*, Edisi Keempat, Jilid I, Penerit Erlangga, Jakarta
- Pelinovsky, E., N., Mazova, R., KH., 1992., *Exact Analytical Solutions of Nonlinear Problems of Tsunami Wave Run-up on Slopes with Different Profiles*, Institute of Applied Physics, Academy of Sciences, Rusia.
- Pinsky, M. A., 1998, *Partial Differential Equations and Boundary Value Problems with Applications*, McGraw-Hill, Boston
- Qiu Jianxian, and SHU Chi-Wang, 2003, *Finite Difference WENO Schemes With LAX-WENDROFF-TYPE Time Discretizations*, journal of Society for Industrial and Applied Mathematics.
- Robinson, R. Colin, 2011, *Shallow Water Equations*, Syracuse University.
- Scott, L., R.,2013, *Tsunami Simulation*, The University of Chicago, Chicago.
- Suprawoto, 2008, *Memahami Bencana*, Jakarta.
- Triatmadja, R., 2009, *Model Matematik Teknik Pantai Menggunakan Diferensi Hingga dan Metode Karakteristik*, Yogyakarta.
- Triatmadja, R., 2010, *Tsunami*, Universitas Gadjah Mada, Yogyakarta.
- Wardani, E., Vinalia, 2014, *Solusi Numerik Untuk Persamaan Navier-Stokes Dimensi Dua Dengan Metode Beda Hingga*, Skripsi, (Tidak Dipublikasikan) Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada, Yogyakarta.
- Weiyang, Tan, 1992, *Shallow Water Hydrodynamics, Mathematical Theory and Numerical Solution for a Two - dimensional Of Shallow water Equations*, Nanjing Research Institute of Hydrology and Water Resources, Penerbit: Water & Power Press, Beijing, China.



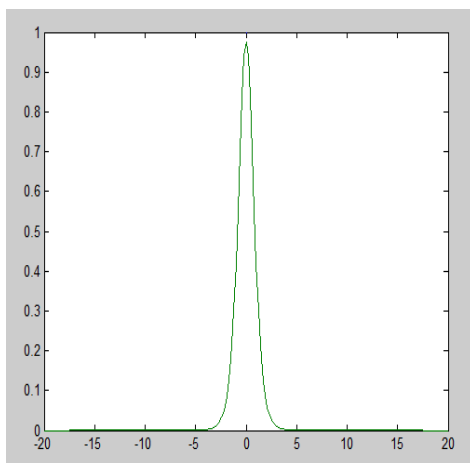
Wu, Y., H., Tian, J., 2000, *Mathematical analysis of long-wave breaking on open channels with bottom friction*, School of Mathematics and Statistics, Curtin University of Technology, Perth, Australia.

Zahibo, N., Pelinovsky, E., Talipova, T., Kozelkov, A., Kurkin, A., 2006, *Analytical and Numerical Study of Nonlinear Effects at Tsunami Modeling*. Journal of Applied Mathematics and Computation 174 (2006) 795–809

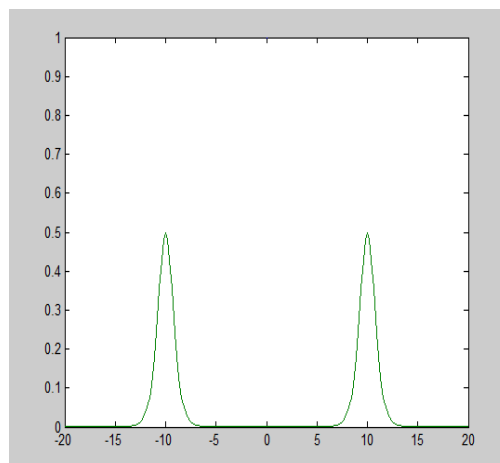
## LAMPIRAN

### Lampiran 1. Listing program linear pada model tsunami dimensi satu

```
clear
g=10;
t=0;
h=10;
dt=0.005;
co=10;
psi=10;
x=-20:0.2:20;
y=sech(x).^2;
plot(x,y);
xlabel('Amplitudo awal/Kedalaman'), ylabel('gelombang
amplitudo/Kedalaman');
for i=1:200;
    t=i*dt;
    y=(sech(x-co*t).^2+sech(x+co*t).^2)/2;
    plot(0,1,x,y)
    pause(0.2)
end
```



**Gambar 4.2.** Sebelum gelombang terpecah



**Gambar 4.3.** Setelah gelombang terpecah



## Lampiran 2. Listing program amplitudo linear dimensi satu

```
clear
n=50;
g=10;
t=0;
h=10;
dt=0.005;
co=10;
psi=10;
x=-20:0.2:20;
y=sech(x).^2;
plot(x,y);
A= max (y);
tt=0;
for i=1:200;
    t=i*dt;
    y=(sech(x-co*t).^2+sech(x+co*t).^2)/2;
    A=[A max(y)]
    tt=[tt t];
    plot (0,1,x,y)
    pause (0.2)
end
plot (tt,A,0,0);
xlabel('jarak'), ylabel('amplitudo gelombang');
```



### Lampiran 3. Listing program efek nonlinier dimensi satu

Menurut (Robinson, 2011) listing program efek nonlinear dimensi satu adalah:

```
clf;
clear all;
% define the grid size
n = 50;
dt = 0.01;
dx = 1;
g = 9.8;
H = ones(1,n+2); % displacement vector (this is what gets drawn)
M = zeros(1,n+2); % x velocity
% draw the mesh
hold all;
% create initial displacement
xx=1:52;
H=sech((xx-25)*0.6).^2;
% empty matrix for half-step calculations
Hx = zeros(n+1);
Mx = zeros(n+1);
for k=1:300
% redraw the mesh
hold off;
plot (H);
ylim([0 2]);
pause (0.2);
% blending the edges keeps the function stable
H(1) = H(2);
H(n+2) = H(n+1);
% reverse direction at the x edges
M(1) = -M(2);
M(n+2) = -M(n+1);
% First half step
i = 1:n;
% height
```



```
Hx(i) = (H(i+1)+H(i))/2 - dt/(2*dx)*(M(i+1)-M(i));  
% x momentum  
Mx(i) = (M(i+1)+M(i))/2 - ...  
dt/(2*dx)*( M(i+1).^2./H(i+1) - M(i).^2./H(i) + ...  
g/2*H(i+1).^2 - g/2*H(i).^2 ...  
);  
% Second half step  
i = 2:n;  
% height  
H(i) = H(i) - (dt/dx)*(Mx(i)-Mx(i-1));  
% x momentum  
M(i) = M(i) - (dt/dx)*((Mx(i).^2./Hx(i) + g/2*Hx(i).^2) - ...  
(Mx(i-1).^2./Hx(i-1) + g/2*Hx(i-1).^2));  
end
```



#### Lampiran 4. Listing program amplitudo efek nonlinier dimensi satu

Menurut (Robinson, 2011) listing program amplitudo efek nonlinier dimensi satu adalah:

```
clf;
clear all;
% define the grid size
n = 50;
dt = 0.01;
dx = 1;
g = 9.8;
H = ones(1,n+2); % displacement vector (this is what gets drawn)
M = zeros(1,n+2); % x velocity
% draw the mesh
hold all;
% create initial displacement
xx=1:52;
H=sech((xx-25)*0.6).^2;
% empty matrix for half-step calculations
Hx = zeros(n+1);
Mx = zeros(n+1);
tt=0;
A = max (H);
for k=1:300
% redraw the mesh
hold off;
plot (H);
ylim([0 2]);
pause (0.2);
% blending the edges keeps the function stable
H(1) = H(2);
H(n+2) = H(n+1);
% reverse direction at the x edges
M(1) = -M(2);
M(n+2) = -M(n+1);
% First half step
i = 1:n;
% height
Hx(i) = (H(i+1)+H(i))/2 - dt/(2*dx)*(M(i+1)-M(i));
% x momentum
Mx(i) = (M(i+1)+M(i))/2 - ...
dt/(2*dx)*( M(i+1).^2./H(i+1) - M(i).^2./H(i) + ...
g/2*H(i+1).^2 - g/2*H(i).^2 ...
);
% Second half step
i = 2:n;
% height
H(i) = H(i) - (dt/dx)*(Mx(i)-Mx(i-1));
A=[A max(H)]
tt=[tt dt*k];
% x momentum
```



```
M(i) = M(i) - (dt/dx)*((Mx(i).^2./Hx(i) + g/2*Hx(i).^2) - ...  
(Mx(i-1).^2./Hx(i-1) + g/2*Hx(i-1).^2));  
end  
plot (tt, A);  
xlabel('jarak'), ylabel('amplitudo gelombang');
```

### Lampiran 5. Listing program perbedaan amplitudo linear dengan nonlinear pada dimensi satu

```
clear all;  
clc;  
h=1;  
Ain=0:0.01:1;  
A=(Ain/4)+(1/2)*(sqrt(1+Ain/h)-1);  
Alin=(1/2)*Ain;  
plot (Ain,A,'r',Ain,Alin,':b');  
legend('Nonlinear','Linear','Location','NorthWest');  
xlabel('Amplitudo awal (A)'), ylabel('Amplitudo akhir (Ain)');
```



## Lampiran 6. Listing program efek nonlinier dimensi dua

Menurut (Robinson, 2011) listing program amplitudo efek nonlinier dimensi dua adalah:

```
clf;
clear all;
% define the grid size
n = 50;
dt = 0.01;
dx = 1;
dy = 1;
g = 9.8;
H = ones(n+2,n+2); % displacement matrix (this is what gets drawn)
M = zeros(n+2,n+2); % x velocity
N = zeros(n+2,n+2); % y velocity
% draw the mesh
grid = surf(H);
axis([1 n 1 n 1 3]);
hold all;
% create initial displacement
[x,y] = meshgrid( linspace(-3,3,10) );
R = sqrt(x.^2 + y.^2) + eps;
Z = (sin(R)./R);
Z = max(Z,0);
% add displacement to the height matrix
w = size(Z,1);
i = 10:w+9;
j = 20:w+19;
H(i,j) = H(i,j) + Z;
% empty matrix for half-step calculations
Hx = zeros(n+1,n+1);
Hy = zeros(n+1,n+1);
Mx = zeros(n+1,n+1);
My = zeros(n+1,n+1);
Nx = zeros(n+1,n+1);
Ny = zeros(n+1,n+1);
while 1==1
% redraw the mesh
set(grid, 'zdata', H);
drawnow
% blending the edges keeps the function stable
H(:,1) = H(:,2);
H(:,n+2) = H(:,n+1);
H(1,:) = H(2,:);
H(n+2,:) = H(n+1,:);
% reverse direction at the x edges
M(1,:) = -M(2,:);
M(n+2,:) = -M(n+1,:);
% reverse direction at the y edges
N(:,1) = -N(:,2);
N(:,n+2) = -N(:,n+1);
% First half step
```



```
i = 1:n+1;
j = 1:n+1;
% height
Hx(i,j) = (H(i+1,j+1)+H(i,j+1))/2 - dt/(2*dx)*(M(i+1,j+1)-
M(i,j+1));
Hy(i,j) = (H(i+1,j+1)+H(i+1,j))/2 - dt/(2*dy)*(N(i+1,j+1)-
N(i+1,j));
% x momentum
Mx(i,j) = (M(i+1,j+1)+M(i,j+1))/2 - ...
dt/(2*dx)*( M(i+1,j+1).^2./H(i+1,j+1) - M(i,j+1).^2./H(i,j+1) +
...
g/2*H(i+1,j+1).^2 - g/2*H(i,j+1).^2 ...
);
My(i,j) = (M(i+1,j+1)+M(i+1,j))/2 - ...
dt/(2*dy)*( (N(i+1,j+1).*M(i+1,j+1)./H(i+1,j+1)) -
(N(i+1,j).*M(i+1,j)./H(i+1,j)) );
% y momentum
Nx(i,j) = (N(i+1,j+1)+N(i,j+1))/2 - ...
dt/(2*dx)*( (M(i+1,j+1).*N(i+1,j+1)./H(i+1,j+1)) - ...
(M(i,j+1).*N(i,j+1)./H(i,j+1)));
Ny(i,j) = (N(i+1,j+1)+N(i+1,j))/2 - ...
dt/(2*dy)*( (N(i+1,j+1).^2./H(i+1,j+1) + g/2*H(i+1,j+1).^2) - ...
(N(i+1,j).^2./H(i+1,j) + g/2*H(i+1,j).^2));
% Second half step
i = 2:n+1;
j = 2:n+1;
% height
H(i,j) = H(i,j) - (dt/dx)*(Mx(i,j-1)-Mx(i-1,j-1)) - ...
(dt/dy)*(Ny(i-1,j)-Ny(i-1,j-1));
% x momentum
M(i,j) = M(i,j) - (dt/dx)*( (Mx(i,j-1).^2./Hx(i,j-1) + g/2*Hx(i,j-
1).^2) - ...
(Mx(i-1,j-1).^2./Hx(i-1,j-1) + g/2*Hx(i-1,j-1).^2)) ...
- (dt/dy)*( (Ny(i-1,j).*My(i-1,j)./Hy(i-1,j)) - ...
(Ny(i-1,j-1).*My(i-1,j-1)./Hy(i-1,j-1)));
% y momentum
N(i,j) = N(i,j) - (dt/dx)*( (Mx(i,j-1).*Nx(i,j-1)./Hx(i,j-1)) - ...
(Mx(i-1,j-1).*Nx(i-1,j-1)./Hx(i-1,j-1))) ...
- (dt/dy)*( (Ny(i-1,j).^2./Hy(i-1,j) + g/2*Hy(i-1,j).^2) - ...
(Ny(i-1,j-1).^2./Hy(i-1,j-1) + g/2*Hy(i-1,j-1).^2));
xlabel('merambat ke arah x (KM)'), ylabel('merambat ke arah y
(KM)'), zlabel('amplitudo');
end
```