

LAMPIRAN A

A.1 Program Octave: Bendungan Bobol di Atas Topografi Datar

```
% Inisialisasi Awal
clear all;
close all;
clc;

% Definisi Parameter
L=200;          % Panjang domain
T=20;          % Waktu simulasi
dx=1;          % Ukuran grid spasial
g=9.81;        % Gravitasi (m/s2)
dt=0.1;        % Waktu langkah (time step)
Nx=floor(L/dx); % Jumlah titik grid dalam ruang
Nt=floor(T/dt); % Jumlah iterasi waktu
x=0:dx:L-2*dx; % Array posisi dalam domain

% Inisialisasi Variabel
h=zeros(Nx,Nt); % Ketebalan air
u=zeros(Nx+1,Nt); % Kecepatan fluida
q=zeros(Nx+1,Nt); % Debit aliran (q = h*u)
thres=0.01;     % Ambang batas tinggi air

% Kondisi Awal
for i=1:Nx
    if i<floor(Nx/2)
        h(i,1)=1;
        w(i,1)=1;
    else
```

```

h(i,1)=0.25;
w(i,1)=0.25;
end
b(i)=0; % topografi dasar dianggap datar.
end

u(1:Nx+1,1)=0; % Kecepatan awal di semua titik adalah 0
b(Nx+1)=0; % kondisi batas topografi dasar di batas kanan

% Kondisi Batas untuk Kecepatan aliran(u)
u(1,1:Nt)=0; % batas kiri
u(Nx+1,1:Nt)=0; % batas kanan

% Kondisi Batas untuk Debit Aliran (q)
q(1,1:Nt)=0; % batas kiri
q(Nx+1,1:Nt)=0; % batas kanan

% Kondisi Batas untuk Ketebalan Air (h)
h(1,1:Nt)=1; % ketinggian tetap 1 di sisi kiri
h(Nx,1:Nt)=0.25; % ketinggian tetap 0.25 di sisi kanan

for n=1:Nt % Perhitungan dalam Loop Waktu
    % Menentukan Nilai Ketinggian untuk Debit Aliran
    for i=1:Nx-1
        if u(i+1,n)>0
            hb(i+1,n)=h(i,n); % Jika kecepatan positif, gunakan tinggi di titik
            sebelumnya
        else
            hb(i+1,n)=h(i+1,n); % Jika kecepatan negatif, gunakan tinggi di titik
            selanjutnya
        end
        hbar(i+1,n)=0.5*(h(i,n)+h(i+1,n)); % Rata-rata ketebalan air
    end
end

```

```

end

% Menghitung Debit Aliran
for i=2:Nx
    q(i,n)=hb(i,n)*u(i,n);
end

% Menghitung Rata-rata Debit dan Kecepatan
for i=1:Nx
    qbar(i,n)=0.5*(q(i+1,n)+q(i,n)); % Rata-rata debit Aliran
    if qbar(i,n)<0
        ub(i,n)=u(i+1,n);
    else
        ub(i,n)=u(i,n);
    end
    w(i,n+1)=w(i,n)-dt*(q(i+1,n)-q(i,n))/dx;
    h(i,n+1)=w(i,n+1)-b(i);
end

% Menghitung Perubahan Kecepatan
for i=1:Nx-1
    uux(i+1,n)=qbar(i,n)*(ub(i+1,n)-ub(i,n))/(hbar(i+1,n)*dx);
end

% Memperbarui Kecepatan
for i=1:Nx-1
    if hb(i+1,n)>thres
        u(i+1,n+1)=u(i+1,n)-dt*g*(w(i+1,n+1)-w(i,n+1))/dx;
    else
        u(i+1,n+1)=0;
    end
end
end

```

```

% Plotting Hasil Simulasi
plot(x, w(1:Nx-1, n), 'b', 'DisplayName', 'Permukaan Air (w)') % Garis biru
untuk air
hold on
plot(x, b(1:Nx-1), 'Color', [0.7, 0.5, 0], 'DisplayName', 'Topografi Dasar (b)')
% Garis coklat untuk topografi
hold off
axis([0 L 0 1.5])
xlabel('x');
ylabel('h');
legend
title(['T= ', num2str(n * dt), ' s'])
drawnow
end

```

A.2. Program Octave: Bendungan Bobol di Atas Topografi Diskontinu dan Hambatan

```

% Inisialisasi Awal
clear all;
close all;
clc;

% Definisi Parameter
L=200;          % Panjang domain
T=120;          % Waktu simulasi
dx=1;           % Ukuran grid spasial
g=9.81;         % Gravitasi (m/s2)
dt=0.1;         % Waktu langkah (time step)
Nx=floor(L/dx); % Jumlah titik grid dalam ruang
Nt=floor(T/dt); % Jumlah iterasi waktu
x=0:dx:L-2*dx; % Array posisi dalam domain

```

```

% Inisialisasi Variabel
h=zeros(Nx,Nt); % Ketebalan air
u=zeros(Nx+1,Nt); % Kecepatan fluida
q=zeros(Nx+1,Nt); % Debit aliran (q = h*u)
thres=0.01; % Ambang batas tinggi air
Cf=0.1; % Koefisien gesekan

% kondisi awal
for i=1:Nx
    if i<floor(Nx/2)
        w(i,1)=1;
        b(i)=0.5;
    else
        w(i,1)=0.25;
        b(floor(1.0*Nx/2):floor(1.2*Nx/2))=0;
    end
    b(floor(1.2*Nx/2):floor(1.3*Nx/2))=0.2;
    b(floor(1.4*Nx/2):floor(1.5*Nx/2))=0.2;
    b(floor(1.6*Nx/2):floor(1.7*Nx/2))=0.2;
end

u(1:Nx+1,1)=0; % Kecepatan awal di semua titik adalah 0
b(Nx+1)=0; % kondisi batas topografi dasar di batas kanan

% Kondisi Batas untuk Kecepatan (u)
u(1,1:Nt)=0; % batas kiri
u(Nx+1,1:Nt)=0; % batas kanan

% Kondisi Batas untuk Debit Aliran (q)
q(1,1:Nt)=0; % batas kiri
q(Nx+1,1:Nt)=0; % batas kanan

```

```

% Kondisi Batas untuk Ketebalan Air (h)
h(1,1:Nt)=1;          % ketinggian tetap 1 di sisi kiri

for n=1:Nt            % Perhitungan dalam Loop Waktu
    % Menentukan Nilai Ketinggian untuk Debit
    for i=1:Nx-1
        if u(i+1,n)>0
            hb(i+1,n)=h(i,n); % Jika kecepatan positif, gunakan tinggi di titik
sebelumnya
        else
            hb(i+1,n)=h(i+1,n); % Jika kecepatan negatif, gunakan tinggi di titik
selanjutnya
        end
        hbar(i+1,n)=0.5*(h(i,n)+h(i+1,n)); % Rata-rata ketebalan air
    end

% Menghitung Debit Aliran Aliran
for i=2:Nx
    q(i,n)=hb(i,n)*u(i,n);
end

% Menghitung Rata-rata Debit dan Kecepatan
for i=1:Nx
    qbar(i,n)=0.5*(q(i+1,n)+q(i,n)); % Rata-rata debit Aliran
    if qbar(i,n)<0
        ub(i,n)=u(i+1,n);
    else
        ub(i,n)=u(i,n);
    end
    w(i,n+1)=w(i,n)-dt*(q(i+1,n)-q(i,n))/dx;
    h(i,n+1)=w(i,n+1)-b(i);

```

```

end

% Menghitung Perubahan Kecepatan
for i=1:Nx-1
    uux(i+1,n)=qbar(i,n)*(ub(i+1,n)-ub(i,n))/(hbar(i+1,n)*dx);
end

% Memperbarui Kecepatan
for i=1:Nx-1
    if hb(i+1,n)>thres
        u(i+1,n+1)=u(i+1,n)-dt*g*(w(i+1,n+1)-w(i,n+1))/dx-uux(i+1,n)-
Cf*u(i,n)*abs(u(i,n))*dt/h(i,n);
    else
        u(i+1,n+1)=0;
    end
end

% Plotting Hasil Simulasi
plot(x, w(1:Nx-1, n), 'b', 'DisplayName', 'Permukaan Air (w)') % Garis biru
untuk air
hold on
plot(x, b(1:Nx-1), 'Color', [0.7, 0.5, 0], 'DisplayName', 'Topografi Dasar (b)')
% Garis coklat untuk topografi
hold off
axis([0 L 0 1.5])
xlabel('x');
ylabel('h');
legend
title(['T= ', num2str(n * dt), ' s'])
drawnow
end

```