

## Fungsi 1

function y = fungsi(x)

y=3-5\*x^3-2\*sin(x)^2+cos(10\*x);

## Fungsi 2

function y = fungsi(x)

y= 3/5\*x^2-2\*sin(x)^2/5\*x^2+cos(10\*x)/5\*x^2;

## Metode Tabulasi

clear

clc

%Penentuan Nilai x1 dan x2

x1=0;

x2=1;

%Batasan nilai error

bne= 10^-1;

%Nilai sebenarnya

nilseb = 0.765237

%Kondisi nilai awal fungsi

fx0=1;

%Perhitungan Iterasi ke-

i=0;

%fungsi

fx1 = fungsi(x1);

fx2 = fungsi(x2);

x0 = 0;

if(fx1\*fx2>0); %Pengecekan Kondisi

disp('Tidak memenuhi kondisi')

disp('=====')

else

disp('=====')

```

for n = 1:10 %while(abs(fx0)>bne); %Loop Iterasi Ke-n
    disp('iterasi    x        fx        |fx|')
    n;
    i=i+1;
    u=x1:10^-i:x2;
    t=1:length(u);
    p=0;
    while(p<length(u)); %Loop mencari nilai x1 dan x2
        p=p+1;
        x=u(p);
        s=p+1;
        if(s>length(u));
            s=length(u);
        end
        x0=u(s);
        fx=fungsi(x);
        fx0=fungsi(x0);
        fprintf(' %d    %f    %f    %f\n',i,x0,fx0,abs(fx0));
        if(fx*fx0<0); %Penentuan ulang nilai x1 dan x2
            x1=x;
            x2=x0;
        end
    end
    p=0;
    disp('=====')
end
end
disp('Hasil data')
x0
abs(fx0);

```

```

Error = 100-x0/nilseb*100;
fprintf('Error = %s ',abs(Error)); disp('%')

```

## Metode Regula Falsi

```

clear
clc

```

```

%Penentuan Nilai x1 dan x2
x1=0;

```

```

x2=1;

%Batasan nilai error
bne=10^-2;

%Nilai sebenarnya
nilseb = 0.765237

%Kondisi nilai awal fungsi
fx0=1;

%Perhitungan Iterasi ke-
i=0;

%fungsi
fx1=fungsi(x1);
fx2=fungsi(x2);

x0 = 0;

if(fx1*fx2>0) %Pengecekan Kondisi
    disp('tidak memenuhi kondisi')
else
    disp('Iterasi ke-    x0          fx0')
    disp('=====')
    for n = 1:10 %while(abs(fx0)>bne);%Loop Iterasi Ke-n
        n;
        i=i+1;
        x0=x2-((fx2*(x2-x1))/(fx2-fx1)); %Mencari nilai x0
        fx0=fungsi(x0);
        fprintf('    %d        %f        %f\n',i,x0,fx0);
        if(fx1*fx0<0)
            x2=x0;
        else
            x1=x0;
        end
    end
end
disp('=====')
disp('Hasil data')
nil = x0

```

```
abs(fungsi(nil));
```

```
Error = 100-nil/nilseb*100;
```

```
fprintf('Error = %s ',abs(Error)); disp('%')
```

## **Metode Bentuk $x = g(x)$**

```
clear
```

```
clc
```

```
% Mengubah Fungsi menjadi bentuk  $g(x)$ 
```

```
x = sym ('x');
```

```
y = fungsi2(x);
```

```
fx= diff(y);
```

```
yaw(x)=fungsi(x);
```

```
yin(x)= y; %fungsi awal
```

```
g(x)=fx; %turunan fungsi
```

```
%Batas Nilai Error
```

```
bne=10^-5;
```

```
nfx = 1;
```

```
%Nilai sebenarnya
```

```
nilseb = 0.765237
```

```
%Kondisi nilai awal fungsi
```

```
fx0= 1/2 ;
```

```
%Perhitungan Iterasi ke-
```

```
i=0;
```

```
%Penentuan Nilai uji
```

```
x1 = 0.5;
```

```
%Fungsi
```

```
fx1=g(x1);%turunan
```

```
x2=0;
```

```
if(abs(fx1)>1) %Pengecekan Kondisi
```

```
    disp('tidak Memenuhi Syarat')
```

```
else
```

```

disp('Iterasi      x      fx  ')
disp('=====')
for n = 1:4 %while (abs(fungsi(x1))>bne); %n = 1:10 %Looping + Letak permasalahan
    n; %jika pake loop
    i=i+1;%Iterasi
    x2=fungsi2(x1);%mencari x baru dari fungsi awal
    fout=fungsi(x2); %mendapat fx
    fprintf('  %d      %f      %f\n',i,x1,fout);
    %i=iterasi,x1=nilai x, fout=nilai fx
    if(abs(fx1)<1);
        x1=x2;
    end
end
end

disp('=====')
disp('hasil')
x2
abs(fungsi(x2));

Error = 100-x2/nilseb*100;
fprintf('Error = %s ',abs(Error)); disp('%')

```

%NOTE : Fungsi Cos dan sin tidak support untuk Metode iterasi bentuk  $x=gx$