LAPORAN PRAKTIKUM METODE NUMERIK

DISUSUN OLEH :

NAMA : Ammar

NIM : 2024573010129

KELAS : TI 2B

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI DAN KOMPUTER

POLITEKNIK NEGERI LHOKSEUMAWE

2025/2026

LAPORAN PENGESAHAN

Nomor Praktikum	: 02
Judul Praktikum	: Metode Numerik Tertutup
Nama Praktikan	: Ammar
NIM	: 2024573010129
Jurusan	: Teknologi Informasi dan Komputer
Prodi	: Teknik Informatika
Tanggal Praktikum	: 8 Oktober 2025
Tanggal Penyerahan	: 29 Oktober 2025
Nilai	:
Keterangan	:

Buket Rata, 29 Oktober 2025

Dosen pengajar

Radhiyatammardhiyah, SST, M.Sc

NIP.197008021999031001

DAFTAR ISI

PENDAHULUAN

Dalam penyelesaian persoalan matematis, khususnya pencarian akar persamaan non-linear, sering kali metode analitik tidak dapat digunakan karena bentuk fungsi yang kompleks atau tidak memiliki solusi eksak. Oleh karena itu, dibutuhkan pendekatan numerik yang dapat memberikan hasil hampiran dengan tingkat ketelitian tertentu.

Metode numerik tertutup merupakan salah satu pendekatan yang digunakan untuk mencari akar fungsi dalam suatu interval tertentu dengan asumsi bahwa fungsi tersebut kontinu dan terdapat perubahan tanda nilai fungsi pada batas intervalnya. Dua metode tertutup yang paling umum digunakan adalah **metode Biseksi** dan **metode Regula Falsi**.

Metode Biseksi bekerja dengan cara membagi dua interval secara berulang untuk mempersempit daerah yang mengandung akar, sedangkan metode Regula Falsi memanfaatkan garis secant yang menghubungkan dua titik pada kurva fungsi untuk memperkirakan posisi akar yang lebih cepat mendekati nilai sebenarnya.

Dengan memanfaatkan metode tertutup ini, diharapkan pengguna dapat memperoleh pemahaman mengenai prinsip dasar pendekatan numerik dalam menentukan akar fungsi serta perbandingan efisiensi antara metode Biseksi dan Regula Falsi dalam hal konvergensi terhadap akar sebenarnya.

Buket Rata, 28 Oktober 2025

Ammar

BAB I
DAFTAR TEORI

1.1 Metode Tertutup

Metode tertutup merupakan metode pencarian akar persamaan yang dilakukan pada suatu interval tertentu [a, b] dengan syarat fungsi kontinu dan memiliki tanda yang berlawanan pada kedua ujung intervalnya, yaitu f(a) * f(b) < 0. Hal ini menunjukkan bahwa di antara titik a dan b terdapat minimal satu akar fungsi.

1.2 Metode Biseksi

Metode Biseksi adalah metode numerik sederhana yang mencari akar fungsi dengan cara membagi dua interval [a, b] secara berulang. Langkah dasarnya:

a) Tentukan dua titik awal a dan b sehingga f(a) dan f(b) memiliki tanda yang berlawanan.

b) Hitung titik tengah c = (a + b) / 2.

c) Hitung nilai f(c).

d) Jika f(a) * f(c) < 0, maka akar berada pada interval [a, c]; jika tidak, akar berada pada [c, b].

e) Proses ini diulangi sampai perbedaan antara a dan b (atau nilai f(c)) berada di bawah batas toleransi.

Kelebihan metode ini adalah **kesederhanaan dan kepastian konvergensi**, sedangkan kekurangannya adalah **konvergensi yang relatif lambat** dibanding metode lain.

1.3 Metode Regula Falsi

Metode Regula Falsi (False Position) juga mencari akar fungsi dalam interval `[a, b]`, tetapi menggunakan pendekatan garis lurus (secant) yang menghubungkan titik `(a, f(a))` dan `(b, f(b))`. Persamaan garis secant digunakan untuk memperkirakan posisi akar:

$$c = b - \frac{f(b)(b-a)}{f(b) - f(a)}$$

Langkah-langkahnya mirip dengan metode Biseksi, tetapi perkiraan titik akar `c` diambil dari hasil perpotongan garis secant, bukan titik tengah.

Kelebihan metode Regula Falsi adalah **konvergensi yang lebih cepat** dibanding Biseksi pada fungsi tertentu, namun dalam beberapa kasus bisa **melambat jika salah satu sisi interval tidak banyak berubah**.

1.4 Perbandingan Singkat

Aspek	**Biseksi**	**Regula Falsi**
Pendekatan	Titik tengah interval	Garis secant (interpolasi linear)
Konvergensi	Pasti konvergen tapi lambat	Lebih cepat, tapi tidak selalu stabil
Syarat awal	f(a) * f(b) < 0	f(a) * f(b) < 0
Kompleksitas	Sederhana	Sedikit lebih kompleks

BAB II
HASIL

2.1 Biseksi Selang kiri [-4, 0]

2.1.1 Program

```
from .. import abc

def f(x):
    return 2*x**2 + 3*x - 4 # Fungsi f(x)

# input awal interval
a = -4
b = 0
c = (a + b) / 2
toleransi = 0.00001
iterasi = 1
x2 = abc.x[1]

c_pref = 0
Era = 100

fa = f(a)
fb = f(b)
fc = f(c)

# Pastikan ada akar di antara a dan b
if f(a) * f(b) > 0:
    print("Tidak ada akar di antara interval [a,b]")

else:
    print("="*45 + "BISEKSI" + "="*45)
    print(f"{'Iterasi':<8}{'a':<10}{'b':<10}{'c':<13}{'f(a)':<10}{'f(b)':<10}{
'f(c)':<10}{'Interval baru':<18}{'Era'}")
    print("-"*98)
    print(f"{iterasi:<8}{a:<10.3f}{b:<10.3f}{c:<13.5f}{fa:<10.3f}{fb:<10.3f}{f
c:.5f}")

    while Era > toleransi:
        c = (a + b) / 2
        fa = f(a)
        fb = f(b)
        fc = f(c)
```

```
        tanda = f(a) * f(c)
        if tanda < 0:
            b = c
            interval = f"[{a:.3f}, {c:.3f}]"
            # interval = "[a, c]"
        else:
            a = c
            interval = f"[{c:.3f}, {b:.3f}]"
            # interval = "[c, b]"

        if c != 0:
            Era = abs((c - c_pref) / c)
        c_pref = c

        print(f"{iterasi:<8}{a:<10.3f}{b:<10.3f}{c:<13.5f}{fa:<10.3f}{fb:<10.3
f}{fc:<10.3f}{interval:<18}{Era:.6f}")
        iterasi += 1

    print("-"*98)
    akar = (a + b) / 2
    print(f"\nAkar hampiran = {akar:.6f}")
    print(f"Nilai Asli = {x2:.6f}")
```

2.1.2 Output

```
=============================================BISEKSI=============================================
Iterasi a        b        c           f(a)      f(b)      f(c)      Interval baru      Era
-------------------------------------------------------------------------------------------------
1       -4.000   0.000    -2.00000    16.000    -4.000    -2.00000
1       -4.000   -2.000   -2.00000    16.000    -4.000    -2.000    [-4.000, -2.000]  1.000000
2       -3.000   -2.000   -3.00000    16.000    -2.000    5.000     [-3.000, -2.000]  0.333333
3       -2.500   -2.000   -2.50000    5.000     -2.000    1.000     [-2.500, -2.000]  0.200000
4       -2.500   -2.250   -2.25000    1.000     -2.000    -0.625    [-2.500, -2.250]  0.111111
5       -2.375   -2.250   -2.37500    1.000     -0.625    0.156     [-2.375, -2.250]  0.052632
6       -2.375   -2.312   -2.31250    0.156     -0.625    -0.242    [-2.375, -2.312]  0.027027
7       -2.375   -2.344   -2.34375    0.156     -0.242    -0.045    [-2.375, -2.344]  0.013333
8       -2.359   -2.344   -2.35938    0.156     -0.045    0.055     [-2.359, -2.344]  0.006623
9       -2.352   -2.344   -2.35156    0.055     -0.045    0.005     [-2.352, -2.344]  0.003322
10      -2.352   -2.348   -2.34766    0.005     -0.045    -0.020    [-2.352, -2.348]  0.001664
11      -2.352   -2.350   -2.34961    0.005     -0.020    -0.007    [-2.352, -2.350]  0.000831
12      -2.352   -2.351   -2.35059    0.005     -0.007    -0.001    [-2.352, -2.351]  0.000415
13      -2.351   -2.351   -2.35107    0.005     -0.001    0.002     [-2.351, -2.351]  0.000208
14      -2.351   -2.351   -2.35083    0.002     -0.001    0.000     [-2.351, -2.351]  0.000104
15      -2.351   -2.351   -2.35071    0.000     -0.001    -0.000    [-2.351, -2.351]  0.000052
16      -2.351   -2.351   -2.35077    0.000     -0.000    -0.000    [-2.351, -2.351]  0.000026
17      -2.351   -2.351   -2.35080    0.000     -0.000    0.000     [-2.351, -2.351]  0.000013
18      -2.351   -2.351   -2.35078    0.000     -0.000    0.000     [-2.351, -2.351]  0.000006
-------------------------------------------------------------------------------------------------

Akar hampiran = -2.350777
```

2.1.3 Analisis

2.2 Biseksi Selang Kanan [0, 4]

2.2.1 Program

```
from .. import abc

def f(x):
    return 2*x**2 + 3*x - 4 # Fungsi f(x)

# input awal interval
a = 0
b = 4
c = (a + b) / 2
toleransi = 0.00001
iterasi = 1
x1 = abc.x[0]

c_pref = 0
Era = 100

fa = f(a)
fb = f(b)
fc = f(c)

# Pastikan ada akar di antara a dan b
if f(a) * f(b) > 0:
    print("Tidak ada akar di antara interval [a,b]")
```

```
else:
    print("="*45 + "BISEKSI" + "="*45)
    print(f"{'Iterasi':<8}{'a':<10}{'b':<10}{'c':<13}{'f(a)':<10}{'f(b)':<10}{
'f(c)':<10}{'Interval baru':<18}{'Era'}")
    print("-"*98)
    print(f"{iterasi:<8}{a:<10.3f}{b:<10.3f}{c:<13.5f}{fa:<10.3f}{fb:<10.3f}{f
c:.5f}")

    while Era > toleransi:
        c = (a + b) / 2
        fa = f(a)
        fb = f(b)
        fc = f(c)

        tanda = f(a) * f(c)
        if tanda < 0:
            interval = f"[{a:.3f}, {c:.3f}]"
            # interval = "[a, c]"
            b = c
        else:
            interval = f"[{c:.3f}, {b:.3f}]"
            # interval = "[c, b]"
            a = c

        if c != 0:
            Era = abs((c - c_pref) / c)
        c_pref = c

        print(f"{iterasi:<8}{a:<10.3f}{b:<10.3f}{c:<13.5f}{fa:<10.3f}{fb:<10.3
f}{fc:<10.3f}{interval:<18}{Era:.6f}")
        iterasi += 1

    print("-"*98)
    akar = (a + b) / 2
    print(f"\nAkar hampiran = {akar:.4f}")
    print(f"Nilai Asli = {x1:.4f}")
```

2.2.2 Output

```
=========================================BISEKSI=========================================
Iterasi a        b        c           f(a)      f(b)      f(c)      Interval baru     Era
-----------------------------------------------------------------------------------------
1       0.000    4.000    2.00000     -4.000    40.000    10.00000
1       0.000    2.000    2.00000     -4.000    40.000    10.000    [0.000, 2.000]    1.000000
2       0.000    1.000    1.00000     -4.000    10.000    1.000     [0.000, 1.000]    1.000000
3       0.500    1.000    0.50000     -4.000    1.000     -2.000    [0.500, 1.000]    1.000000
4       0.750    1.000    0.75000     -2.000    1.000     -0.625    [0.750, 1.000]    0.333333
5       0.750    0.875    0.87500     -0.625    1.000     0.156     [0.750, 0.875]    0.142857
6       0.812    0.875    0.81250     -0.625    0.156     -0.242    [0.812, 0.875]    0.076923
7       0.844    0.875    0.84375     -0.242    0.156     -0.045    [0.844, 0.875]    0.037037
8       0.844    0.859    0.85938     -0.045    0.156     0.055     [0.844, 0.859]    0.018182
9       0.844    0.852    0.85156     -0.045    0.055     0.005     [0.844, 0.852]    0.009174
10      0.848    0.852    0.84766     -0.045    0.005     -0.020    [0.848, 0.852]    0.004608
11      0.850    0.852    0.84961     -0.020    0.005     -0.007    [0.850, 0.852]    0.002299
12      0.851    0.852    0.85059     -0.007    0.005     -0.001    [0.851, 0.852]    0.001148
13      0.851    0.851    0.85107     -0.001    0.005     0.002     [0.851, 0.851]    0.000574
14      0.851    0.851    0.85083     -0.001    0.002     0.000     [0.851, 0.851]    0.000287
15      0.851    0.851    0.85071     -0.001    0.000     -0.000    [0.851, 0.851]    0.000143
16      0.851    0.851    0.85077     -0.000    0.000     -0.000    [0.851, 0.851]    0.000072
17      0.851    0.851    0.85080     -0.000    0.000     0.000     [0.851, 0.851]    0.000036
18      0.851    0.851    0.85078     -0.000    0.000     0.000     [0.851, 0.851]    0.000018
19      0.851    0.851    0.85078     -0.000    0.000     -0.000    [0.851, 0.851]    0.000009
-----------------------------------------------------------------------------------------

Akar hampiran = 0.850780
Nilai Asli = 0.850781
```

2.2.3 Analisis

2.3 Regula falsi Selang kiri [-4, 0]

2.3.1 Program

```
from .. import abc

def f(x):
    return 2*x**2 + 3*x - 4 # Fungsi f(x)

# input awal interval
a = -4
b = 0
# c = a + b / 2
toleransi = 0.00001
iterasi = 1
x2 = abc.x[1]

c_pref = 0
Era = 100

fa = f(a)
fb = f(b)
c = b - (fb * (b - a)) / (fb - fa)
fc = f(c)

# Pastikan ada akar di antara a dan b
if f(a) * f(b) > 0:
    print("Tidak ada akar di antara interval [a,b]")

else:
    print("="*45 + "REGULAFALSI" + "="*45)
    print(f"{'Iterasi':<8}{'a':<10}{'b':<10}{'c':<13}{'f(a)':<10}{'f(b)':<10}{
'f(c)':<10}{'Interval baru':<18}{'Era'}")
    print("-"*98)
    print(f"{iterasi:<8}{a:<10.3f}{b:<10.3f}{c:<13.5f}{fa:<10.3f}{fb:<10.3f}{f
c:.5f}")

    while Era > toleransi:
        fa = f(a)
        fb = f(b)
        c = b - (fb * (b - a)) / (fb - fa)
        fc = f(c)

        tanda = f(a) * f(c)
        if tanda < 0:
            interval = f"[{a:.3f}, {c:.3f}]"
            # interval = "[a, c]"
            b = c
```

```
        else:
            interval = f"[{c:.3f}, {b:.3f}]"
            # interval = "[c, b]"
            a = c

        if c != 0:
            Era = abs((c - c_pref) / c)
        c_pref = c

        print(f"{iterasi:<8}{a:<10.3f}{b:<10.3f}{c:<13.5f}{fa:<10.3f}{fb:<10.3
f}{fc:<10.3f}{interval:<18}{Era:.6f}")
        iterasi += 1

    print("-"*98)
    akar = b - (fb * (b - a)) / (fb - fa)
    print(f"\nAkar hampiran = {akar:.6f}")
    print(f"Nilai Asli = {x2:.6f}")
```

2.3.2 Output

```
==========================================REGULAFALSI==========================================
Iterasi a        b        c           f(a)     f(b)     f(c)     Interval baru     Era
------------------------------------------------------------------------------------------------
1       -4.000   0.000    -0.80000    16.000   -4.000   -5.12000
1       -4.000   -0.800   -0.80000    16.000   -4.000   -5.120   [-4.000, -0.800]  1.000000
2       -4.000   -1.576   -1.57576    16.000   -5.120   -3.761   [-4.000, -1.576]  0.492308
3       -4.000   -2.037   -2.03717    16.000   -3.761   -1.811   [-4.000, -2.037]  0.226499
4       -4.000   -2.237   -2.23679    16.000   -1.811   -0.704   [-4.000, -2.237]  0.089241
5       -4.000   -2.311   -2.31109    16.000   -0.704   -0.251   [-4.000, -2.311]  0.032151
6       -4.000   -2.337   -2.33718    16.000   -0.251   -0.087   [-4.000, -2.337]  0.011160
7       -4.000   -2.346   -2.34614    16.000   -0.087   -0.030   [-4.000, -2.346]  0.003822
8       -4.000   -2.349   -2.34920    16.000   -0.030   -0.010   [-4.000, -2.349]  0.001303
9       -4.000   -2.350   -2.35024    16.000   -0.010   -0.003   [-4.000, -2.350]  0.000443
10      -4.000   -2.351   -2.35060    16.000   -0.003   -0.001   [-4.000, -2.351]  0.000151
11      -4.000   -2.351   -2.35072    16.000   -0.001   -0.000   [-4.000, -2.351]  0.000051
12      -4.000   -2.351   -2.35076    16.000   -0.000   -0.000   [-4.000, -2.351]  0.000017
13      -4.000   -2.351   -2.35077    16.000   -0.000   -0.000   [-4.000, -2.351]  0.000006
------------------------------------------------------------------------------------------------

Akar hampiran = -2.350788
Nilai Asli = -2.350781
```

2.3.3 Analisis

2.4 Regula falsi Selang kanan [0, 4]

2.4.1 Program

```
from .. import abc

def f(x):
    return 2*x**2 + 3*x - 4 # Fungsi f(x)

# input awal interval
a = 0
b = 4
# c = a + b / 2
toleransi = 0.00001
iterasi = 1
x1 = abc.x[0]

c_pref = 0
Era = 100

fa = f(a)
fb = f(b)
c = b - (fb * (b - a)) / (fb - fa)
fc = f(c)

# Pastikan ada akar di antara a dan b
if f(a) * f(b) > 0:
    print("Tidak ada akar di antara interval [a,b]")

else:
    print("="*42 + "REGULAFALSI" + "="*45)
    print(f"{'Iterasi':<8}{'a':<10}{'b':<10}{'c':<13}{'f(a)':<10}{'f(b)':<10}{
'f(c)':<10}{'Interval baru':<18}{'Era'}")
```

```
    print("-"*98)
    print(f"{iterasi:<8}{a:<10.3f}{b:<10.3f}{c:<13.5f}{fa:<10.3f}{fb:<10.3f}{f
c:.5f}")

    while Era > toleransi:
        fa = f(a)
        fb = f(b)
        c = b - (fb * (b - a)) / (fb - fa)
        fc = f(c)

        tanda = f(a) * f(c)
        if tanda < 0:
            interval = f"[{a:.3f}, {c:.3f}]"
            # interval = "[a, c]"
            b = c
        else:
            interval = f"[{c:.3f}, {b:.3f}]"
            # interval = "[c, b]"
            a = c

        if c != 0:
            Era = abs((c - c_pref) / c)
        c_pref = c

        print(f"{iterasi:<8}{a:<10.3f}{b:<10.3f}{c:<13.5f}{fa:<10.3f}{fb:<10.3
f}{fc:<10.3f}{interval:<18}{Era:.6f}")
        iterasi += 1

    print("-"*98)
    akar = b - (fb * (b - a)) / (fb - fa)
    print(f"\nAkar hampiran = {akar:.6f}")
    print(f"Nilai Asli = {x1:.6f}")
```

2.4.2 Program

```
=========================================REGULAFALSI=========================================
Iterasi a        b        c           f(a)      f(b)      f(c)      Interval baru     Era
-----------------------------------------------------------------------------------------
1       0.000    4.000    0.36364     -4.000    40.000    -2.64463
1       0.364    4.000    0.36364     -4.000    40.000    -2.645    [0.364, 4.000]    1.000000
2       0.589    4.000    0.58915     -2.645    40.000    -1.538    [0.589, 4.000]    0.382775
3       0.715    4.000    0.71547     -1.538    40.000    -0.830    [0.715, 4.000]    0.176557
4       0.782    4.000    0.78222     -0.830    40.000    -0.430    [0.782, 4.000]    0.085338
5       0.816    4.000    0.81641     -0.430    40.000    -0.218    [0.816, 4.000]    0.041880
6       0.834    4.000    0.83365     -0.218    40.000    -0.109    [0.834, 4.000]    0.020672
7       0.842    4.000    0.84226     -0.109    40.000    -0.054    [0.842, 4.000]    0.010229
8       0.847    4.000    0.84655     -0.054    40.000    -0.027    [0.847, 4.000]    0.005067
9       0.849    4.000    0.84868     -0.027    40.000    -0.013    [0.849, 4.000]    0.002511
10      0.850    4.000    0.84974     -0.013    40.000    -0.007    [0.850, 4.000]    0.001245
11      0.850    4.000    0.85026     -0.007    40.000    -0.003    [0.850, 4.000]    0.000617
12      0.851    4.000    0.85052     -0.003    40.000    -0.002    [0.851, 4.000]    0.000306
13      0.851    4.000    0.85065     -0.002    40.000    -0.001    [0.851, 4.000]    0.000152
14      0.851    4.000    0.85072     -0.001    40.000    -0.000    [0.851, 4.000]    0.000075
15      0.851    4.000    0.85075     -0.000    40.000    -0.000    [0.851, 4.000]    0.000037
16      0.851    4.000    0.85077     -0.000    40.000    -0.000    [0.851, 4.000]    0.000019
17      0.851    4.000    0.85077     -0.000    40.000    -0.000    [0.851, 4.000]    0.000009
-----------------------------------------------------------------------------------------

Akar hampiran = 0.850781
Nilai Asli = 0.850781
```

2.4.3 Analisis

BAB III
PENUTUP

3.1 Kesimpulan

3.2 Daftar Pustaka

3.2.1 Chapra, Steven C. dan Raymond P. Canale. *Numerical Methods for Engineers*. 8th Edition. New York: McGraw-Hill Education, 2020.