**LAPORAN PRAKTIKUM METODE NUMERIK**

**A green and white logo

AI-generated content may be incorrect.**

DISUSUN OLEH :

NAMA : Ammar

NIM : 2024573010129

KELAS : TI 2B

A yellow and purple logo

AI-generated content may be incorrect.

**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

[LAPORAN PENGESAHAN ii](#_Toc212559273)

[DAFTAR ISI iii](#_Toc212559274)

[PENDAHULUAN iv](#_Toc212559275)

[BAB I DAFTAR TEORI 1](#_Toc212559276)

[1.1 Metode Tertutup 1](#_Toc212559277)

[1.2 Metode Biseksi 1](#_Toc212559278)

[1.3 Metode Regula Falsi 2](#_Toc212559279)

[1.4 Perbandingan Singkat 2](#_Toc212559280)

[BAB II HASIL 3](#_Toc212559281)

[2.1 Biseksi Selang kiri [-4, 0] 3](#_Toc212559282)

[2.1.1 Program 3](#_Toc212559283)

[2.1.2 Output 4](#_Toc212559284)

[2.1.3 Analisis 5](#_Toc212559285)

[2.2 Biseksi Selang Kanan [0, 4] 5](#_Toc212559286)

[2.2.1 Program 5](#_Toc212559287)

[2.2.2 Output 7](#_Toc212559288)

[2.2.3 Analisis 7](#_Toc212559289)

[2.3 Regula falsi Selang kiri [-4, 0] 8](#_Toc212559290)

[2.3.1 Program 8](#_Toc212559291)

[2.3.2 Output 9](#_Toc212559292)

[2.3.3 Analisis 10](#_Toc212559293)

[2.4 Regula falsi Selang kanan [0, 4] 10](#_Toc212559294)

[2.4.1 Program 10](#_Toc212559295)

[2.4.2 Program 12](#_Toc212559296)

[2.4.3 Analisis 12](#_Toc212559297)

[BAB III PENUTUP 13](#_Toc212559298)

[3.1 Kesimpulan 13](#_Toc212559299)

[3.2 Daftar Pustaka 13](#_Toc212559300)

[3.2.1 Chapra, Steven C. dan Raymond P. Canale. *Numerical Methods for Engineers*. 8th Edition. New York: McGraw-Hill Education, 2020. 13](#_Toc212559301)

# 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

# DAFTAR TEORI

## 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.

## Metode Biseksi

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

1. Tentukan dua titik awal a dan b sehingga f(a) dan f(b) memiliki tanda yang berlawanan.
2. Hitung titik tengah c = (a + b) / 2.
3. Hitung nilai f(c).
4. Jika f(a) \* f(c) < 0, maka akar berada pada interval [a, c]; jika tidak, akar berada pada [c, b].
5. 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.

## 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:

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**.

## 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 |

# HASIL

## Biseksi Selang kiri [-4, 0]

### 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}{fc:.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:<10.3f}{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}")

### Output

A screenshot of a computer

AI-generated content may be incorrect.

### Analisis

## Biseksi Selang Kanan [0, 4]

### 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}{fc:.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:<10.3f}{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}")

### Output

A screenshot of a computer screen

AI-generated content may be incorrect.

### Analisis

## Regula falsi Selang kiri [-4, 0]

### 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}{fc:.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:<10.3f}{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}")

### Output

A screenshot of a computer

AI-generated content may be incorrect.

### Analisis

## Regula falsi Selang kanan [0, 4]

### 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}{fc:.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:<10.3f}{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}")

### Program

A screenshot of a computer

AI-generated content may be incorrect.

### Analisis

# PENUTUP

## Kesimpulan

## Daftar Pustaka

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