

Laporan Praktikum 1

Komputasi Numerik B – Kelompok 17

William Hans Chandra 5025241138

Yoseph Kevin Hendrata 5025241146

Maulana Ikhsan 5025241163

Soal

Anda sudah mengerti algoritma pemrosesan metode Regula Falsi, dan anda sudah memahami cara kerjanya. Sekarang anda tinggal mengimplementasikan algoritma tersebut menjadi sebuah program komputer metode Regula Falsi (yang dapat menampilkan proses iteratif numerik, lengkap dengan grafik fungsinya).

Ide Penyelesaian :

- Menggunakan python dengan berbagai library yang dapat digunakan.
- Meng-import semua library yang dibutuhkan untuk pengerjaan.
- Membuat function `user_func` untuk mengkonversi inputan fungsi dari user ke ekspresi matematika dalam program.
- Membuat function `regula_falsi` untuk melakukan perhitungan.
 - a. Jika nilai $f(a) \times f(b) \geq 0$ atau bernilai positif maka fungsi tidak bisa dikerjakan karena nilai $f(a)$ dan $f(b)$ harus berbeda tanda (satu negatif dan satu positif yang menandakan adanya perpotongan di sumbu x).
 - b. Melakukan iterasi looping untuk mendapatkan c dengan rumus.
 - c. Memasukkan data ke dalam array untuk nantinya dipublish dalam bentuk tabel.
 - d. Melakukan update dengan ketentuan:
 - 1. Jika nilai $f(c)$ sudah berada di bawah 10^{-6} maka hentikan iterasi.
 - 2. Update batas b dengan nilai c jika $f(a) \times f(c)$ negatif.
 - 3. Update batas a dengan nilai c jika $f(a) \times f(c)$ positif.
 - e. Return nilai c dan data tabel.
- Membuat function `plot_function` untuk menampilkan grafik fungsi dengan tanda tambahan untuk melihat titik tempat hasil akar aproksimasi regula falsi terletak.
- Membuat function `main` dimana:
 - a. Membuat panduan untuk menulis fungsi agar user dapat melakukan input dengan benar (program tidak error).
 - b. Meminta inputan fungsi, batas bawah a dan batas atas b dari user.
 - c. Mengkonversi input fungsi user dengan memanggil `user_func`.
 - d. Memanggil function `regula_falsi` untuk memperoleh hasil akar aproksimasi dan data tabel iterasi.
 - e. Menampilkan jumlah iterasi, nilai akar aproksimasi, tabel data iterasi, dan plot grafik.
- Untuk menjalankan code, dapat run `main()`.

Code :

```
import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

from sympy import symbols, lambdify, sympify, pi, sin, cos, tan, exp, log, ln, sqrt

MAX_ITERATION = 1000000

def user_func(user_input):

    x = symbols('x')

    expr = sympify(user_input)

    return lambdify(x, expr)

def regula_falsi(f, a, b):

    if f(a) * f(b) >= 0:

        raise ValueError("f(a) dan f(b) harus memiliki tanda yang berbeda")

    data = []

    c = a

    for i in range(MAX_ITERATION):

        fa = f(a)

        fb = f(b)

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

        fc = f(c)

        data.append((i+1, a, b, c, fc))

        if abs(fc) < 1e-6:

            break

        if fa * fc < 0:

            b = c

        else:

            a = c

    return c, data

def plot_function(f, user_input, a, b, akar=None):

    x_vals = np.linspace(a, b, 400)

    y_vals = f(x_vals)

    fig, ax = plt.subplots(figsize=(10, 5))
```

```

ax.plot(x_vals, y_vals, label=f'f(x) = {user_input}')
ax.axhline(0, color='black', linewidth=0.5)

if akar is not None:
    ax.axvline(akar, color='red', linestyle='--', label=f'Akar aproksimasi: {akar:.6f}')

ax.set_title("Metode Regula Falsi")
ax.set_xlabel("x")
ax.set_ylabel("f(x)")
ax.grid(True)
ax.legend()

return fig

```

```
def main():
```

```

    print("=== Panduan Penulisan Fungsi f(x) ===")
    print("Gunakan simbol matematika standar:")
    print("- Pangkat: gunakan '**')
    print("- Perkalian: gunakan '*'")
    print("- Akar: gunakan 'sqrt(x)')
    print("- Logaritma natural: gunakan 'ln(x)')
    print("- Eksponensial: gunakan 'exp(x)')
    print("- Fungsi lain yang didukung: sin(x), cos(x), tan(x), log(x, base)")
    print("Contoh input valid: x**3 - 2*x + 1, exp(x) - 4, ln(x + 1), sqrt(x)")
    print("=====")

```

```

    user_input = input("Masukkan fungsi f(x): ")
    a = float(input("Masukkan nilai a: "))
    b = float(input("Masukkan nilai b: "))
    f = user_func(user_input)
    akar, iterasi_data = regula_falsi(f, a, b)

```

```

    df = pd.DataFrame(iterasi_data, columns=["Iterasi", "a", "b", "c", "f(c)"])
    fig = plot_function(f, user_input, a, b, akar)
    print(f"Total Iterasi: {len(df)}")
    print(f"Akar aproksimasi: {akar:.6f}")

    return df, fig

```

```
main()
```

Test Case, Input Output

Input 0

$$x^3 - x^2 + 2$$

a = -2

b = 2

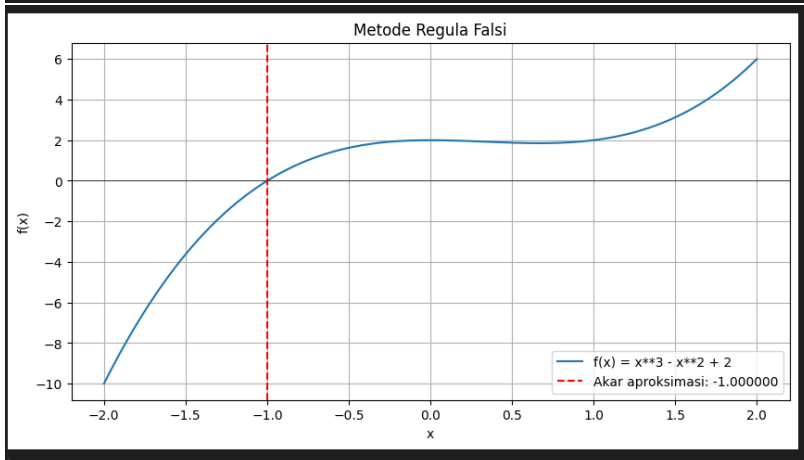
```
=== Panduan Penulisan Fungsi f(x) ===  
Gunakan simbol matematika standar:  
- Pangkat: gunakan '**'  
- Perkalian: gunakan '*'  
- Akar: gunakan 'sqrt(x)'  
- Logaritma natural: gunakan 'ln(x)'  
- Eksponensial: gunakan 'exp(x)'  
- Fungsi lain yang didukung: sin(x), cos(x), tan(x), log(x, base)  
Contoh input valid: x**3 - 2*x + 1, exp(x) - 4, ln(x + 1), sqrt(x)
```

Total Iterasi: 26

Akar aproksimasi: -1.000000

(Iterasi	a	b	c	f(c)
0	1	-2.0	2.000000	0.500000	1.875000e+00
1	2	-2.0	0.500000	0.105263	1.990086e+00
2	3	-2.0	0.105263	-0.244163	1.925828e+00
3	4	-2.0	-0.244163	-0.527703	1.574581e+00
4	5	-2.0	-0.527703	-0.727991	1.084216e+00
5	6	-2.0	-0.727991	-0.852414	6.540188e-01
6	7	-2.0	-0.852414	-0.922861	3.623533e-01
7	8	-2.0	-0.922861	-0.960526	1.911967e-01
8	9	-2.0	-0.960526	-0.980028	9.827262e-02
9	10	-2.0	-0.980028	-0.989954	4.982757e-02
10	11	-2.0	-0.989954	-0.994962	2.508954e-02
11	12	-2.0	-0.994962	-0.997477	1.258908e-02
12	13	-2.0	-0.997477	-0.998738	6.305665e-03
13	14	-2.0	-0.998738	-0.999369	3.155620e-03
14	15	-2.0	-0.999369	-0.999684	1.578507e-03
15	16	-2.0	-0.999684	-0.999842	7.894282e-04
16	17	-2.0	-0.999842	-0.999921	3.947577e-04
17	18	-2.0	-0.999921	-0.999961	1.973898e-04
18	19	-2.0	-0.999961	-0.999980	9.869762e-05
19	20	-2.0	-0.999980	-0.999990	4.934949e-05
20	21	-2.0	-0.999990	-0.999995	2.467492e-05
21	22	-2.0	-0.999995	-0.999998	1.233750e-05
22	23	-2.0	-0.999998	-0.999999	6.168761e-06
23	24	-2.0	-0.999999	-0.999999	3.084383e-06
24	25	-2.0	-0.999999	-1.000000	1.542192e-06
25	26	-2.0	-1.000000	-1.000000	7.710963e-07

<Figure size 1000x500 with 1 Axes>



Input 1

$\ln(x) - 2$

a = 1

b = 10

```
=== Panduan Penulisan Fungsi f(x) ===
Gunakan simbol matematika standar:
- Pangkat: gunakan '**'
- Perkalian: gunakan '*'
- Akar: gunakan 'sqrt(x)'
- Logaritma natural: gunakan 'ln(x)'
- Eksponensial: gunakan 'exp(x)'
- Fungsi lain yang didukung: sin(x), cos(x), tan(x), log(x, base)
Contoh input valid: x**3 - 2*x + 1, exp(x) - 4, ln(x + 1), sqrt(x)
=====
Total Iterasi: 23
Akar aproksimasi: 7.389061

( Iterasi  a      b      c      f(c)
0      1  1.0  10.000000  8.817301  1.767158e-01
1      2  1.0   8.817301  8.182656  1.020168e-01
2      3  1.0   8.182656  7.834062  5.848111e-02
3      4  1.0   7.834062  7.639907  3.338544e-02
4      5  1.0   7.639907  7.530889  1.901307e-02
5      6  1.0   7.530889  7.469387  1.081298e-02
6      7  1.0   7.469387  7.434599  6.144611e-03
7      8  1.0   7.434599  7.414890  3.490174e-03
8      9  1.0   7.414890  7.403715  1.981929e-03
9     10  1.0   7.403715  7.397376  1.125293e-03
10    11  1.0   7.397376  7.393778  6.388623e-04
11    12  1.0   7.393778  7.391736  3.626839e-04
12    13  1.0   7.391736  7.390578  2.058912e-04
13    14  1.0   7.390578  7.389920  1.168801e-04
14    15  1.0   7.389920  7.389546  6.634979e-05
15    16  1.0   7.389546  7.389334  3.766487e-05
16    17  1.0   7.389334  7.389214  2.138121e-05
17    18  1.0   7.389214  7.389146  1.213744e-05
18    19  1.0   7.389146  7.389107  6.890043e-06
19    20  1.0   7.389107  7.389085  3.911257e-06
20    21  1.0   7.389085  7.389073  2.220295e-06
21    22  1.0   7.389073  7.389065  1.260390e-06
22    23  1.0   7.389065  7.389061  7.154826e-07,
<Figure size 1000x500 with 1 Axes>)
```

