PROYEK 2

Metode Numerik (A)

Iterative Methods

C14210022 - Calvin Ananthapindika L
C14210016 - Michael Adi Pratama
C14210104 - Timothy Jeremy
C14210251 - Markus Nugraha. S
C14210144 - Billy Emanoel

1. **Gauss-Seidel**

Anggota Kelompok Metode Numerik:

1. Michael Adi Pratama - C14210016

2. Calvin Ananthapindika L - C14210022

3. Timothy Jeremy - C14210104

4. Markus Nugraha S. - C14210251

5. Billy Emanoel - c14210144

Gauss-Seidel

```
[32] import numpy as np
     def gauss_seidel(koef_matrix, konstanta, initial, max_iter=100, error_tolerance=0.000001):
         n = len(koef_matrix)
         x = np.array(initial, dtype=float)
         koef_matrix = np.array(koef_matrix, dtype=float)
         konstanta = np.array(konstanta, dtype=float)

         for _ in range(max_iter):
             for i in range(n):
                 sum_ = np.dot(koef_matrix[i, :i], x[:i]) + np.dot(koef_matrix[i, i + 1:], x[i + 1:])
                 x[i] = (konstanta[i] - sum_) / koef_matrix[i, i]

             # Periksa kekonvergenan
             if np.all(np.abs(np.dot(koef_matrix, x) - konstanta) < error_tolerance):
                 return x
         return x
```

Contoh Penggunaan Gauss-Seidel

```
# Contoh penggunaan
A = [[4, 1, 2],
     [3, 5, 1],
     [1, 1, 3]]
b = [4, 7, 3]
initial = [0, 0, 0]

hasil = gauss_seidel(A, b, initial)
print("Solusi Gauss-Seidel:", hasil)
```

```
Solusi Gauss-Seidel: [0.50000026 0.9999999  0.49999995]
```

2. Jacobi

Jacobi

[28]
```
def jacobi(koef_matrix, konstanta, initial, max_iter=100, error_tolerance=0.000001):
    pjg_arr = len(koef_matrix)
    z = initial.copy()
    for iterasi in range(max_iter):
        z_new = z.copy()
        for a in range(pjg_arr):
            total_val = 0
            for b in range(pjg_arr):
                if b != a:
                    total_val += koef_matrix[a][b] * z[b]
            z_new[a] = (konstanta[a] - total_val) / koef_matrix[a][a]
        if all(abs(z_new[a] - z[a]) < error_tolerance for a in range(pjg_arr)):
            return z_new
        z = z_new
    return z
```

Contoh Penggunaan Jacobi

```
# Contoh penggunaan
koef_matrix = [[4, 1, -1],
     [2, 7, 1],
     [1, 3, 5]]
konstanta = [3, 19, 71]
initial = [0, 0, 0]


hasil = jacobi(koef_matrix, konstanta, initial)

print("Solusi:")
for c, nilai in enumerate(hasil):
    print(f"x{c+1} = {nilai}")
```

```
Solusi:
x1 = 4.266665924011477
x2 = -0.4499993134184308
x3 = 13.616666061600474
```

3. Newton-Raphson

Newton-Raphson

```
[30] import sympy as sp

     def newton_raphson(persamaan, x, x0=2.5, max_iterations=100, error_tolerance=0.000001):
         f_prime = f.diff(x)
         x_n = x0
         for iteration in range(max_iterations):
             f_val = f.subs(x, x_n)
             f_prime_val = f_prime.subs(x, x_n)
             x_new = x_n - f_val / f_prime_val
             if abs(x_new - x_n) < error_tolerance:
                 return x_new
             x_n = x_new
         return x_n
```

Contoh Penggunaan Newton-Raphson

```
# Contoh penggunaan
x = sp.symbols('x')
persamaan = x**3 - 2*x - 5

result = newton_raphson(persamaan, x)
print("Aproksimasi akar:")
print(result)
```

```
Aproksimasi akar:
2.09455148154233
```