

## Tugas Metnum codingan

```
1 import sympy as sp
2
3 # Fungsi bisection untuk menemukan akar persamaan
4 def bisection(f, a, b, tol):
5     x = sp.Symbol('x') # Variabel simbolik
6     f_lambdified = sp.lambdify(x, f) # Konversi fungsi ke bentuk yang bisa dievaluasi secara numerik
7
8     # Pastikan bahwa f(a) dan f(b) memiliki tanda yang berbeda
9     if f_lambdified(a) * f_lambdified(b) >= 0:
10         print("Metode bisection tidak dapat digunakan.")
11         return None
12
13     # Inisialisasi iterasi
14     c = a
15     while (b - a) / 2.0 > tol:
16         # Hitung nilai titik tengah
17         c = (a + b) / 2.0
18
19         # Jika f(c) adalah akar, atau jika intervalnya sudah cukup kecil, maka selesai
20         if f_lambdified(c) == 0 or (b - a) / 2.0 < tol:
21             return c
22
23         # Tentukan subinterval baru untuk iterasi berikutnya
24         if f_lambdified(c) * f_lambdified(a) < 0:
25             b = c
26         else:
27             a = c
28
29     return c
30
31 # Input persamaan dari pengguna
32 persamaan_input = input("Masukkan persamaan nirlanjar (misal: x**3 - x - 2): ")
33 f = sp.sympify(persamaan_input) # Mengubah string input menjadi persamaan simbolik
34
35 # Input batas interval dan toleransi
36 a = float(input("Masukkan batas bawah interval: "))
37 b = float(input("Masukkan batas atas interval: "))
38 tol = float(input("Masukkan toleransi error: "))
39
40 # Mencari akar dengan metode bisection
41 akar = bisection(f, a, b, tol)
42 if akar is not None:
43     print(f"Akar yang ditemukan: {akar:.6f}")
```

Hasil dan contoh run:

```
Masukkan persamaan nirlanjar (misal: x**3 - x - 2): x**3 - x - 2
Masukkan batas bawah interval: 1
Masukkan batas atas interval: 2
Masukkan toleransi error: 1e-6
Akar yang ditemukan: 1.521379
```