Implementasi Bahasa Pemrograman Python sebagai Solusi Persamaan Nirlanjar dan Integrasi Numerik

Nazla Salsabila

Program Studi Informatika, Fakultas Teknik, Universitas Mulawarman
Jl. Kuaro, Gn. Kelua, Kec. Samarinda Ulu, Kota Samarinda, Kalimantan Timur 75119
nazlasalsabila171@gmail.com

***Intisari*— Penerapan bahasa pemrograman Python dalam penyelesaian sistem persamaan nirlanjar dan integrasi numerik telah menunjukkan efisiensi dan fleksibilitas yang signifikan. Jurnal ini menganalisis pemanfaat Python melalui library numerik seperti NumPy, SciPy, dan SymPy untuk menyelesaikan permasalahan matematis yang tidak dapat dipecahkan secara analitik. Metode numerik merupakan pendekatan yang sangat penting dalam menyelesaikan berbagai permasalahan matematika yang tidak dapat diselesaikan secara analitik, khususnya dalam pencarian akar persamaan non-linear dan perhitungan integral. Penelitian atau kajian ini bertujuan untuk membandingkan efektivitas dan akurasi beberapa metode numerik populer. Dalam setiap metode, dilakukan analisis langkah-langkah komputasi, hasil aproksimasi, serta kelebihan dan kekurangan masing-masing. Hasil kajian ini menunjukkan bahwa pemilihan metode yang tepat sangat bergantung pada karakteristik fungsi, kebutuhan tingkat ketelitian, serta efisiensi komputasi. Kajian ini juga menyoroti pentingnya metode numerik dalam dunia sains, teknik, dan ekonomi sebagai solusi terhadap permasalahan yang kompleks dan tidak dapat diselesaikan secara eksak. Tujuan lainnya untuk memahami dan membandingkan berbagai metode numerik dalam pencarian akar dan integrasi.**

***Kata Kunci*: Python, solusi persamaan nirlanjar, integrasi numerik, metode tertutup, metode terbuka, metode pias, metode Newton-Cotes, metode kuadratur gauss.**

I. Pendahuluan

Dalam dunia sains dan rekayasa, banyak permasalahan kompleks tidak dapat diselesaikan secara analitik akibat bentuk persamaan atau fungsi yang tidak sederhana. Ketika metode analitik tradisional gagal memberikan solusi eksak, metode numerik menjadi pilihan utama memperoleh solusi pendekatan yang akurat dan efisien. Dua topik kunci dalam metode numerik yang sering dijumpai dalam model masalah nyata adalah sistem persamaan nirlanjar dan integrasi numerik. Keduanya memainkan peran penting dalam menyelesaikan tantangan komputasi yang melibatkan ketidaklinearan dan kompleksitas fungsi.

Sistem persamaan nirlanjar merupakan kumpulan persamaan dengan variabel-variabel yang terkait secara tidak linear. Berbeda dengan sistem linear yang dapat diselesaikan menggunakan metode matriks seperti eliminasi Gauss, sistem nirlanjar memerlukan pendekatan iteratif seperti metode Newton-Raphson, metode Bagidua, metode Regula-Falsi, dan metode Secant. Di sisi lain, integrasi numerik bertujuan untuk menghitung nilai integral suatu fungsi ketika metode analitik tidak dapat diterapkan, misalnya karena fungsi tersebut terlalu kompleks, tidak memiliki antiturunan elementer, atau hanya tersedia dalam bentuk data diskrit. Metode seperti aturan trapesium, Simpson, dan kuadratur Gauss menjadi solusi untuk mengaproksimasi luas di bawah kurva dengan meminimalkan error.

Implementasi bahasa pemrograman Python, sebagai bahasa pemrograman tingkat tinggi yang bersifat *open-source*, telah menjadi bahasa pilihan untuk menyelesaikan permasalahn numerik ini. Fleksibilitas sintaksis Python memungkinkan pengguna untuk mengimplementasikan algoritma numerik dengan cepat.

II. Tinjauan Pustaka

Sistem Persamaan Nirlanjar (nonlinear) adalah kumpulan dua atau lebih persamaan yang tidak linear, yang berarti pangkat variabelnya lebih dari satu atau mengandung suku hasil kali dari dua atau lebih variabel. Dengan kata lain, sistem ini terdiri dari persamaan-persamaan yang tidak dapat digambarkan sebagai garis lurus dalam grafik.

Dalam metode numerik, pencarian akar $f(x) = 0$ dilakukan secara lelaran (iteratif). Sampai saat ini sudah banyak ditemukan metode pencarian akar. Secara umum, semua metode pencarian akar tersebut dapat dikelompokkan menjadi dua golongan besar :

A. *Metode Tertutup (Bracketing Methods)*

Metode tertutup adalah teknik yang memerlukan interval awal yang menjamin keberadaan akar (*bracketing interval*). Metode ini bersifat konvergen karena memastikan solusi berada dalam interval yang ditentukan.

Metode Bagi-Dua (Bisection Method) Metode ini membagi interval menjadi dua bagian secara iteratif dan memilih subinterval yang mengandung akar berdasarkan perubahan tanda fungsi. Meskipun konvergen secara linier dan lambat, metode ini sangat stabil dan mudah diimplementasikan. Kelemahan utamanya adalah tidak dapat diterapkan pada sistem multidimensi.

Metode Posisi Palsu (Regula Falsi) Metode ini merupakan pengembangan dari metode bagi-dua dengan menggunakan interpolasi linear untuk memperkirakan akar. Regula Falsi lebih cepat konvergen dibandingkan metode bagi-dua, tetapi masih terbatas pada kasus satu dimensi.

B. *Metode Terbuka (Open Methods)*

Metode terbuka tidak memerlukan interval awal yang mengurung akar, tetapi menggunakan tebakan awal tunggal atau ganda. Metode ini memiliki laju konvergensi lebih cepat namun tidak selalu konvergen jika tebakan awal tidak cukup dekat dengan solusi.

Lelaran titip tetap merupakan metode numerik yang efektif jika fungsi $g(x)$ dipilih dengan memenuhi syarat kontraksi. Meski konvergensinya linier, fleksibilitasnya menjadikannya alat penting dalam analisis numerik, terutama ketika turunan fungsi sulit diperoleh. Metode ini kadang-kadang dinamakan juga metode lelaran sederhana, metode langsung, atau metode sulih beruntun.

Metode Newton-Raphson menggunakan turunan fungsi untuk membentuk aproksimasi linear sistem dan iterasi menuju solusi. Untuk sistem persamaan nirlanjar multidimensi, metode ini melibatkan matriks Jacobian dan inversinya. Konvergensi kuadratiknya membuatnya sangat efisien, tetapi perhitungan Jacobian yang mahal dan sensitivitas terhadap tebakan awal menjadi kelemahan utama.

Metode Secant adalah modifikasi dari Newton-Raphson yang mengganti turunan dengan beda hingga (finite difference). Metode ini cocok untuk kasus satu dimensi ketika turunan analitik sulit diperoleh. Namun, laju konvergensinya superlinier, lebih lambat daripada Newton-Raphson.

Integrasi numerik, juga dikenal sebagai kuadratur numerik, adalah proses memperkirakan nilai integral tertentu dari suatu fungsi pada interval tertentu menggunakan metode numerik. Metode ini digunakan ketika integrasi analitis (menggunakan rumus kalkulus) tidak mungkin atau sangat sulit untuk dilakukan.

Terdapat tiga pendekatan dalam menurunkan rumus integrasi numerik. Pendekatan pertama adalah berdasarkan tafsiran geometri integral Tentu. Daerah integrasi dibagi atas sejumlah pias (*strip*) yang berbentuk segiempat. Luas daerah integrasi dihampiri dengan luas seluruh pias. Kaidah integrasi numerik yang dapat diturunkan dengan metode pias adalah :

1. Kaidah segiempat (*rectangle rule*)
2. Kaidah trapesium (*trapezoidal rule*)
3. Kaidah titik tengah (*midpoint rule*)

Pendekatan kedua adalah berdasarkan polinom interpolasi. Di sini fungsi *integrand* $f(x)$ dihampiri dengan polinom interpolasi $Pn(x)$. Selanjutnya, integrasi dilakukan terhadap $Pn(x)$ karena polinom lebih mudah diintegralkan daripada mengintegralkan $f(x)$. Rumus integrasi numerik yang diturunkan dengan pendekatan ini digolongkan ke dalam metode Newton-Cotes, yaitu metode yang umum untuk menurunkan rumus integarsi numerik. Dari beberapa kaidah integrasi numerik yang diturunkan dari metode Newton-Cotes, tiga di antaranya yang terkenal adalah :

1. Kaidah trapesium (*Trapezoidal rule*)
2. Kaidah Simpson 1/3 (*Simpson's 1/3 rule*)
3. Kaidah Simpson 3/8 (*Simpson's 3/8 rule*)

Pendekatan ketiga sama sekali tidak menggunakan titik -titik diskrit sebagaimana pada kedua pendekatan di atas. Nilai integral diperoleh dengan mengevaluasi nilai fungsi pada sejumlah titik tertentu di dalam selang [-1, 1], mengalikannya dengan suatu konstanta, kemudian menjumlahkan keseluruhan perhitungan. Pendekatan ketiga ini dinamakan Kuadratur Gauss.

III. METODE PENELITIAN

Penelitian ini bersifat studi komputasional yang bertujuan untuk mengimplementasikan metode numerik dalam menyelesaikan system persamaan nirlanjar dan integrasi numerik menggunakan bahasa pemgrograman Python. Pendekatan yang digunakan meliputi :

A. Analisis algoritma

Memahami prinsip matematis metode numerik seperti Newton-Raphson, metode Bagi-Dua, metode Regula-Falsi, metode lelaran titik tetap untuk solusi persamaan nirlanjar dan metode pias dengan kaidah segiempat, kaidah trapesium, kaidah titik tengah, metode Newton-Cotes dengan kaidah trapesium, kaidah Simpson 1/3, kaidah Simpson 3/8 untuk integrasi numerik

B. Implementasi kode

Mengembangkan skrip Python untuk memodelkan solusi numerik. Implementasi kode merujuk pada proses menerjemahkan algoritma atau prosedur matematis ke dalam serangkaian instruksi terstruktur yang dapat dieksekusi oleh komputer untuk menyelesaikan suatu masalah. Seperti melakukan estimasi awal atau inisialisasi, pengulangan iterasi terkontrol, kriteria penghentian (toleransi error dan batas iterasi maksimum), konvergensi, dan output solusi.

C. Validasi Hasil

Membandingkan solusi numerik dan memastikan bahwa solusi numerik yang diperoleh kaurat, andal, dan konsisten dengan ekspektasi matematis nya. Memverifikasi bahwa metode yang diimplementasikan bekerja sesuai dengan teori dan bebas dari kesalahan pemrograman atau asumsi yang keliru. Solusi tidak boleh berubah drastis akibat perubahan kecil pada input, dengan validasi yang ketat kesalahan implementasi atau kelemahan algoritm adapat diidentifikasi dan diperbaiki, sehingga hasil yang diperoleh memenuhi standar akurasi yang diharapkan.

D. Evaluasi

Evaluasi bertujuan untuk menentukan sejauh mana hasil yang diperoleh (seperti solusi persamaan nirlanjar atau integrasi numerik) memenuhi tujuan yang ditetapkan, baik dari segi akurasi, efisiensi, stabiltas, maupun relevansi dengan masalah yang dihadapi. Perbedaan evaluasi dengan validasi adalah, evaluasi fokus pada kualitas hasil (akurasi dan efisiensi) sedangkan validasi fokus pada kebenaran hasil (apakah solusi memenuhi persamaan atau model asli).

E. Pelaporan dan Kesimpulan

Pelaporan dengan mendokumentasi metode, hasil, dan analisis error serta memberikan kesimpulan metode mana yang paling efektif untuk kasus tertentu, rekomendasi penggunaan algoritma berdasarkan akurasi dan efisiensi, serta identifikasikan Batasan dan tantangannya.

IV. HASIL DAN PEMBAHASAN

Program ini mengimplementasikan lima metode numerik untuk menyelesaikan persamaan nirlanjar $f(x) = x^3 - 4x + 1$ pada selang [1,0]. Berikut hasil pengujian dan analisisnya.

Sebelum mengeksekusi contoh fungsi di atas kita diperintahkan untuk memilih menu metode penyelesaian yang ingin digunakan terlebih dahulu.

```
Pilih Metode Solusi Persamaan Nirlanjar yang Ingin Digunakan Ya!:
1. Metode Bagi Dua
2. Metode Regula-Falsi
3. Metode Newton-Raphson
4. Metode Secant
5. Metode Leltaran Titik Tetap
6. Keluar
Masukkan pilihan:
```

Gambar 1. Tampilan menu awal program solusi persamaan nirlanjar dengan 5 metode dan 1 pilihan menu keluar dari program.

```
Masukkan pilihan: 1

Metode Bagi Dua
Masukkan batas bawah (a): 1
Masukkan batas atas (b): 0
Masukkan toleransi error: 0.0001
Masukkan jumlah iterasi maksimum: 20
Iter a          b          c          f(c)       Error
1    1.000000   0.000000   0.500000   -0.875000  0.500000
2    0.500000   0.000000   0.250000   0.015625   0.250000
3    0.500000   0.250000   0.375000   -0.447266  0.125000
4    0.375000   0.250000   0.312500   -0.219482  0.062500
5    0.312500   0.250000   0.281250   -0.102753  0.031250
6    0.281250   0.250000   0.265625   -0.043758  0.015625
7    0.265625   0.250000   0.257812   -0.014114  0.007812
8    0.257812   0.250000   0.253906   0.000744   0.003906
9    0.257812   0.253906   0.255859   -0.006688  0.001953
10   0.255859   0.253906   0.254883   -0.002973  0.000977
11   0.254883   0.253906   0.254395   -0.001115  0.000488
12   0.254395   0.253906   0.254150   -0.000185  0.000244
13   0.254150   0.253906   0.254028   0.000279   0.000122
14   0.254150   0.254028   0.254089   0.000047   0.000061

Akar ditemukan pada x = 0.254089 setelah 14 iterasi.
```

Gambar 2. Tampilan input dan output pilihan pertama metode Bagi Dua

Berdasarkan output pada gambar, program melakukan pencarian akar menggunakan metode Bagi Dua. Metode Bagi Dua digunakan untuk mencari akar persamaan f(x) = 0 dengan cara membagi dua interval [a, b] yang didalamnya terdapat perubahan tanda nilai fungsi (f (a) · f (b) < 0). Pada setiap iterasi :

- Titik tengah c dihitung
- f(c) dievaluasi. Jika f(c) sangat dekat dengan nol (atau error sudah di bawah toleransi), maka c dianggap sebagai akar.
- Interval baru diambil berdasarkan perubahan tanda fungsi
- Error dihitung sebagai selisih nilai c saat ini dengan c sebelumnya.

```
Masukkan pilihan: 2

Metode Regula-Falsi
Masukkan batas bawah (a): 1
Masukkan batas atas (b): 0
Masukkan toleransi error: 0.0001
Masukkan jumlah iterasi maksimum: 20
Iter a          b          c          f(c)        Error
1    1.000000   0.000000   0.333333   -0.296296   0.666667
2    0.333333   0.000000   0.257143   -0.011569   0.076190
3    0.257143   0.000000   0.254202   -0.000382   0.002941
4    0.254202   0.000000   0.254105   -0.000013   0.000097

Akar ditemukan pada x = 0.254105 setelah 4 iterasi.
```

Gambar 3. Tampilan input dan output pilihan kedua metode Regula-Falsi.

Metode Regula-Falsi digunakan untuk mencari akar persamaan f(x) = 0 dengan cara memperbaiki pendekatan akar menggunakan garis lurus yang menghubungkan dua titik pada fungsi (bukan hanya titik tengah seperti metode Bagi Dua). Pada setiap iterasi :

- Nilai c dihitung menggunakan rumus Regula-Falsi, yaitu c = b - f(b) * (a - b) / (f (a) - f(b)).
- f(c) dievaluasi. Jika f(c) sangat dekat dengan nol (atau error sudah di bawah toleransi), maka c dianggap sebagai akar.
- Salah satu batas (a atau b) diganti dengan c, tergantung tanda hasil f(c).
- Error dihitung sebagai selisih nilai c saat ini dengan c sebelumnya.

Proses berhenti jika error < toleransi yang ditentukan atau iterasi maksimum tercapai.

```
Masukkan pilihan: 3

Metode Newton-Raphson
Masukkan tebakan awal (x0): 1
Masukkan toleransi error: 0.0001
Masukkan jumlah iterasi maksimum: 20
Iter x             f(x)          Error
1    -1.00000000   4.00000000    2.00000000
2    3.00000000    16.00000000   4.00000000
3    2.30434783    4.01873921    0.69565217
4    1.96748948    0.74622306    0.33685835
5    1.86947047    0.05576756    0.09801901
6    1.86087068    0.00041414    0.00859979
7    1.86080586    0.00000002    0.00006483

Akar ditemukan pada x = 1.86080586 setelah 7 iterasi.
```

Gambar 4. Tampilan input dan output pilihan keempat metode Newton-Raphson.

Metode Newton-Raphson ini membutuhkan tebakan awal dan turunan fungsi.
Setiap iterasi:

- Dihitung f(x) dan error (|x_*n+1* - *x*_n|).
- Jika error < toleransi, iterasi berhenti.
- Jika tidak, x diperbarui dengan rumus di atas.
- Output menunjukkan konvergensi nilai x menuju akar, dengan error yang semakin kecil tiap iterasi.

```
Masukkan pilihan: 4

Metode Secant
Masukkan tebakan awal pertama (x0): 1
Masukkan tebakan awal kedua (x1): 0
Masukkan toleransi error: 0.0001
Masukkan jumlah iterasi maksimum: 20
Iter x0           x1           x2           f(x2)         Error
1    1.00000000   0.00000000   0.33333333   -0.29629630   0.33333333
2    0.00000000   0.33333333   0.25714286   -0.01156851   0.07619048
3    0.33333333   0.25714286   0.25404723   0.00020728    0.00309563
4    0.25714286   0.25404723   0.25410172   -0.00000013   0.00005449

Akar ditemukan pada x = 0.25410172 setelah 4 iterasi.
```

Gambar 5. Tampilan input dan output pilihan keempat metode Secant.

Metode Secant adalah metode numerik untuk mencari akar persamaan f(x) = 0 dengan menggunakan dua tebakan awal tanpa memerlukan turunan. Dua nilai awal (x_0 dan x_1) diperlukan untuk memulai proses iterasi.

Setiap iterasi :

- Hitung nilai x2 menggunakan rumus di atas.
- Nilai f(x2) dievaluasi.
- Error dihitung sebagai selisih antara x2 dengan x1 sebelumnya.
- Iterasi berakhir jika error < toleransi atau jumlah iterasi maksimum tercapai.

Output menunjukkan metode Secant sangat cepat konvergen pada kasus ini, karena hanya membutuhkan 4 iterasi.

```
Masukkan pilihan: 5

Metode Lelaran Titik Tetap
Masukkan tebakan awal (x0): 1
Masukkan toleransi error: 0.0001
Masukkan jumlah iterasi maksimum: 20
Iter x            g(x)          Error
1    0.50000000   0.28125000    0.50000000
2    0.28125000   0.25556183    0.21875000
3    0.25556183   0.25417280    0.02568817
4    0.25417280   0.25410513    0.00138902
5    0.25410513   0.25410186    0.00006767

Akar ditemukan pada x = 0.25410513 setelah 5 iterasi.
```

Gambar 6. Tampilan input dan output pilihan kelima metode lelaran titik tetap.

Metode titik tetap digunakan untuk menyelesaikan persamaan f(x) = 0 dengan mengubah persamaan menjadi bentuk x = g(x), lalu melakukan iterasi, iterasi dilakukan hingga nilai |x_{*n+1*} - *x*_n| < toleransi error.

Proses setiap iterasi:

- Hitung g(x) dari nilai x sebelumnya.
- Evaluasi error sebagai selisih absolut antara g(x) dengan x lama.
- Jika error < toleransi, iterasi dihentikan.

Output menunjukkan konvergensi yang cukup cepat, hanya memerlukan 5 iterasi untuk mencapai error di bawah toleransi.

Selanjutnya pada program integrasi numerik ini mengimplementasikan enam metode dari masing-masing metode pias dan metode Newton-Cotes.

f (x) = 2x + 1 pada interval [0, 2]. Berikut hasil dan pembahasannya.

```
Program Integral Numerik
Fungsi contoh: f(x) = 2*x + 1 pada interval [0, 2]
1. Metode Pias (Kaidah Segiempat)
2. Metode Pias (Kaidah Trapesium)
3. Metode Pias (Kaidah Titik Tengah)
4. Newton-Cotes (Kaidah Trapesium)
5. Newton-Cotes (Simpson 1/3)
6. Newton-Cotes (Simpson 3/8)
7. Keluar
Pilih metode:
```

Gambar 7. Tampilan awal program integral numerik dengan enam metode utama dan satu menu keluar.

```
Pilih metode: 1
Masukkan batas bawah (a): 2
Masukkan batas atas (b): 0
Masukkan jumlah pias (n): 4

Langkah-langkah Metode Pias (Kaidah Segiempat):
h = (b - a) / n = (0.0 - 2.0) / 4 = -0.5
f(x0) = f(2.0) = 5.0
f(x1) = f(1.5) = 3.25
f(x2) = f(1.0) = 2.0
f(x3) = f(0.5) = 1.25
Integral ≈ h * jumlah f(xi) = -0.5 * 11.5 = -5.75

Hasil integral (Pias Segiempat): -5.75
```

Gambar 8. Tampilan penyelesaian dengan metode pias kaidah segiempat.

```
7. Keluar
Pilih metode: 2
Masukkan batas bawah (a): 2
Masukkan batas atas (b): 0
Masukkan jumlah pias (n): 4

Langkah-langkah Metode Pias (Kaidah Trapesium):
h = (b - a) / n = (0.0 - 2.0) / 4 = -0.5
f(a) = f(2.0) = 5.0
f(b) = f(0.0) = 1.0
f(x1) = f(1.5) = 3.25
f(x2) = f(1.0) = 2.0
f(x3) = f(0.5) = 1.25
Jumlah total = f(a) + f(b) + 2 * jumlah f(xi) tengah
Integral ≈ (h/2) * total = (-0.5/2) * 19.0 = -4.75

Hasil integral (Pias Trapesium): -4.75
```

Gambar 9. Tampilan penyelesaian dengan metode pias trapesium.

```
Pilih metode: 3
Masukkan batas bawah (a): 2
Masukkan batas atas (b): 0
Masukkan jumlah pias (n): 4

Langkah-langkah Metode Pias (Kaidah Titik Tengah):
h = (b - a) / n = (0.0 - 2.0) / 4 = -0.5
f(x_tengah0) = f(1.75) = 4.0625
f(x_tengah1) = f(1.25) = 2.5625
f(x_tengah2) = f(0.75) = 1.5625
f(x_tengah3) = f(0.25) = 1.0625
Integral ≈ h * jumlah f(x_tengah) = -0.5 * 9.25 = -4.625

Hasil integral (Pias Titik Tengah): -4.625
```

Gambar 10. Tampilan penyelesaian dengan metode pias kaidah titik tengah.

```
Pilih metode: 4
Masukkan batas bawah (a): 2
Masukkan batas atas (b): 0
Masukkan jumlah pias (n): 4

Langkah-langkah Metode Pias (Kaidah Trapesium):
h = (b - a) / n = (0.0 - 2.0) / 4 = -0.5
f(a) = f(2.0) = 5.0
f(b) = f(0.0) = 1.0
f(x1) = f(1.5) = 3.25
f(x2) = f(1.0) = 2.0
f(x3) = f(0.5) = 1.25
Jumlah total = f(a) + f(b) + 2 * jumlah f(xi) tengah
Integral ≈ (h/2) * total = (-0.5/2) * 19.0 = -4.75

Hasil integral (Newton-Cotes Trapesium): -4.75
```

Gambar 11. Tampilan penyelesaian dengan metode Newton-Cotes kaidah trapesium.

```
Pilih metode: 5
Masukkan batas bawah (a): 2
Masukkan batas atas (b): 0
Masukkan jumlah pias genap (n): 4

Langkah-langkah Metode Simpson 1/3:
h = (b - a) / n = (0.0 - 2.0) / 4 = -0.5
f(a) = f(2.0) = 5.0
f(b) = f(0.0) = 1.0
4 * f(x1) = 4 * f(1.5) = 13.0
2 * f(x2) = 2 * f(1.0) = 4.0
4 * f(x3) = 4 * f(0.5) = 5.0
Integral ≈ (h/3) * total = (-0.5/3) * 28.0 = -4.666666666666666

Hasil integral (Simpson 1/3): -4.666666666666666
```

Gambar 12. Tampilan penyelesaian dengan metode Newton-Cotes kaidah Simpson 1/3.

```
Pilih metode: 6
Masukkan batas bawah (a): 2
Masukkan batas atas (b): 0
Masukkan jumlah pias kelipatan 3 (n): 6

Langkah-langkah Metode Simpson 3/8:
h = (b - a) / n = (0.0 - 2.0) / 6 = -0.3333333333333333
f(a) = f(2.0) = 5.0
f(b) = f(0.0) = 1.0
3 * f(x1) = 3 * f(1.6666666666666667) = 11.333333333333334
3 * f(x2) = 3 * f(1.3333333333333335) = 8.333333333333334
2 * f(x3) = 2 * f(1.0) = 4.0
3 * f(x4) = 3 * f(0.6666666666666667) = 4.333333333333334
3 * f(x5) = 3 * f(0.3333333333333335) = 3.3333333333333335
Integral = (3h/8) * total = (3*-0.3333333333333333/8) * 37.333333333333334 = -4.666666666666668

Hasil integral (Simpson 3/8): -4.666666666666668
```

Gambar 13. Tampilan penyelesaian dengan metode Newton-Cotes kaidah Simpson 3/8.

Metode Pias (Kaidah Segiempat) membagi interval

- [a, b] menjadi n bagian dengan lebar sama (h) dan menghitung jumlah nilai fungsi pada titik-titik tertentu, lalu dikalikan dengan lebar pias.
- Pengambilan batas bawah lebih besar daripada batas atas (a=2, b=0) menyebabkan h bernilai negatif, sehingga hasil integral juga menjadi negatif. Ini konsisten dengan arah integrasi pada sumbu x.
- Nilai f(xi) diambil pada titik-titik x = 2.0, 1.5, 1.0, dan 0.5. Nilai-nilai fungsi dijumlahkan dan dikalikan dengan h.
- Metode ini hanya merupakan aproksimasi, namun cukup baik untuk fungsi yang tidak terlalu berfluktuasi di interval integrasi.

Metode Trapesium adalah salah satu metode numerik untuk menghitung integral tak tentu dengan membagi daerah integral menjadi beberapa trapesium, lalu menghitung luas total trapesium tersebut.

- Pengambilan batas bawah lebih besar daripada batas atas (a = 2, b = 0) menyebabkan h bernilai negatif, sehingga hasil integral juga negatif. Ini sesuai dengan definisi integral yang mengacu pada arah integrasi pada sumbu-x.
- Nilai f(x) dihitung di titik awal dan akhir serta titik-titik di antara keduanya, dengan titik tengah dikalikan dua.
- Hasil ini biasanya lebih akurat dibandingkan metode pias segiempat, karena metode trapesium memperhitungkan perubahan nilai fungsi secara linier di antara dua titik.
- Hasil integral yang didapatkan adalah -4.75, merepresentasikan luas daerah di bawah kurva untuk interval [2,0].

Metode Pias Titik Tengah (Midpoint Rule) adalah metode numerik untuk menghitung integral dengan membagi interval [a, b] menjadi n pias, lalu menghitung nilai fungsi di titik tengah masing-masing pias, kemudian mengalikan rata-rata nilai fungsi dengan lebar pias.

- Pengambilan batas bawah lebih besar daripada batas atas (a = 2, b = 0) menyebabkan h bernilai negatif, sehingga hasil integral juga negatif. Ini sesuai dengan definisi arah integral pada sumbu-x.
- Nilai f(x) dievaluasi di titik tengah setiap pias, memberikan aproksimasi yang umumnya lebih akurat daripada metode pias segiempat biasa, karena memperhitungkan "rata-rata" tinggi fungsi di setiap pias.
- Total nilai fungsi di titik-titik tengah dijumlahkan dan dikalikan dengan h untuk memperoleh hasil integral.
- Hasil integral yang diperoleh adalah -4.625, merepresentasikan luas daerah di bawah kurva untuk interval [2, 0] sesuai arah integrasi.

Metode Trapesium/Newton-Cotes merupakan salah satu metode numerik yang digunakan untuk menghitung aproksimasi nilai integral dengan cara membagi interval menjadi beberapa pias dan menghitung luas trapesium di bawah kurva.

- Tanda negatif pada hasil integral disebabkan oleh urutan batas integrasi (a > b), sehingga h bernilai negatif.
- Metode ini memperhitungkan nilai fungsi di kedua ujung interval dan di titik-titik tengah dengan bobot dua, sehingga lebih akurat dibanding metode segiempat (Riemann).
- Hasil yang diperoleh, -4.75, menunjukkan aproksimasi luas daerah di bawah kurva untuk interval [2, 0] sesuai arah integrasi.

Metode Simpson 1/3 merupakan metode integrasi numerik yang lebih akurat dibandingkan Trapesium atau Segiempat, karena menggunakan polinomial derajat dua untuk aproksimasi fungsi di tiap subinterval.

- Bobot untuk titik-titik tertentu
- Penggunaan batas bawah lebih besar dari batas atas (a=2, b=0) menyebabkan h negatif, sehingga hasil integral juga negatif. Ini konsisten dengan arah integrasi pada sumbu x.
- Hasil yang diperoleh -4.666… biasanya lebih akurat daripada metode trapesium atau pias

segiempat untuk jumlah pias genap dan fungsi yang cukup halus.

Metode Simpson 3/8 adalah metode integrasi numerik yang menggunakan polinomial derajat tiga untuk aproksimasi fungsi di setiap subinterval, dan membutuhkan jumlah pias kelipatan tiga.

- Bobot pada titik-titik tertentu
- Tanda negatif pada hasil integral berasal dari urutan batas (a = 2, b = 0), sehingga h negatif dan hasil integral negatif.
- Hasil integral dari metode Simpson 3/8 biasanya sangat akurat untuk fungsi polinomial derajat tiga atau kurang, dan pada soal ini hasilnya -4.666666666666668, sangat mirip dengan hasil Simpson 1/3 pada soal sebelumnya.
- Metode Simpson 3/8 lebih fleksibel untuk jumlah pias yang merupakan kelipatan 3 (bukan hanya genap seperti Simpson 1/3).

V. KESIMPULAN

Dari seluruh hasil dan pembahasan beberapa metode numerik untuk mencari akar persamaan dan menghitung integral dalam program, dapat diambil beberapa poin penting sebagai berikut:

1. Metode Pencarian Akar

Metode Newton-Raphson, metode Bagi Dua, metode Regula-Falsi, metode Secant, dan Iterasi Lelaran Titik Tetap merupakan metode numerik untuk mencari akar persamaan non-linear.

Newton-Raphson sangat cepat konvergen jika tebakan awal dekat akar dan fungsi/turunan tidak terlalu sulit.

Secant tidak membutuhkan turunan fungsi, dan biasanya lebih cepat dari metode iterasi titik tetap jika tebakan awal baik.

Iterasi Titik Tetap sederhana, namun konvergensinya sangat bergantung pada bentuk fungsi g(x).

2. Metode Integrasi Numerik

Metode Pias (Segiempat), Trapesium, Titik Tengah, Newton-Cotes Trapesium, Simpson 1/3, dan Simpson 3/8 digunakan untuk menghitung aproksimasi nilai integral dari fungsi tertentu pada interval tertentu.

Metode Segiempat adalah yang paling sederhana, hanya menjumlahkan luasan persegi panjang. Akurasinya rendah untuk fungsi yang tidak konstan.

Metode Trapesium memperbaiki akurasi dengan memperhitungkan perubahan fungsi secara linier.

Metode Titik Tengah biasanya lebih akurat daripada segiempat biasa, karena mengambil nilai fungsi di tengah pias.

Simpson 1/3 dan 3/8 menggunakan polinomial derajat dua dan tiga untuk aproksimasi, sehingga lebih akurat untuk fungsi yang halus dan jumlah pias genap/kelipatan tiga.

Hasil integral negatif diperoleh jika batas bawah lebih besar dari batas atas, sesuai dengan sifat integral tertentu.

3. Perbandingan dan Efektivitas

Untuk pencarian akar, Newton-Raphson dan Secant biasanya lebih cepat konvergen dibandingkan metode lain, namun Newton-Raphson membutuhkan turunan fungsi.

Untuk integrasi, Simpson 1/3 dan 3/8 secara umum memberikan hasil yang paling akurat untuk jumlah pias yang sesuai, diikuti metode trapesium dan titik tengah.

Pilihan metode tergantung karakteristik fungsi, interval, dan kebutuhan ketelitian hasil.

DAFTAR PUSTAKA

[1] Munir, R. (2015). *Metode Numerik untuk Teknik*. Bandung: Informatika

[2] Virtanen, P., et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17(3), 261–272.

[3] Chapra, S. C., & Canale, R. P. (2015). *Numerical Methods for Engineers* (7th ed.). McGraw-Hill Education.

[4] Chapra, S. C., & Canale, R. P. (2015). *Numerical Methods for Engineers* (7th ed.). McGraw-Hill Education.

[5] Atkinson, K. E. (1989). *An Introduction to Numerical Analysis* (2nd ed.). John Wiley & Sons.

[6] Sulistyono, B. (2018). *Analisis Galat pada Metode Integrasi Numerik Trapesium dan Titik Tengah*. Jurnal Matematika dan Aplikasinya, 12(1), 45-56.

[7] Wahyudi, A. (2020). *Penerapan Metode Pias dalam Perhitungan Volume Bendungan*. Jurnal Teknik Sipil Indonesia, 8(2), 112-120.

[8] Arifin, R. (2017). "Perbandingan Metode Regula Falsi dan Metode Newton-Raphson dalam Mencari Akar Persamaan Non Linier". Jurnal Gaussian, 6(2), 161-170.

[9] Pratama, A. I., & Supriyanto, E. (2018). "Perbandingan Metode Trapesium dan Simpson 1/3 pada Perhitungan Integral Numerik". Jurnal Ilmiah Pendidikan Matematika, 7(2), 124–132.

[10] Kurniawati, N., & Herawati, S. (2019). "Perbandingan Metode Numerik pada Penghitungan Integral Tertentu". Jurnal Matematika Integratif, 15(2), 125–132.

[11] Sari, S. W., & Anshory, A. (2021). "Perbandingan Metode Bagi Dua dan Metode Regula Falsi dalam Menentukan Akar Persamaan Non-Linear". Jurnal Cendekia: Jurnal Pendidikan Matematika, 5(2), 1172-1180.