



STUDI KOMPARATIF TURUNAN NUMERIK DENGAN METODE SELISIH MENGGUNAKAN PYTHON DAN MAPLE: AKURASI DAN KEMUDAHAN IMPLEMENTASI

Hilmina Rofa' Baedowi^{1,*}, Nur Hidayati²), Ari Wibowo³)

*^{1,2,3}Program Studi Tadris Matematika, Fakultas Ilmu Tarbiyah, UIN Raden Mas
Said Surakarta*

**email: hilminarofa@gmail.com*

Abstrak: Implementasi konsep turunan tidak hanya digunakan dalam penyelesaian turunan fungsi eksplisit secara analitik saja, pada praktiknya juga diperlukan pendekatan numerik untuk menentukan hampiran fungsi yang kompleks maupun hampiran pada data diskrit. Penelitian ini mengkaji mengenai akurasi dan kemudahan implementasi aplikasi komputasi Python dan Maple dalam menyelesaikan turunan numerik menggunakan metode selisih, baik metode selisih maju, selisih mundur, maupun selisih pusat. Jenis penelitian ini adalah penelitian eksperimen yang terdiri atas tahapan meliputi perumusan masalah, pengumpulan data, pengolahan data, analisis data, dan penarikan kesimpulan. Penilaian akurasi hasil turunan numerik pada aplikasi Python dan Maple dilakukan melalui nilai galat mutlak dan penilaian kemudahan implementasi dilihat dari proses pemrograman, tampilan *output*, dan kecepatan eksekusi. Hasil analisis menunjukkan bahwa Python dan Maple sama-sama akurat dalam menyelesaikan turunan fungsi eksplisit yang rumit maupun data diskrit menggunakan turunan numerik metode selisih maju, selisih mundur, maupun selisih pusat. Metode selisih pusat memiliki galat terkecil, artinya metode selisih pusat menghasilkan nilai yang paling mendekati turunan eksaknya. Python unggul dalam kecepatan eksekusi dan ketelitian hasil, sedangkan Maple lebih unggul dalam kemudahan penulisan fungsi.

Kata Kunci: Maple, Metode Selisih, Python, Turunan Numerik

Abstract: *The concept of derivatives is applied not only to analytically solving explicit functions but also to numerically approximating complex functions and discrete data. This study adopts an experimental approach, involving several stages such as: problem formulation, data collection, data processing, data analysis, and conclusion drawing. The accuracy of the numerical derivative results in Python and Maple was evaluated using absolute error values, while the ease of implementation was assessed based on the programming process, output display, and execution speed. The results of the analysis show that both Python and Maple are accurate in computing numerical derivatives for both complex explicit functions and discrete data using the forward, backward, and central difference methods. The central difference method yielding the smallest error, indicating it provides the closest approximation to the exact derivative. Python excels in execution speed and computational precision, while Maple offers an advantage in ease of function notation.*

Keywords: *Maple, Finite Difference Method, Python, Numerical Differentiation*



PENDAHULUAN

Matematika adalah ilmu yang menjadi fondasi pokok perkembangan teknologi dan ilmu pengetahuan masa kini (Judijanto, dkk., 2024: 452). Cabang ilmu matematika yang memiliki peran esensial dalam menganalisis perubahan dan memahami dinamika sistem adalah kalkulus (Neldiana & Harisman, 2025: 274). Salah satu konsep pokok dalam kalkulus adalah turunan. Turunan adalah terma dalam matematika yang menyatakan perubahan suatu besaran sebagai akibat dari adanya perubahan dari besaran yang lain (Rifa'i, 2020: 36). Turunan dalam kalkulus dasar memiliki kontribusi yang luas dalam pemecahan masalah nyata, misalnya untuk menentukan titik stasioner, menentukan nilai maksimum dan minimum dari suatu kurva, mengukur kecepatan dan percepatan suatu partikel sepanjang garis lurus, menghitung optimasi fungsi biaya, menentukan laju perubahan, dan sebagainya (Sroyer, dkk., 2024: 113–118).

Implementasi konsep turunan ini tidak hanya digunakan dalam penyelesaian turunan fungsi eksplisit sederhana yang dapat diturunkan secara analitik saja. Akan tetapi, pada praktiknya ditemui fungsi yang terlalu rumit untuk diturunkan secara simbolik. Selain itu, sering dijumpai pula data dalam bentuk tabel atau nilai-nilai diskrit yang turunannya perlu dicari untuk keperluan analisis ataupun pemodelan. Oleh karena itu, diperlukan pendekatan turunan numerik untuk menentukan aproksimasi nilai turunan dari fungsi eksplisit yang rumit atau data dalam bentuk diskrit (Samaray, 2023: 45). Perhitungan numerik dapat dilakukan dengan iterasi yang bertujuan untuk menghasilkan ketelitian yang akurat dengan eror sekecil mungkin (Maure & Mungkasi, 2021; Tukan et al., 2024).

Kajian mengenai turunan numerik sangat penting mengingat keterbatasan pengaplikasian metode analitik untuk menyelesaikan persoalan matematis yang kompleks (Zakaria & Muharramah, 2023: 2). Turunan numerik yang kerap digunakan dalam menghitung hampiran nilai turunan suatu fungsi adalah metode selisih (Samaray, 2023: 45). Metode selisih dalam turunan numerik terdiri atas hampiran selisih-maju (*forward difference approximation*), hampiran selisih-mundur (*backward difference approximation*), dan hampiran selisih-pusat (*central difference approximation*) (Munir, 2015: 332). Masing-masing metode selisih menghasilkan tingkat galat atau *error* yang berbeda-beda. Dengan demikian, implementasi dari metode-metode turunan numerik tersebut perlu disesuaikan dengan konteks penggunaan. Metode selisih dalam turunan numerik dapat dihitung menggunakan aplikasi komputasi, seperti Python dan Maple.

Perangkat lunak komputasi yang populer digunakan baik di kalangan akademik maupun profesional saat ini adalah Python dan Maple, karena sintaksnya cenderung mudah dimengerti, hasil komputasinya akurat, dan fleksibel untuk digunakan dalam berbagai implementasi (Nazar, 2024: 51; Ramdhani, 2021: 79). Python adalah bahasa pemrograman interpreter berbasis teks yang secara komprehensif memadukan sintaks kode disertai dengan pustaka standar (Wibowo, dkk., 2020: 1). Python sebagai bahasa pemrograman interereter artinya kode program pada Python akan secara langsung diproses dari satu baris ke baris selanjutnya tanpa membutuhkan proses *compile* (Rahmatudin & Sumliyah, 2024: 16). Python



memiliki pustaka, seperti *numpy*, *scipy*, *matplotlib*, dan *sympy* yang digunakan untuk memudahkan perhitungan secara numerik maupun simbolik dan memadukan komputasi matematika dengan visualisasi (Enterprise, 2017: 2; Maulida Surbakti, dkk., 2024: 99s). Selain itu, keunggulan Python adalah dapat digunakan pada berbagai *platform* sistem operasi (Enterprise, 2017: 1). Namun, karena Python tidak menyediakan *default* ekspresi matematika, implementasi turunan numerik metode selisih pada Python perlu menginput rumus perhitungannya secara manual.

Maple adalah program komputer yang mampu melakukan komputasi secara simbolik dan numerik yang dapat merekayasa model, metode, dan perhitungan algoritma (Nur Aini, dkk., 2017: 1). Aplikasi Maple unggul dalam menyelesaikan komputasi simbolik maupun numerik yang sangat besar, sehingga dapat digunakan untuk menyelesaikan berbagai fungsi matematika disertai dengan visualisasinya (Ramdhani, 2021: 74). Sama dengan Python, Maple juga dapat digunakan pada berbagai *platform* sistem operasi. Maple menyediakan perintah spesifik untuk menuliskan ekspresi matematika (Ramdhani, 2021: 74). Namun, Maple merupakan aplikasi berbayar, sehingga akses bagi pengguna terbatas tanpa adanya lisensi resmi.

Penelitian yang mengkaji penggunaan bahasa pemrograman Python dan aplikasi Maple, seperti penelitian oleh Arvi1, dkk. (2024) yang membahas implikasi Python untuk mengerjakan induksi matematika, penelitian oleh Rahmadi, dkk. (2024) yang melakukan kompresi gambar menggunakan SVD dengan Python, serta penelitian yang dilakukan Pongsapan dan Pawaragan (2021) yang membahas penyelesaian kasus mekanika klasik menggunakan Maple. Penelitian lain mengenai turunan numerik metode selisih, yaitu penelitian yang dilakukan oleh Tukan (2024) yang mengkaji keakuratan metode selisih dalam menyelesaikan turunan persamaan non-linier. Sejauh ini sebagian besar penelitian hanya membahas penerapan turunan numerik dengan *platform* Python dan Maple secara terpisah tanpa melakukan perbandingan antara keduanya. Penelitian sebelumnya lebih banyak menitikberatkan pada perbandingan tingkat akurasi antar metode numerik, seperti metode selisih maju, metode selisih mundur, dan metode selisih pusat dalam perhitungan turunan numerik. Selain itu, seiring dengan perkembangan terkini, kebutuhan akan perangkat komputasi yang efisien dalam menyelesaikan model matematis meningkat. Akan tetapi, belum ada penelitian mengenai evaluasi terintegrasi dari aspek akurasi dan kemudahan dari aplikasi komputasi, seperti Python dan Maple. Berdasarkan uraian tersebut, peneliti tertarik untuk mengkaji akurasi dan kemudahan implementasi aplikasi komputasi Python dan Maple dalam menghitung turunan numerik dengan metode selisih.

METODE PENELITIAN

Penelitian ini menggunakan jenis penelitian eksperimen yang dilakukan melalui langkah-langkah penelitian berikut (Samaray, 2023: 46):

1. Perumusan Masalah



Penelitian ini diawali dengan mengidentifikasi masalah yang kemudian menjadi dasar dalam melakukan analisis.

2. Pengumpulan Data

Mencari kasus turunan numerik yang berbentuk fungsi eksplisit yang rumit dan data dalam bentuk tabel atau nilai-nilai diskrit. Kedua kasus turunan numerik tersebut akan diselesaikan dengan tiga metode selisih, yaitu selisih maju, selisih mundur, dan selisih pusat. Pada penelitian ini kedua kasus turunan numerik yang digunakan merupakan kasus fiktif yang bersumber dari literatur oleh Munir (2015). Berikut disajikan kasus turunan numerik dari sebuah fungsi eksplisit berbentuk komposit transendental yang gabungan fungsi aljabar, eksponensial, dan logaritmik:

$$f(x) = x e^{(2x+2)} \ln(4x^2)$$

(Munir, 2015: 351)

Untuk nilai $x = 3$ dan nilai $h = 0.01$.

Adapun kasus turunan numerik dalam bentuk data nilai-nilai diskrit disajikan pada tabel sebagai berikut:

Tabel 1. Nilai-nilai Diskrit

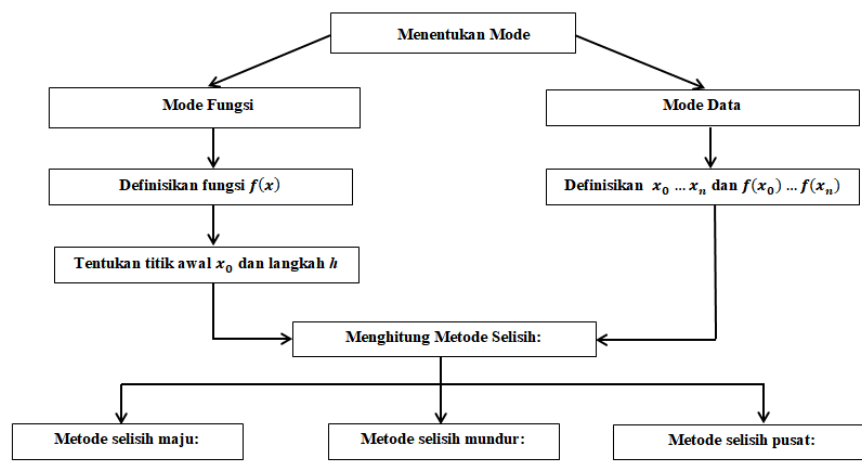
x	f(x)
1.3	3.669
1.5	4.482
1.7	5.474
1.9	6.686
2.1	8.166
2.3	9.974
2.5	12.182

(Munir, 2015: 363)

Tabel 1. menyajikan data nilai-nilai diskrit fungsi $f(x)$ pada 7 titik x dengan selisih tiap nilai x (h) adalah 0.2.

3. Pengolahan Data

- a. Menyelesaikan kasus turunan menggunakan pendekatan turunan numerik metode selisih dengan bantuan aplikasi komputasi Python dan Maple. Berikut disajikan gambar algoritma yang digunakan untuk kedua aplikasi komputasi tersebut:



Gambar 1. Gambar Algoritma Pengolahan Data

Pada Gambar 1. ditunjukkan alur proses penentuan turunan numerik menggunakan metode selisih, dimulai dari pemilihan mode (fungsi atau data), hingga perhitungan turunan menggunakan metode selisih maju, mundur, atau pusat. Adapun metode selisih yang digunakan dalam penelitian ini bersumber pada Munir (2015: 332), dengan rumus-rumus sebagai berikut:

Hampiran selisih-maju:

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} = \frac{f_1 - f_0}{h}$$

Hampiran selisih-mundur:

$$f'(x_0) = \frac{f(x_0) - f(x_0 - h)}{h} = \frac{f_0 - f_{-1}}{h}$$

Hampiran selisih-pusat:

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} = \frac{f_1 - f_{-1}}{2h}$$

- b. Untuk kasus turunan numerik dalam bentuk fungsi, setelah diperoleh nilai numerik masing-masing metode selisihnya, selanjutnya dihitung galat mutlak dari metode selisih maju, selisih mundur, dan selisih pusat.
 - c. Baik kasus turunan numerik dalam bentuk fungsi eksplisit yang rumit maupun data nilai-nilai diskrit, keduanya ditampilkan grafik dari masing-masing metode selisih maju, selisih mundur, dan selisih pusat. Khusus untuk kasus turunan numerik dalam bentuk fungsi eksplisit yang rumit, ditampilkan juga grafik turunan eksaknya.
4. Analisis Data

Penilaian akurasi hasil turunan numerik pada kedua aplikasi komputasi dilakukan dengan membandingkan nilai turunan analitik dan galat mutlaknya. Galat mutlak menurut Imron dkk., (2022:11) merupakan selisih antara nilai turunan analitik dan nilai pendekatan hasil metode numerik yang dirumuskan sebagai berikut:

$$E = | p - p^* |$$



dengan E merupakan galat mutlak, kemudian p merupakan nilai turunan numerik dan p^* merupakan nilai turunan analitik. Kemudian, untuk menilai kemudahan implementasi pada kedua aplikasi komputasi dilakukan secara deskriptif dengan mempertimbangkan beberapa aspek praktis berdasarkan pengalaman peneliti. Aspek-aspek yang dimaksud yaitu mengenai fitur bawaan untuk memasukkan fungsi atau data dan kecepatan eksekusi dari kedua aplikasi komputasi.

5. Penarikan Kesimpulan

Melakukan penarikan kesimpulan berdasarkan hasil analisis data.

HASIL DAN PEMBAHASAN

Hasil Penelitian

Pada bagian ini disajikan hasil komputasi turunan numerik untuk fungsi eksplisit yang rumit dan data dalam bentuk nilai-nilai diskrit dengan menggunakan metode selisih maju, metode selisih mundur, dan metode selisih pusat yang diimplementasikan melalui Python dan Maple. Hasil komputasi yang meliputi nilai-nilai aproksimasi turunan ini digunakan sebagai dasar untuk menganalisis perbandingan aplikasi komputasi Python dan Maple pada bagian pembahasan.

Aplikasi Turunan Numerik Dengan Metode Selisih Menggunakan Python

Implementasi turunan numerik menggunakan Python memerlukan penyusunan skrip program, mulai dari pendefinisian fungsi, penentuan titik diferensiasi, penentuan nilai selisih antar titik (h), hingga penerapan rumus metode selisih. Proses ini akan melibatkan penggunaan berbagai pustaka Python untuk mempermudah pengolahan data. Dengan demikian, dalam implementasi ini pengguna dituntut untuk memahami bahasa pemrograman Python dan cara pengaturan parameter numerik agar hasil komputasi sesuai dengan kebutuhan analisis.

Hasil komputasi untuk kasus turunan numerik dalam bentuk fungsi eksplisit yang rumit dengan menggunakan Python terlihat pada gambar berikut:

```
=====|| TURUNAN NUMERIK ||=====
Pilih mode (1: Fungsi Eksplisit Rumit, 2: Data Tabel): 1
Masukkan fungsi f(x) (misal: sin(x), x**2 + 3*x): x * exp(2*x + 2) * ln(4*x**2)
Masukkan nilai x (misal: 2 atau pi/4): 3
Masukkan nilai h: 0.01

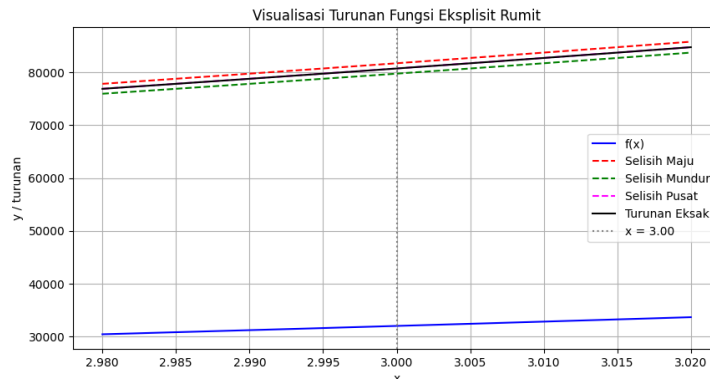
=====|| TURUNAN NUMERIK FUNGSI EKSPRESI RUMIT ||=====
x = 3.00000, h = 0.01
Turunan Eksak = 80738.1517832257
Selisih Maju = 81729.7471966416 | Galat = 9.92e+02
Selisih Mundur = 79762.1357149612 | Galat = 9.76e+02
Selisih Pusat = 80745.9414558014 | Galat = 7.79e+00
Simpan grafik? (yes/no): yes
```

Gambar 2. Penyelesaian Fungsi Eksplisit Rumit dengan Python

Berdasarkan hasil perhitungan pada Gambar 6, diperoleh nilai eksak sebesar 80738.1517832257, nilai hampiran selisih maju sebesar 81729.7471966416 dengan galat



mutlak sebesar $9.92e+02$, nilai hampiran selisih mundur sebesar 79762.1357149612 dengan galat mutlak sebesar $9.76e+02$, serta nilai hampiran selisih pusat sebesar 80745.9414558014 dengan galat mutlak sebesar $7.79e+00$. Hal ini juga terlihat dari grafik yang dipaparkan pada gambar 7 berikut:



Gambar 3. Grafik Turunan Numerik Fungsi Eksplisit Rumit dengan Python

Gambar 7. menunjukkan masing-masing nilai hampiran metode selisih maju yang berupa garis strip berwarna merah, metode selisih mundur yang berupa garis strip berwarna hijau, dan metode selisih pusat yang berupa garis strip berwarna ungu, serta turunan eksak dari fungsi eksplisit rumit yang berupa garis solid berwarna hitam. Terlihat bahwa garis strip yang berwarna ungu berada paling dekat dengan garis solid yang berwarna hitam. Artinya, nilai hampiran selisih pusat hampir sama dengan nilai turunan eksak. Ini menunjukkan bahwa nilai pendekatan metode selisih pusat memberikan estimasi turunan yang lebih akurat pada titik yang dianalisis.

Hasil komputasi untuk kasus turunan numerik dalam bentuk data diskrit dengan menggunakan Python akan disajikan pada gambar berikut:

```

=====|| TURUNAN NUMERIK ||=====
Pilih mode (1: Fungsi Eksplisit Rumit, 2: Data Tabel): 2
Masukkan data x (penulisan data dengan koma): 1.3, 1.5, 1.7, 1.9, 2.1, 2.3, 2.5
Masukkan data y (penulisan data dengan koma): 3.669, 4.482, 5.474, 6.686, 8.166, 9.974, 12.182

=====|| TURUNAN NUMERIK DATA ||=====
  x | f(x) | Forward | Backward | Central
1.30 | 3.67 | 4.07 | nan | nan
1.50 | 4.48 | 4.96 | 4.07 | 4.51
1.70 | 5.47 | 6.06 | 4.96 | 5.51
1.90 | 6.69 | 7.40 | 6.06 | 6.73
2.10 | 8.17 | 9.04 | 7.40 | 8.22
2.30 | 9.97 | 11.04 | 9.04 | 10.04
2.50 | 12.18 | nan | 11.04 | nan
Simpan hasil ke CSV? (yes/no): yes
Simpan grafik? (yes/no): yes
  
```

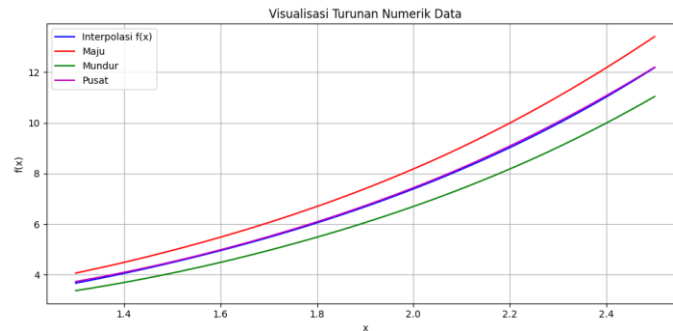
Gambar 4. Penyelesaian Data Diskrit dengan Python

Gambar 8. menunjukkan bahwa perhitungan menggunakan kode Python yang dikembangkan pada penelitian ini berlaku untuk mencari nilai turunan numerik dengan metode selisih di



semua titik. Hasil turunan numerik yang berupa “nan” mengindikasikan bahwa metode selisih terkait tidak dapat dihitung pada titik tersebut, karena data yang dibutuhkan (sebelum atau sesudah) tidak tersedia.

Adapun visualisasi grafik hasil turunan numeriknya disajikan pada gambar di bawah ini:



Gambar 5. Grafik Turunan Numerik Data Diskrit dengan Python

Grafik di atas menunjukkan masing-masing nilai hampiran metode selisih maju berupa garis solid berwarna merah, nilai hampiran metode selisih mundur berupa garis solid berwarna hijau, dan nilai hampiran metode selisih pusat berupa garis solid berwarna ungu, serta nilai interpolasi $f(x)$ berupa garis solid berwarna biru. Tampak bahwa garis berwarna ungu yang mewakili nilai turunan numerik dengan metode selisih pusat berada sangat dekat bahkan hampir berhimpit dengan garis interpolasi $f(x)$ yang berwarna biru.

Aplikasi Turunan Numerik Dengan Metode Selisih Menggunakan Maple

Selain menggunakan Python, penelitian ini juga merancang dan mengimplementasikan program numerik menggunakan perangkat lunak Maple. Rangkaian perintah yang disusun dalam *worksheet* Maple ini mampu menghitung turunan pertama dari fungsi eksplisit yang kompleks maupun data berupa nilai diskrit, dengan menerapkan pendekatan metode selisih. Gambar hasil komputasi untuk kasus turunan numerik dalam bentuk fungsi eksplisit yang rumit dengan menggunakan Maple adalah sebagai berikut:



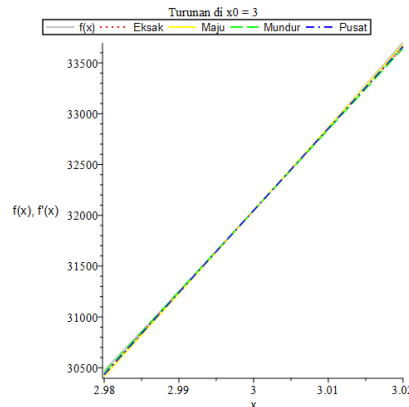
=== TURUNAN NUMERIK UNTUK $f(x)$ ===

```
df:= x->rtable(1..1, {1 = e^{2*x+2}.ln(4*x^2) + 2*x*e^{2*x+2}.ln(4*x^2) + 2*e^{2*x+2}}, datatype = anything, subtype = Vector, columnwise storage = rectangular, order = Fortran_order)
x0 := 3
h := 0.01
f_xmh := [ 31249.33685 ]
f_x0 := [ 6 e^8 ln(6) ]
f_xph := [ 32864.25568 ]
df_maju := [ 3.286425568 x 10^6 - 600. e^8 ln(6) ]
df_mundur := [ 600. e^8 ln(6) - 3.124933685 x 10^6 ]
df_pusat := [ 80745.9415000001 ]
df_exact := [ 14 e^8 ln(6) + 2 e^8 ]
galat_maju := [ 3.286425568 x 10^6 - 614. e^8 ln(6) - 2 e^8 ]
galat_mundur := [ -586. e^8 ln(6) + 3.124933685 x 10^6 + 2 e^8 ]
galat_pusat := [ 80745.9415000001 - 14 e^8 ln(6) - 2 e^8 ]
```

Metode	Turunan	Galat
Maju	81729.74800000	9.92e+02
Mundur	79762.13500000	9.76e+02
Pusat	80745.94150000	7.79e+00
Eksak	80738.15177000	

Gambar 6. Penyelesaian Fungsi Eksplisit Rumit dengan Maple

Berdasarkan hasil perhitungan pada Gambar 12, diperoleh nilai eksak sebesar 80738.15177000, nilai hampiran selisih maju sebesar 81729.74800000 dengan galat mutlak sebesar 9.92e+02, nilai hampiran selisih mundur sebesar 79762.13500000 dengan galat mutlak sebesar 9.76e+02, serta nilai hampiran selisih pusat sebesar 80745.941500000 dengan galat mutlak sebesar 7.79e+00. Berdasarkan hasil komputasi Maple tersebut, nilai galat mutlak untuk masing-masing metode menunjukkan bahwa metode selisih pusat menghasilkan nilai turunan yang paling mendekati nilai eksak. Hal ini juga terlihat dari visualisasi grafik dari masing-masing metode pada gambar di bawah ini:



Gambar 7. Grafik Turunan Numerik Fungsi Eksplisit Rumit dengan Maple

Grafik pada Gambar 13. tersebut menunjukkan nilai turunan numerik metode selisih maju yang berupa garis solid berwarna kuning, nilai turunan numerik metode selisih mundur yang berupa garis strip berwarna hijau, nilai turunan numerik metode selisih pusat yang berupa garis strip putus berwarna biru, dan nilai eksak berupa garis putus-putus berwarna



merah, serta nilai $f(x)$ berupa garis solid berwarna abu-abu. Terlihat bahwa garis hasil metode selisih pusat berada paling dekat dengan garis turunan eksak. Ini menunjukkan bahwa pendekatan metode selisih pusat memberikan estimasi turunan yang lebih akurat pada titik yang dianalisis. Namun, grafik visualisasi tersebut belum begitu jelas jika dibandingkan dengan grafik visualisasi dengan Python.

Hasil komputasi untuk kasus turunan numerik dalam bentuk data diskrit dengan menggunakan Maple akan disajikan pada gambar sebagai berikut:

```

=== TURUNAN NUMERIK DARI DATA TABEL ===

                                xdata := [1.3, 1.5, 1.7, 1.9, 2.1, 2.3, 2.5]
                                fddata := [3.669, 4.482, 5.474, 6.686, 8.166, 9.974, 12.182]
                                n := 7

                                i      x0      Maju      Mundur      Pusat

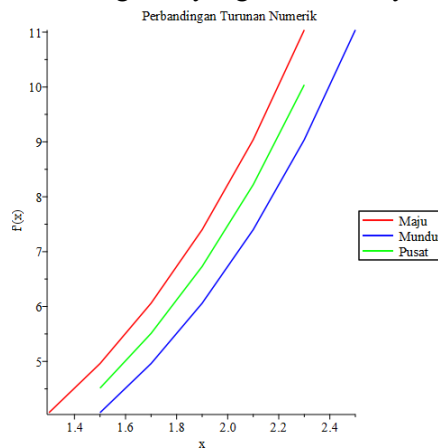
                                -----

                                pts_fwd := [ ]
                                pts_bwd := [ ]
                                pts_ctr := [ ]

                                1      1.30      4.065000      -            -
                                2      1.50      4.960000      4.065000     4.512500
                                3      1.70      6.060000      4.960000     5.510000
                                4      1.90      7.400000      6.060000     6.730000
                                5      2.10      9.040000      7.400000     8.220000
                                6      2.30      11.040000     9.040000     10.040000
                                7      2.50      -            11.040000     -
  
```

Gambar 8. Visualisasi Hasil Data Diskrit dengan Maple

Gambar 14. menunjukkan bahwa perhitungan menggunakan rangkaian perintah yang dikembangkan pada penelitian ini berlaku untuk mencari nilai turunan numerik dengan metode selisih di semua titik. Adapun visualisasi grafik yang dihasilkan, yaitu:



Gambar 9. Grafik Turunan Numerik Data Diskrit dengan Maple

Grafik pada Gambar 15. menunjukkan masing-masing nilai hampiran metode selisih maju yang berupa garis solid berwarna merah, hampiran metode selisih mundur yang berupa garis solid berwarna biru, dan hampiran metode selisih pusat yang berupa garis solid berwarna hijau.



Perbandingan Hasil Metode Selisih pada Python dan Maple

Tabel berikut menunjukkan hasil pendekatan menggunakan metode selisih maju dan mundur untuk fungsi eksplisit, beserta nilai eksak dan galatnya.

Tabel 2. Perbandingan Hasil Metode Selisih pada Fungsi Eksplisit

Metode Selisih	Nilai Eksak	Python	Maple	Galat
Maju	80738.15177000	81729.7471966416	81729.74800000	9.92e+02
Mundur	80738.15177000	79762.1357149612	79762.13500000	9.76e+02
Pusat	80738.15177000	80745.9414558014	80745.94150000	7.79e+00

Berdasarkan tabel di atas, aproksimasi turunan menggunakan metode selisih maju, metode selisih mundur, dan metode selisih pusat menggunakan Python dan Maple bernilai hampir sama. Kemudian, tabel selanjutnya menyajikan hasil pendekatan turunan menggunakan metode selisih maju, mundur, dan pusat pada titik-titik yang sesuai:

Tabel 3. Perbandingan Hasil Metode Selisih pada Data Nilai-Nilai Diskrit

Aplikasi Komputasi	Nilai x	Python	Maple
Selisih Maju	1.3	4.07	4.065000
	1.5	4.96	4.960000
	1.7	6.06	6.060000
	1.9	7.40	7.400000
	2.1	9.04	9.040000
	2.3	11.04	11.040000
	2.5	nan	-
Selisih Mundur	1.3	nan	-
	1.5	4.07	4.065000
	1.7	4.96	4.960000
	1.9	6.06	6.060000
	2.1	7.40	7.400000
	2.3	9.04	9.040000
	2.5	11.04	11.040000
Selisih Pusat	1.3	nan	-
	1.5	4.51	4.512500
	1.7	5.51	5.510000
	1.9	6.73	6.730000
	2.1	8.22	8.220000
	2.3	10.4	10.040000
	2.5	nan	-

Berdasarkan tabel di atas, aproksimasi turunan menggunakan metode selisih maju, metode selisih mundur, dan metode selisih pusat menggunakan Python dan Maple bernilai sama.

Selanjutnya, masing-masing metode dibahas secara terpisah untuk mengkaji kecenderungan hasil dan performa pendekatannya terhadap kedua kasus:



1. Hasil Metode Selisih Maju

Pada kasus turunan numerik berbentuk fungsi eksplisit yang rumit diperoleh nilai turunan numerik dengan menggunakan Python sebesar 81729.7471966416 sedangkan dengan menggunakan Maple sebesar 81729.74800000. Adapun galat mutlak untuk metode selisih maju adalah $9.92e+02$. Kemudian, pada kasus data dalam bentuk nilai-nilai diskrit diperoleh nilai turunan numerik sebagai berikut:

Tabel 4. Perbandingan Hasil Metode Selisih Maju pada Data Nilai-Nilai Diskrit

Nilai x	Aplikasi Komputasi	
	Python	Maple
1.3	4.07	4.065000
1.5	4.96	4.960000
1.7	6.06	6.060000
1.9	7.40	7.400000
2.1	9.04	9.040000
2.3	11.04	11.040000
2.5	nan	-

Kedua aplikasi komputasi baik Python maupun Maple menghasilkan nilai turunan numerik metode selisih maju yang hampir sama pada kedua kasus yang diteliti. Namun, pada Maple, jumlah digit di belakang koma tampak lebih sedikit. Hal ini disebabkan oleh pengaturan tampilan bawaan Maple yang secara *default* membatasi jumlah digit signifikan yang ditampilkan.

2. Hasil Metode Selisih Mundur

Pada kasus turunan numerik berbentuk fungsi eksplisit yang rumit diperoleh nilai turunan numerik dengan menggunakan Python sebesar 79762.1357149612 sedangkan dengan menggunakan Maple sebesar 79762.13500000. Adapun galat mutlak untuk metode selisih mundur adalah $9.76e+02$. Kemudian, pada kasus data dalam bentuk nilai-nilai diskrit diperoleh nilai turunan numerik sebagai berikut:

Tabel 5. Perbandingan Hasil Metode Selisih Mundur pada Data Nilai-Nilai Diskrit

Nilai x	Aplikasi Komputasi	
	Python	Maple
1.3	nan	-
1.5	4.07	4.065000
1.7	4.96	4.960000
1.9	6.06	6.060000
2.1	7.40	7.400000
2.3	9.04	9.040000
2.5	11.04	11.040000

Sama dengan hampiran selisih maju, aplikasi komputasi Python maupun Maple menghasilkan nilai turunan numerik metode selisih mundur yang hampir sama.



3. Hasil Metode Selisih Pusat

Pada kasus turunan numerik berbentuk fungsi eksplisit yang rumit diperoleh nilai turunan numerik dengan menggunakan Python sebesar 80745.9414558014 sedangkan dengan menggunakan Maple sebesar 80745.94150000. Adapun galat mutlak untuk metode selisih pusat adalah $7.79e+00$. Kemudian, pada kasus data dalam bentuk nilai-nilai diskrit diperoleh nilai turunan numerik sebagai berikut:

Tabel 5. Perbandingan Hasil Metode Selisih Pusat pada Data Nilai-Nilai Diskrit

Nilai x	Aplikasi Komputasi	
	Python	Maple
1.3	nan	-
1.5	4.51	4.512500
1.7	5.51	5.510000
1.9	6.73	6.730000
2.1	8.22	8.220000
2.3	10.4	10.040000
2.5	nan	-

Sama dengan hampiran selisih maju dan selisih mundur, aplikasi komputasi Python maupun Maple menghasilkan nilai turunan numerik metode selisih pusat yang hampir sama pada kedua kasus yang diteliti. Berdasarkan hasil komputasi dengan aplikasi Python maupun, nilai galat mutlak untuk metode selisih pusat cenderung lebih kecil dibandingkan dengan metode selisih maju dan mundur.

Berdasarkan uraian di atas, hampiran selisih pusat menghasilkan aproksimasi turunan yang paling mendekati nilai eksak. Hal ini menunjukkan bahwa metode selisih pusat menghasilkan aproksimasi yang lebih akurat dibandingkan dengan metode selisih maju ataupun metode selisih mundur. Hal ini sejalan dengan penelitian Sanaky (2021: 21) yang menyatakan bahwa penggunaan metode selisih pusat lebih akurat dibandingkan dengan metode selisih maju dan metode selisih mundur. Selain itu, penelitian oleh Tukan, dkk. (2024: 21) juga memperoleh temuan yang sama, di mana ketepatan metode selisih pusat lebih baik dibandingkan metode selisih maju dan mundur. Hal ini dikarenakan metode selisih pusat memanfaatkan nilai fungsi dari dua sisi titik evaluasi, sehingga pendekatannya lebih seimbang.

Pembahasan

Bagian ini membandingkan dua aplikasi komputasi, yaitu Python dan Maple dalam mengimplementasikan metode selisih baik selisih maju, mundur, maupun pusat pada dua jenis kasus, yakni fungsi eksplisit yang rumit dan data diskrit. Studi komparatif dilakukan dengan mempertimbangkan aspek keakuratan hasil perhitungan dilihat dari galat mutlak yang dihasilkan dan aspek kemudahan implementasi, seperti proses pemrograman, tampilan *output*, dan kecepatan eksekusi.



Keakuratan Hasil Perhitungan

Berdasarkan Tabel 2. nilai hampiran metode selisih maju, selisih mundur, dan selisih pusat kedua aplikasi komputasi adalah hampir sama, tetapi terdapat perbedaan presisi pada angka dibelakang koma. Maple secara *default* menampilkan angka dengan digit yang lebih sedikit, sedangkan Python mampu menampilkan lebih banyak angka di belakang koma. Pada penelitian ini, hasil komputasi Python untuk fungsi eksplisit yang rumit diformat menampilkan sepuluh digit di belakang koma. Sebagai contoh, Maple menghasilkan 81729.74800000, sedangkan Python menampilkan 81729.7471966416 untuk perhitungan yang sama. Dilansir dari *website* resmi Maple dan Python, hal ini disebabkan oleh pengaturan *default* Maple yang membatasi jumlah digit tampilan, sementara Python menampilkan lebih banyak digit dan memungkinkan pengaturan jumlah digit menggunakan format *string*.

Kemudian, pengerjaan turunan numerik baik menggunakan Python maupun Maple memiliki akurasi yang tinggi terhadap turunan eksaknya. Hal ini ditunjukkan dengan melihat pada galat mutlak dari kedua aplikasi komputasi yang identik, ini menunjukkan bahwa kedua aplikasi memiliki akurasi komputasi yang setara. Kesamaan ini disebabkan oleh metode yang digunakan yaitu metode selisih serta kesamaan titik awal dan langkah h yang digunakan pada proses perhitungan. Temuan ini sejalan dengan penelitian oleh Arvil dkk., (2024) yang menyatakan bahwa Python memberikan verifikasi yang akurat terhadap perhitungan matematika dan juga penelitian oleh Junaidi (2016: 199) yang memaparkan bahwa Maple memberikan penyelesaian soal-soal matematika dengan akurat.

Selanjutnya, mengacu pada tabel 3, pada kasus hasil turunan numerik untuk data dalam bentuk nilai-nilai diskrit diperoleh nilai hampiran selisih maju, selisih mundur, dan selisih pusat pada Python disajikan dengan pembulatan dua digit. Hal ini berbeda dengan Maple yang nilai hampirannya tidak dibulatkan. Namun, untuk data diskrit ini tidak dilakukan perhitungan galat karena tidak tersedia nilai turunan analitik sebagai pembanding. Hal ini menjadi salah satu keterbatasan dalam penelitian, karena evaluasi akurasi metode hanya dapat dilakukan pada soal fungsi eksplisit yang rumit. Selain itu, perihal lain yang menjadi keterbatasan penelitian adalah bahwa penelitian ini hanya melibatkan implementasi turunan numerik dengan h sebesar 0,001 untuk kasus turunan numerik berbentuk fungsi eksplisit yang rumit dan h sebesar 0,2 untuk turunan numerik berbentuk data diskrit.

Python dan Maple sama-sama memiliki akurasi tinggi dalam pendekatan turunan numerik. Namun, kedua hasil komputasi memiliki tampilan hasil yang berbeda. Python lebih unggul dalam presisi digit karena fleksibel dalam pengaturan format, sedangkan Maple lebih ringkas untuk penulisan ekspresi matematis.

Kemudahan Implementasi

Penggunaan skrip aplikasi Python dan Maple yang dikembangkan pada penelitian ini sama-sama dikatakan mudah, karena menggunakan algoritma yang sama sehingga pengguna dapat dengan mudah menyelesaikan turunan numerik dengan metode selisih baik pada fungsi



eksplisit rumit maupun data nilai-nilai diskrit. Pada Maple, pengguna dapat langsung menuliskan ekspresi matematika dengan menggunakan *tools* yang tersedia. Tung (2005: 2) mengatakan bahwa Maple menyediakan pilihan komputasi dari berbagai fungsi atau model matematika. Hal ini diperkuat oleh temuan dalam penelitian lain oleh Rejeki dkk., (2025) yang memaparkan bahwa Maple lebih mudah digunakan karena memiliki fungsi bawaan. Sedangkan pada Python pengguna harus lebih hati-hati dalam menginput fungsi karena tidak tersedia *tools* seperti pada Maple. Meskipun demikian, Python lebih unggul dalam hal kecepatan eksekusi perhitungan dibandingkan dengan Maple. Hal ini sejalan dengan penelitian Talia, dkk. (2024: 46) yang menyatakan bahwa Python dapat melakukan komputasi yang rumit secara cepat. Penelitian oleh Robby dkk., (2023: 3) juga menyebutkan bahwa Python dapat melakukan proses perhitungan numerik dan simbolik dalam skala besar dengan menggunakan pustaka *numpy* dan *sympy*.

Penggunaan Python dan Maple sama-sama memudahkan penyelesaian turunan numerik karena menggunakan algoritma yang serupa. Maple lebih unggul dari segi kemudahan penggunaan karena menyediakan *tools* bawaan untuk menuliskan fungsi secara langsung. Sebaliknya, Python membutuhkan ketelitian lebih dalam penulisan sintaks, tetapi lebih unggul dalam kecepatan eksekusi komputasi.

SIMPULAN

Berdasarkan hasil komputasi turunan numerik dengan metode selisih menggunakan Python dan Maple, diketahui bahwa nilai aproksimasi turunan dengan metode selisih pusat paling mendekati nilai turunan eksak. Kedua aplikasi komputasi dapat dimanfaatkan untuk menyelesaikan turunan, baik pada fungsi eksplisit yang rumit maupun data nilai-nilai diskrit, dengan mengimplementasikan metode selisih. Hasil komputasi menggunakan Python maupun Maple menghasilkan nilai turunan numerik metode selisih pusat yang hampir sama. Nilai galat mutlak untuk metode selisih pusat cenderung lebih kecil dibandingkan dengan metode selisih maju dan mundur. Hal ini menunjukkan bahwa metode selisih maju menghasilkan aproksimasi yang lebih akurat dibandingkan dengan metode selisih maju ataupun metode selisih mundur.

Kemudian, melihat pada aspek akurasi hasil komputasi, kedua aplikasi memiliki akurasi yang tinggi terhadap turunan eksaknya. Hal ini ditunjukkan dengan melihat pada galat mutlak dari kedua aplikasi komputasi yang identik. Selanjutnya, melihat pada aspek kemudahan penggunaannya, kedua aplikasi ini sama-sama dikatakan mudah, karena menggunakan algoritma yang sama. Maple unggul dalam hal penulisan fungsi yang akan diinput, sedangkan Python lebih unggul dalam hal kecepatan eksekusi perhitungan dibandingkan dengan Maple. Penelitian Penelitian ini memiliki beberapa keterbatasan, di antaranya hanya difokuskan pada pendekatan numerik untuk menentukan turunan dari satu fungsi eksplisit kompleks yang berupa kombinasi fungsi aljabar, eksponensial, dan logaritmik dan satu kasus data diskrit. Kemudian, evaluasi akurasi hanya dilakukan pada fungsi eksplisit, karena data diskrit tidak dihitung galatnya. Penelitian ini juga dibatasi pada penggunaan nilai



$h = 0,001$ untuk fungsi eksplisit dan $h = 0,2$ untuk data diskrit. Penelitian selanjutnya disarankan mengeksplorasi aplikasi komputasi lain, metode turunan numerik yang berbeda, serta variasi nilai h .

DAFTAR PUSTAKA

- Arvil, S., Triadi, I. S., Putri, Z., Lubis, R. A., & Fitriyani, F. (2024). Penggunaan Python dalam Pengerjaan Induksi Matematika. *Jurnal Ilmiah Matematika, Kebumihan Dan Angkasa*, 2(5), 17–25.
- Enterprise, J. (2017). *Otodidak Pemrograman Python*. PT Elex Media Komputindo.
- Imron, C., Mardlijah, & Asfihani, T. (2022). Metode Numerik. In *Informatika*. dMath.
- Judijanto, L., Manu, C. M. A., Sitopu, J. W., Mangelep, N. O., & Hardiansyah, A. (2024). The Role of Mathematics in Science and Technology Development. *International Journal of Teaching and Learning (INJOTEL)*, 2(2), 451–458.
- Junaidi. (2016). Penggunaan Software Maple dalam Pembelajaran Matematika pada Materi Integral. *Visipena Journal*, 7(2).
- Maple. (n.d.). *Digits*.
- Maulida Surbakti, N., Angelyca, Talia, A., Br Perangin-Angin, C., Olivia Nainggolan, D., Devi Friskauly, N., & Ruth Br Tumorang, S. (2024). Penggunaan Bahasa Pemrograman Python dalam Pembelajaran Kalkulus Fungsi Dua Variabel. *Algoritma: Jurnal Matematika, Ilmu Pengetahuan Alam, Kebumihan Dan Angkasa*, 2(3), 98–107.
- Maure, O. P., & Mungkasi, S. (2021). Verifikasi Tingkat Keakuratan Beberapa Metode Integrasi Numerik Fungsi Atas Satu Peubah Bebas. *JURNAL SILOGISME: Kajian Ilmu Matematika Dan Pembelajarannya*, 6(1), 58. <https://doi.org/10.24269/silogisme.v6i1.3540>
- Nazar, R. (2024). Implementasi Pemrograman Python Menggunakan Google Colab. *Jurnal Informatika Dan Komputer (JIK)*, 15(1), 50–56.
- Neldiana, R., & Harisman, Y. (2025). Peranan Integral dan Diferensial Terhadap Kalkulus. *Jurnal Cendekia: Jurnal Pendidikan Matematik*, 9(1), 273–284.
- Nur Aini, A., Wulandari, D., Sutrisno, & Buchori, A. (2017). *Aplikasi Teknologi Pembelajaran dengan Maple*. Universitas PGRI Semarang.
- Pongsapan, F. P., & Pawarangan, I. (2021). Penggunaan Bahasa Pemrograman Maple untuk Menyelesaikan Masalah Mekanika Gerak dalam Fisika. *Journal Dynamic Saint*, 6(1), 31–35. <https://doi.org/10.47178/dynamicsaint.v6i1.1273>
- Python. (n.d.). *Aritmatika Pecahan Floating Point: Masalah dan Keterbatasan*.
- Rahmadi, D., Naufal, M. H., Rohman, A. N., & Setyaji, F. D. (2024). Kompresi Gambar Menggunakan Singular Value Decomposition (SVD) dengan Python. *Jurnal Ilmiah Matematika Dan Pendidikan Matematika*, 16(1), 57–72.
- Rahmatudin, J., & Sumliyah. (2024). *Pemrograman Python*. PT Literasi Nusantara Abadi Grup.
- Ramdhani, V. (2021). Penggunaan Software Maple pada Pembelajaran Persamaan Diferensial Biasa. *GAUSS: Jurnal Pendidikan Matematika*, 4(1), 70–80. <https://doi.org/10.30656/gauss.v4i1.2985>
- Rejeki, T. B., Lesmana, A. C., & Wibowo, A. (2025). Perbandingan Penggunaan Matlab dan Maple dalam Menentukan Akar Persamaan Non-Linier Menggunakan Metode Bagi Dua. 7, 624–638.



- Rifa'i, M. (2020). *Kalkulus Diferensial*. Penerbit Deepublish.
- Rinaldi Munir. (2015). *Metode Numerik: Vol. II* (Issue 1).
- Robby, M., Anwar, B. P., Nurmala, M., Listanto, M., & Rosyani, P. (2023). *Analisis Penerapan Python Dalam Perhitungan Turunan Fungsi*. 1(1), 1–6.
- Samaray, S. (2023). Perbandingan Metode Diferensiasi Numerik Berbasis Matlab Mobile. *Seminar Nasional Corisindo*, 44–49.
- Sanaky, M. M. (2021). Analisis Faktor-Faktor Keterlambatan Pada Proyek Pembangunan Gedung Asrama Man 1 Tulehu Maluku Tengah. *Jurnal Simetrik*, 11(1), 432–439. <https://doi.org/10.31959/js.v11i1.615>
- Sroyer, A., Yanti, W., Manurung, M. M. H., Abraham, Sumarni, Malmia, W., Fajriana, Tahir, U., Nugroho, D. H., Maja, I., & Tandianga, T. (2024). *Kalkulus 1*. Literasi Langsung Terbit.
- Talia, A., Tumanggor, A. A., & Hasibuan, F. (2024). *Implementasi Metode Numerik dan Simbolik dengan Python untuk Penentuan Nilai Limit Fungsi*. 4.
- Tukan, H., Maure, O. P., & Ina, K. T. D. (2024). Analisis Keakuratan Metode Numerik Dalam Menyelesaikan Turunan Persamaan Nonlinier. *Leibniz: Jurnal Matematika*, 4(2), 10–22. <https://doi.org/10.59632/leibniz.v4i02.446>
- Tung, K. Y. (2005). *Komputasi Simbolik Fisika Mekanika Berbaiss Maple*. ANDI.
- Wibowo, W., Ulama, B. S. S., & Azies, H. Al. (2020). *Belajar Pemrograman Bahasa Python*. ITS PRESS.
- Zakaria, L., & Muharramah, U. (2023). *Pengantar Metode Numerik (Solusi Masalah Dengan Matematika)*. CV Anugrah Utama Raharja.