

Penerapan Python Dengan Perhitungan Kalkulus Dalam Fungsi Turunan

Rama Achmad Fadillah^{1*}, Nicholas^{2*}, Teguh Riyan Susanto^{3*}, Fajar Iswandi^{4*}, Perani Rosyan^{5*}

^{1,2,3,4,5} Teknik Informatika, Universitas Pamulang, Tangerang, Selatan, Indonesia

*Email: ¹adenfadill28@gmail.com, ²nichooo2104@gmail.com, ³teguhriyan108@gmail.com,
⁴v2z43nq13@gmail.com, ⁵dosen00837@unpam.ac.id

(*: corresponding author)

Abstrak-Turunan merupakan salah satu konsep dasar dalam matematika yang memiliki banyak aplikasi dalam berbagai bidang, seperti fisika, kimia, ekonomi, dan teknik. Turunan dapat digunakan untuk menentukan gradien, kecepatan, percepatan, dan berbagai besaran lainnya. Jurnal ini, akan dibahas tentang konsep turunan, termasuk definisi, sifat-sifat, dan aplikasinya. Pembahasan akan dimulai dengan definisi turunan fungsi polinomial, kemudian dilanjutkan dengan turunan fungsi trigonometri, fungsi eksponen, dan fungsi logaritma. Selanjutnya, akan dibahas tentang turunan fungsi komposisi, turunan fungsi implicit, dan turunan fungsi parsial. Selain itu, jurnal ini juga membahas penerapan Python dalam kalkulus diferensial. Python adalah bahasa pemrograman yang populer dan memiliki banyak perpustakaan matematika yang dapat digunakan untuk menghitung turunan secara numerik atau simbolis. Python membuat perhitungan diferensial lebih mudah dan efisien. Perhitungan kalkulus dengan contoh perhitungan diferensial menggunakan Python dan juga perpustakaan seperti SymPy dan NumPy. Kasus terhitung antara lain turunan polinomial, fungsi trigonometri, fungsi eksponensial, dan fungsi logaritma. Kami juga membahas penggunaan Python dalam menghitung fungsi komposit, fungsi implisit, dan turunan fungsi parsial.

Kata kunci: Turunan, Python, Numerik, Simbolis, Kalkulus diferensial, SymPy, NumPy

Abstract-Derivative is one of the basic concepts in mathematics that has many applications in various fields, such as physics, chemistry, economics, and engineering. Derivatives can be used to determine gradient, velocity, acceleration, and various other quantities. This journal will discuss the concept of derivative, including its definition, properties, and applications. The discussion will start with the definition of the derivative of polynomial functions, then continue with the derivative of trigonometric functions, exponential functions, and logarithmic functions. Furthermore, it will discuss the derivative of the composition function, the derivative of the implicit function, and the derivative of the partial function. Python is a popular programming language and has many mathematical libraries that can be used to calculate derivatives numerically or symbolically. Python makes differential calculations easier and more efficient. Calculus calculations with differential calculation examples using Python and also libraries such as SymPy and NumPy. Calculated cases include polynomial derivatives, trigonometric functions, exponential functions, and logarithmic functions. We also discuss the use of Python in calculating composite functions, implicit functions, and partial derivatives of functions.

Keywords: Derivative, Python, Numerical, Symbolic, Differential calculus, SymPy, NumPy

1. PENDAHULUAN

Kalkulus diferensial adalah cabang kalkulus dalam matematika yang mempelajari bagaimana nilai suatu fungsi berubah menurut perubahan input nilainya. Topik utama kalkulus diferensial adalah turunan, yang menjelaskan sifat-sifat fungsi yang mendekati nilai. Untuk fungsi yang bernilai real dengan variabel real tunggal, turunan pada sebuah titik sama dengan kemiringan dari garis singgung grafik fungsi pada titik.

Perhitungan kalkulus dengan konteks aplikasi Python mengacu pada proses melakukan perhitungan dalam konteks atau lingkungan tertentu dalam pemrograman Python dengan mencakup kalkulasi yang melibatkan variabel, fungsi, atau objek dalam konteks tertentu, dan telah ditinjau oleh para ahli di bidangnya, dapat disetujui untuk dipublikasikan, diakui.

Penelitian ini bermaksud untuk memanfaatkan fleksibilitas Python untuk analisis matematis dan komputasi ilmiah. Python, dengan ekosistemnya yang kaya akan perpustakaan seperti NumPy, SciPy, dan SymPy, menyediakan platform untuk menghitung turunan, yang dapat memberikan wawasan berharga tentang perilaku, tren, dan karakteristik fungsi matematika.

Daripada itu juga, Penggunaan Python dalam perhitungan kalkulus juga relevan untuk aplikasi kecerdasan buatan, di mana pemahaman komprehensif tentang matematika, termasuk kalkulus, sangat penting untuk mengembangkan solusi kecerdasan buatan yang tangguh. khususnya fungsi pada turunan.

Dengan memahami lebih lanjut peran yang telah ada pada python, maka kesederhanaan dan kemudahan membaca Python dapat diakses oleh mahasiswa dengan berbagai tingkat keahlian, mulai dari pemula yang sedang belajar dasar-dasar kalkulus hingga peneliti yang berpengalaman dan sedang menjelajahi topik-topik lanjutan.

2. METODE PENELITIAN

Terdapat beberapa metode penelitian yang dapat diterapkan menggunakan Python dalam perhitungan kalkulus khususnya pada fungsi turunan. Python dapat digunakan sebagai alat untuk memahami, menganalisis, dan memecahkan permasalahan yang melibatkan kalkulus diferensial, terutama dalam penghitungan turunan fungsi. Dengan menggunakan pustaka-pustaka seperti NumPy, SciPy, SymPy, dan algoritma-algoritma khusus, Python dapat menjadi alat yang kuat dalam menganalisis dan memodelkan fenomena matematika yang melibatkan turunan. Berikut beberapa metode yang umum digunakan:

2.1 Metode Beda Hingga (Finite Difference Method):

Metode ini menggunakan pendekatan numerik untuk menghitung turunan suatu fungsi dengan memanfaatkan beda hingga. Hal ini melibatkan pendekatan persamaan diferensial dengan menggunakan beda hingga untuk mendekati turunan. Implementasi dengan Python dapat dilakukan dengan menghitung perbedaan antara nilai-nilai fungsi pada titik-titik yang berdekatan. Contohnya, menggunakan pendekatan *forward*, *backward*, atau *central difference*.

2.2 Penggunaan Pustaka Matematika dalam Python:

Memanfaatkan pustaka seperti *NumPy*, *SciPy*, *SymPy* untuk menghitung turunan secara simbolis atau numerik. Anda bisa memanfaatkan fungsi-fungsi seperti *scipy.misc.derivative* untuk perhitungan turunan numerik, melakukan perhitungan turunan secara simbolis. Ini memungkinkan untuk bekerja dengan ekspresi matematika secara langsung.

2.3 Optimasi dengan Turunan:

Memanfaatkan turunan untuk menyelesaikan masalah optimasi. Python dapat digunakan untuk mengimplementasikan algoritma optimasi yang memanfaatkan informasi dari turunan untuk mencari nilai maksimum atau minimum suatu fungsi.

3. ANALISIS DAN PEMBAHASAN

3.1 Analisa & Masalah Penelitian dalam Perhitungan Kalkulus

Dalam penelitian yang dilakukan oleh Imelda dan Pasaribu (2019) menunjukkan bahwa mahasiswa mengalami kesulitan dalam memahami konsep turunan. Kesulitan tersebut meliputi kesulitan

dalam memahami pengertian turunan, perbedaan turunan pertama dan turunan kedua, serta aplikasi turunan dalam berbagai bidang.

Penelitian yang dilakukan oleh Kurniawan dan Prasetyo (2020) menunjukkan bahwa mahasiswa mengalami kesulitan dalam menerapkan konsep turunan dalam menyelesaikan soal. Kesulitan tersebut meliputi kesulitan dalam memilih metode yang tepat untuk mencari turunan, kesulitan dalam melakukan operasi perhitungan, serta kesulitan dalam memahami makna jawaban.

Wibowo dan Hartono (2021) menunjukkan bahwa mahasiswa mengalami kesulitan dalam operasi perhitungan turunan. Kesulitan tersebut meliputi kesulitan dalam menggunakan rumus turunan, kesulitan dalam melakukan operasi perkalian, pembagian, dan perpangkatan, serta kesulitan dalam melakukan operasi logaritma dan eksponen.

Penyelsaian Setiawan dan Nugroho (2018) menunjukkan bahwa turunan memiliki banyak aplikasi dalam berbagai bidang, seperti matematika, fisika, ekonomi, dan teknik.

Astuti dan Setiawan (2020) menunjukkan bahwa turunan dapat digunakan untuk menyelesaikan berbagai masalah dalam kehidupan sehari-hari, seperti menghitung laju pertumbuhan bakteri, menentukan jarak yang ditempuh mobil, dan menghitung biaya produksi.

3.2 Keunggulan Python dalam Penelitian dalam Perhitungan Kalkulus

Python dapat mengevaluasi fungsi-fungsi dengan memeriksa Hal yang dapat memberikan pengalaman menggunakan Python untuk menyelesaikan masalah kalkulus secara simbolis maupun numerik. Geoffrey Poore (2011).

Metode numerik untuk memecahkan persamaan. Hal ini berkaitan dengan bagaimana komputer benar-benar menyelesaikan berbagai persamaan fisika. Tidak hanya mempelajari matematika dan teori di balik metode komputasi, melainkan menggunakan program sendiri untuk menyelesaikan masalah. Geoffrey Poore (2011).

Menggunakan Python untuk merencanakan dan menyelesaikan persamaan dalam pembelajaran tentang fungsi kompleks. Hal ini untuk pengembangan keterampilan merencanakan dan menyelesaikan yang telah dipelajari siswa, karena fungsi-fungsi ini memerlukan metode yang agak berbeda. Geoffrey Poore (2011).

Python adalah pilihan yang bagus untuk banyak aplikasi matematika komputasional berkat kombinasi NumPy dan SymPy. Seperti yang dijelaskan oleh Harris et al. (2020) dan Meurer et al. (2017), keunggulan integrasi memungkinkan pengguna untuk memanfaatkan keuntungan dari perhitungan numerik dan simbolik secara bersamaan. Keunggulan tambahan dalam pengembangan dan penelitian ilmiah berasal dari kemudahan penggunaan Python untuk mengedit data numerik dan simbolik.

Kemampuan perhitungan matematika komputasional diperluas oleh NumPy dan SymPy. NumPy, pustaka dasar dalam ekosistem Python, mendukung operasi numerik dengan vektorisasi. Harris et al. (2020) menekankan bahwa NumPy memungkinkan manipulasi array numerik dengan kecepatan tinggi. Ini membawa NumPy ke dalam keunggulan dalam perhitungan numerik kompleks, yang biasanya ditemukan dalam analisis data, ilmu komputer, dan pengolahan sinyal.

Python telah menjadi bahasa pemrograman yang sangat penting untuk pengembangan perangkat lunak matematika karena banyak digunakan di berbagai industri dan disiplin ilmu. Menurut VanderPlas (2016), Python memiliki ekosistem yang aktif dan dinamis. berkembang, dengan dukungan aktif dari komunitas. Keunggulan Python dalam perhitungan turunan menciptakan fondasi yang kokoh untuk inovasi dan perkembangan lebih lanjut dalam bidang matematika komputasional.

3.3 Pelaksanaan Perhitungan Turunan dengan Python

3.3.1 SymPy

Perhitungan Turunan dengan Python dapat dilakukan dengan menggunakan library seperti SymPy. SymPy adalah library Python yang digunakan untuk matematika simbolis, yang dapat digunakan untuk melakukan perhitungan turunan.

```
main.py
1 import sympy as sp
2
3 # Mendefinisikan variabel dan fungsi
4 x = sp.symbols('x')
5 f = x**2 + 3*x + 5
6
7 # Menghitung turunan pertama
8 f_prime = sp.diff(f, x)
9 print("Turunan pertama dari f adalah:", f_prime)
10
11 # Menghitung turunan kedua
12 f_double_prime = sp.diff(f_prime, x)
13 print("Turunan kedua dari f adalah:", f_double_prime)
```

Gambar 1. Code Python SymPy

Code Python SymPy di atas digunakan untuk menghitung turunan pertama dan kedua dari suatu fungsi matematika. terdapat tiga bagian utama yaitu mendefinisikan variabel dan fungsi, menghitung turunan pertama, dan menghitung turunan kedua. Variabel dan fungsi didefinisikan menggunakan fungsi 'symbols' dan '**' digunakan untuk operasi pangkat.

Kemudian, fungsi 'diff' digunakan untuk menghitung turunan pertama dan kedua dari fungsi yang didefinisikan sebelumnya. Hasil perhitungan ditampilkan menggunakan fungsi 'print'. Code tersebut menggunakan library SymPy yang merupakan library Python untuk matematika simbolik.

Dalam kasus ini, kode Python SymPy di atas mendefinisikan fungsi kuadrat sebagai $x^2 + 3x + 5$. Turunan pertama dari fungsi kuadrat adalah dua kali variabel bebasnya, yaitu $2x$. Turunan kedua dari fungsi kuadrat adalah dua kali konstanta, yaitu 4

3.3.2 NumPy

Perhitungan turunan dengan NumPy, kita dapat menggunakan pendekatan numerik dengan menggunakan metode beda hingga (finite difference method). Berikut adalah contoh sederhana bagaimana melakukan perhitungan turunan menggunakan NumPy:

```
main.py
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # Mendefinisikan fungsi
5 def f(x):
6     return x**2 + 3*x + 5
7
8 # Membuat data input
9 x = np.linspace(-5, 5, 100)
10 y = f(x)
11
12 # Menghitung turunan menggunakan metode beda hingga
13 dx = x[1] - x[0]
14 dy_dx = np.gradient(y, dx)
15
16 # Plot fungsi dan turunannya
17 plt.plot(x, y, label='f(x) = x^2 + 3x + 5')
18 plt.plot(x, dy_dx, label='f'(x) (turunan)')
19 plt.legend()
20 plt.show()
21
```

Gambar 2. Code Python NumPy

Code NumPy diatas merupakan penggunaan library NumPy dan Matplotlib untuk menghitung dan memplot fungsi beserta turunannya dengan menyediakan data struktur yang heterogeny, seperti matriks, bidang dan digunakan untuk membuat array x dan y, yang akan digunakan dalam fungsi dan turunan.

Dalam kode ini, Matplotlib digunakan untuk menampilkan fungsi $f(x)$ dan turunannya dengan memanfaatkan fungsi `plt.plot()` dan `plt.legend()`. Sebagai $f(x) = x^2 + 3x + 5$. Fungsi ini akan dihitung untuk setiap nilai x yang berada dalam array x.

3.4 Demonstrasi & Keunggulan dalam Perhitungan Turunan menggunakan Python

Dengan adanya penerapan juga perhitungan turunan menggunakan Python, memungkinkan pengguna untuk melakukan evaluasi ekspresi dengan presisi yang sewenang-wenang, melakukan manipulasi aljabar pada ekspresi simbolik, serta menyelesaikan tugas dasar kalkulus seperti batas, diferensiasi, dan integrasi dengan ekspresi simbolik

Ada pula kelebihan lain nya, seperti, pemrograman Python mudah dipelajari dan dipahami oleh para pemula, digunakan di berbagai system seperti Windows, Mac OS X, juga Linux. Berkemungkinan juga pengguna untuk mengembangkan kode numerik dari ekspresi simbolik, yang berguna untuk menghasilkan kode numerik yang cepat dan akurat.

Sympy memiliki keunggulan dalam perhitungan turunan karena mampu melakukan perhitungan secara simbolis, artinya dapat menangani variabel dan fungsi sebagai simbol matematika, bukan hanya sebagai nilai numerik, di sisi lain, NumPy memiliki keunggulan dalam perhitungan turunan karena mampu melakukan perhitungan dengan cepat dan efisien menggunakan array dan operasi vektorisasi yang mana Kedua library ini memiliki keunggulan masing-masing dalam perhitungan turunan.

4. KESIMPULAN

Studi ini membahas penggunaan Python dalam perhitungan kalkulus diferensial, khususnya turunan, yang menggambarkan sifat-sifat fungsi pada titik tertentu. Kalkulus diferensial mempelajari

bagaimana nilai suatu fungsi berubah terhadap perubahan inputnya. Beberapa pendekatan, termasuk Metode Beda Hingga, penggunaan pustaka matematika seperti NumPy, SciPy, dan SymPy, dan optimasi dengan turunan, digunakan dalam pendekatan penelitian Python. Python mudah digunakan dan fleksibel, terutama berkat ekosistemnya yang kaya akan perpustakaan matematika.

Analisis dan diskusi mencakup masalah yang dihadapi siswa dalam memahami konsep turunan dan aplikasinya. Python dianggap memiliki keunggulan dalam penelitian kalkulus karena dapat menyelesaikan masalah kalkulus secara simbolis maupun numerik serta membantu dalam pengembangan keterampilan pemecahan masalah matematis.

Oleh karena itu, penelitian ini menekankan bahwa Python dapat menjadi alat yang sangat baik untuk memahami, menganalisis, dan memecahkan masalah kalkulus yang berbeda. Ini akan sangat membantu terutama dalam pendidikan tinggi dan penelitian matematika komputasional.

REFERENCES

- Muhammad Razali, Mahmud N. Siregar, Faridawaty Marpaung (2010). *Differential Calculus Analytic*.
Peter Farell (2020). *The Statistics and Calculus with Python Workshop: A comprehensive introduction to mathematics in Python for artificial intelligence applications*. Packt Publishing New York.
Dario Redecic (2019). *Taking Derivatives in python*. Toward Data Science published.
Aris Munandar. (2017). *Kalkulus*. Jakarta: Erlangga.
Thomas, G.B., Finney, R.L., Weir, M., & Giordano, F.R. (2017). *Calculus: Early Transcendentals*. Boston: Pearson Education.
Stewart, J. (2016). *Calculus: Early Transcendentals*. Boston: Cengage Learning.
Python Software Foundation (2011). *General Python FAQ*. Retrieved from <http://docs.python.org/faq/general.html>.
Daher, W. (2006, February 1). *EECS Revamps Course Structure*. *The Tech*, 125 (65), 1 & 17.
SymPy Development Team. <https://www.sympy.org/en/index.html>
SymPy Tutorial: Calculus
NumPy API Team. <https://numpy.org/doc/stable/reference/routines.math.html>
Buku: "Numerical Methods in Engineering with Python" oleh Jaan Kiusalaas.
Dokumentasi SciPy: <https://docs.scipy.org/doc/scipy/reference/tutorial/integrate.html>
Dokumentasi resmi SymPy: <https://docs.sympy.org/latest/index.html>
Buku: "Optimization with Python: How to Solve Linear Programming Problems" oleh Enrique Blair dan Pierre-Louis Poirion