

## Analisis Penggunaan Bahasa Pemrograman Python Untuk Materi Kalkulus Limit

Mochamad Fachriz Azis Wantah<sup>1\*</sup>, Alya Salsabila Az Zahra<sup>2</sup>, Diana Apriyanti<sup>3</sup>, Nursyifa<sup>4</sup>, Perani Rosyani<sup>5</sup>.

<sup>1,2,3,4,5</sup> Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia.

\*Email: <sup>1</sup>[fachrizazizwantah@gmail.com](mailto:fachrizazizwantah@gmail.com), <sup>2</sup>[alyashalsabilaa08@gmail.com](mailto:alyashalsabilaa08@gmail.com), <sup>3</sup>[dianaapriyanti2304@gmail.com](mailto:dianaapriyanti2304@gmail.com), <sup>4</sup>[syf.nr18@gmail.com](mailto:syf.nr18@gmail.com), <sup>5</sup>[dosen00837@gmail.com](mailto:dosen00837@gmail.com)

(\* : corresponding author)

**Abstrak-**Penelitian ini mendalami penggunaan bahasa pemrograman dengan menerapkan metode penghitungan limit menggunakan program Python sebagai pendekatan analisis. Tujuan utama adalah mengidentifikasi dan mengevaluasi karakteristik kritis dari berbagai bahasa pemrograman dalam konteks kompleksitas kode. Metode penghitungan limit digunakan untuk mengukur sejumlah faktor, termasuk efisiensi eksekusi, kompleksitas algoritma, dan keterbacaan kode. Hasil analisis memberikan wawasan mendalam tentang performa relatif berbagai bahasa pemrograman dalam skenario pengembangan perangkat lunak. Penelitian ini tidak hanya memfasilitasi pemahaman lebih baik tentang kekuatan dan kelemahan masing-masing bahasa pemrograman tetapi juga memberikan dasar untuk pemilihan bahasa yang optimal sesuai dengan kebutuhan proyek. Implikasi praktis dari temuan ini diharapkan dapat meningkatkan kualitas pengembangan perangkat lunak dan efisiensi pemrograman.

**Kata Kunci:** Python, Limit, Pemrograman, Implikasi Praktis, Efisiensi Pemrograman

**Abstract-** This research explores the use of programming languages by applying the limit calculation method using the Python program as an analytical approach. The main goal is to identify and evaluate the critical characteristics of various programming languages in the context of code complexity. The limit calculation method is used to measure a number of factors, including execution efficiency, algorithm complexity, and code readability. The analysis results provide deep insight into the relative performance of different programming languages in software development scenarios. This research not only facilitates a better understanding of the strengths and weaknesses of each programming language but also provides a basis for optimal language selection according to project requirements. The practical implications of these findings are expected to improve the quality of software development and programming efficiency.

**Keywords:** Python, Limits, Programming, Practical Implications, Programming Efficiency

### 1. PENDAHULUAN

Dalam matematika, konsep limit digunakan untuk menjelaskan perilaku suatu fungsi saat peubah bebasnya mendekati suatu titik tertentu, atau menuju tak hingga; atau perilaku dari suatu barisan saat indeks mendekati tak hingga. Limit dipakai dalam kalkulus (dan cabang lainnya dari analisis matematika) untuk membangun pengertian kekontinuan, turunan dan integral.

Limit merupakan salah satu pokok bahasan dalam mata kuliah kalkulus yang memegang peranan penting sebagai pembuka untuk beberapa pokok bahasan lainnya. Konsep batas fungsi merupakan konsep dasar untuk membangun beberapa konsep kalkulus lainnya, misalnya konsep turunan dan integral. Sehingga batas penguasaan mendukung tingkat penguasaan pada kalkulus. Secara pemahaman konsep batas sukar untuk dipahami, tetapi pada realisasi konsep batas sering digunakan dalam kehidupan sehari-hari. Misalnya kita tidak dapat menentukan besar perubahan volume udara yang tertua ke dalam gelas tepat pada waktu tertentu. Namun, dengan penerapan konsep batasan, kita dapat menentukannya melalui nilai pendekatannya.

Dalam tulisan ini, paket yang berbeda disajikan. Semua kode untuk paket ini ditulis dalam bahasa pemrograman Python 11. Python dipilih sebagai bahasa implementasi karena kemudahan penggunaannya, banyaknya dukungan yang tersedia, dan paket NumPy, yang menggunakan algoritma BLAS level 3 untuk linier numerik aljabar. Paket diferensiasi berisi tiga algoritma untuk perhitungan numerik masing-masing metode beda hingga mundur Grünwald-Letnikov (GL), metode beda terpusat GL yang ditingkatkan, dan metode kuadratur Riemann-Liouville (RL). Bagian 2 memperkenalkan setiap algoritma dan kompleksitas komputasinya. Pada bagian 3, fungsi inti dan

fungsi tambahan dari paket yang berbeda dijelaskan, dengan beberapa contoh cara kerja paket yang berbeda.

## 2. METODE PENELITIAN

Dalam riset ini, kami menerapkan metode studi literature review dan analisis perhitungan mandiri guna mendalami pemahaman mengenai perhitungan materi kalkulus limit menggunakan pemrograman Python. Proses awal terdiri dari mengeksplorasi literature yang terkait, dimana kami membaca dan mengumpulkan informasi tentang konsep dasar perhitungan turunan, algoritma numerik, dan juga pendekatan simbolik yang diterapkan dalam matematika komputasional. Selanjutnya, kami melakukan analisis kritis terhadap sumber-sumber tersebut untuk mengidentifikasi kelebihan, kelemahan, serta tren terkini dalam pemanfaatan Python dalam perhitungan limit. Dengan menggabungkan studi literatur dan analisis, upaya kami bertujuan memahami sumbangan unik Python dalam konteks perhitungan matematika, dengan harapan memberikan wawasan baru serta kontribusi positif bagi pengembangan metode perhitungan turunan.

## 3. ANALISA DAN PEMBAHASAN

### 3.1 Keunggulan Python dalam Perhitungan Limit

Keunggulan Python dalam perhitungan limit menjadi sorotan utama dalam konteks pengembangan perangkat lunak matematika. Python membedakan dirinya dengan sintaks yang bersifat ekspresif dan mudah dibaca. VanderPlas (2016) menekankan bahwa keterbacaan kode Python memfasilitasi pengembangan algoritma perhitungan limit dengan jelas dan efisien, meningkatkan kolaborasi antara pemrogram dan ahli matematika. Keunggulan ini mempercepat proses pengembangan solusi matematika, terutama dalam konteks aplikasi yang melibatkan analisis numerik.

Python juga terkenal dengan dukungan komunitas yang luas, yang menciptakan ekosistem dinamis untuk pengembangan matematika komputasional. McKinney (2017) mencatat bahwa keunggulan Python terletak pada kemampuannya untuk menggabungkan kekuatan pemrograman dengan analisis data. Dengan perkembangan pustaka dan modul yang terus bertambah, Python menjadi platform yang sangat fleksibel dan relevan dalam berbagai konteks penghitungan matematika.

Pentingnya Python dalam perhitungan limit juga terletak pada penggunaan pustaka khusus seperti NumPy dan SymPy. Harris et al. (2020) menjelaskan bahwa NumPy memberikan dukungan efisien untuk perhitungan numerik dengan vektorisasi, yang mengoptimalkan kinerja perhitungan numerik pada array data. SymPy, menurut Meurer et al. (2017), melengkapi Python dengan alat perhitungan simbolik, memungkinkan representasi dan manipulasi ekspresi matematika secara simbolik. Integrasi antara NumPy dan SymPy membuat Python menjadi platform yang handal dan serbaguna dalam menangani perhitungan turunan, baik numerik maupun simbolik.

Dalam konteks pengembangan perangkat lunak, keterbacaan dan kemudahan sintaks Python memberikan keunggulan tambahan. McKinney (2017) menekankan bahwa Python dapat dengan mudah diintegrasikan dengan analisis data, visualisasi, dan pemodelan matematika, menjadikannya pilihan yang optimal dalam proyek-proyek yang melibatkan perhitungan turunan. Keunggulan ini diperkuat oleh fakta bahwa Python dapat diakses oleh berbagai kalangan, dari pemula hingga ahli. membuka pintu untuk kolaborasi yang lebih luas dalam pengembangan solusi matematika.

Dengan penggunaan yang meluas di berbagai industri dan disiplin ilmu, Python telah menjadi bahasa pemrograman yang sangat relevan dalam pengembangan perangkat lunak matematika. VanderPlas (2016) mencatat bahwa Python memiliki ekosistem yang dinamis dan terus berkembang, dengan dukungan komunitas yang aktif. Keunggulan Python dalam perhitungan turunan menciptakan fondasi yang kokoh untuk inovasi dan perkembangan lebih lanjut dalam bidang matematika komputasional. Pustaka Python seperti NumPy dan SymPy memainkan peran penting dalam memperluas

### 3.2 Keunggulan Python dalam Perhitungan Limit

Kemampuan perhitungan matematika komputasional. NumPy, yang merupakan pustaka dasar dalam ekosistem Python, memberikan dukungan efisien untuk operasi numerik dengan vektorisasi. Harris et al. (2020) menyoroti bahwa NumPy memungkinkan manipulasi array numerik dengan kecepatan tinggi, memberikan keunggulan dalam perhitungan numerik kompleks, seperti yang sering ditemui dalam analisis data, ilmu komputer, dan pengolahan sinyal.

SymPy, di sisi lain, membuka pintu bagi perhitungan simbolik yang diperlukan dalam analisis matematika tingkat tinggi. Pustaka ini, seperti yang dijelaskan oleh Meurer et al. (2017), memungkinkan representasi dan manipulasi ekspresi matematika dalam bentuk simbolik. Kelebihan SymPy menjadi krusial dalam konteks pemodelan matematika yang kompleks, di mana pemahaman aspek simbolik sangat diperlukan.

Kombinasi NumPy dan SymPy menjadikan Python pilihan yang kuat untuk berbagai aplikasi matematika komputasional. Keunggulan integrasi antara keduanya, seperti yang dijelaskan oleh Harris et al. (2020) dan Meurer et al. (2017), memungkinkan pengguna untuk memanfaatkan keuntungan dari perhitungan numerik dan simbolik secara bersamaan. Fleksibilitas Python dalam memanipulasi data numerik dan simbolik dengan mudah memberikan keunggulan tambahan dalam pengembangan dan penelitian ilmiah.

Penggunaan pustaka Python ini tidak hanya memperkaya kemampuan bahasa ini dalam perhitungan matematika, tetapi juga meningkatkan efisiensi dalam penanganan perhitungan kompleks. Pustaka NumPy, dengan kemampuan vektorisasinya, mempercepat proses perhitungan numerik pada data dalam skala besar. Sementara itu, SymPy memberikan cara yang efektif untuk melakukan perhitungan simbolik, memanfaatkan representasi matematika dalam bentuk simbolik.

### 3.3 Demonstrasi Perhitungan Limit dengan Python

Penghitungan turunan dengan Python dapat dijelaskan melalui demonstrasi praktis dengan menggunakan contoh kode. Pada program Python kali ini, kami memanfaatkan pustaka data NumPy dan SymPy yang ada di Python untuk melakukan perhitungan Limit serta kami akan menggunakan simbol X dan Y sebagai deklarasinya. Sebagai contoh pengerjaan, kita akan melakukan percobaan

penghitungan Limit menggunakan NumPy pada soal Limit yaitu :  $\lim_{x \rightarrow 3} \frac{x^2 + x - 6}{x^2 + 5x + 6}$ .

```

Demonstrasi Limit via Python.ipynb
Demonstrasi Limit via Python.ipynb > smp.limit((x**2+x-6)/(x**2+5*x+6), x,3, dir='+-')
+ Code + Markdown | ▶ Run All ⌂ Restart ☰ Clear All Outputs | 📄 Variables ☰ Outline ...

import numpy as np
import sympy as smp
from sympy import *

[1] ✓ 1.4s

x,y = symbols('x,y')

[2] ✓ 0.0s

smp.diff

[3] ✓ 0.0s
... <function sympy.core.function.diff(f, *symbols, **kwargs)>
    
```

Gambar 1. Source Code Python NumPy

```

smp.Limit((x**2+x-6)/(x**2+5*x+6), x,3, dir='+-')
[4] ✓ 0.1s
...
lim_{x \to 3} \left( \frac{x^2 + x - 6}{x^2 + 5x + 6} \right)

smp.limit((x**2+x-6)/(x**2+5*x+6), x,3, dir='+-')
[5] ✓ 0.0s
...
1/5
    
```

Gambar 2. Source Code Python NumPy

Source Code di atas adalah program Python yang menggunakan pustaka NumPy dan SymPy untuk melakukan perhitungan limit dan turunan. Langkah pertama dari program ini adalah mengimpor pustaka yang diperlukan yaitu NumPy dan SymPy dengan menggunakan perintah “import”. NumPy biasanya digunakan untuk operasi numerik dan perhitungan array, sementara SymPy digunakan untuk melakukan komputasi simbolik, seperti perhitungan matematika dengan menggunakan simbol-simbol aljabar. Kemudian, program mendefinisikan simbol-simbol ‘x’ dan ‘y’ sebagai simbol-simbol matematika menggunakan fungsi “symbols” dari pustaka SymPy.

Berikutnya, program melakukan dua perhitungan limit dari fungsi  $\lim_{x \rightarrow 3} \frac{x^2 + x - 6}{x^2 + 5x + 6}$  dengan arah mendekati 3 dari kedua sisi yaitu dari positif (‘dir=+’) dan negatif (‘dir=-’). Perhitungan limit ini dilakukan menggunakan fungsi ‘limit’ dari pustaka SymPy. Limit adalah konsep dalam kalkulus yang mewakili nilai yang dihampiri oleh suatu fungsi saat variabelnya mendekati nilai tertentu. Dalam kasus ini, fungsi yang diberikan  $\lim_{x \rightarrow 3} \frac{x^2 + x - 6}{x^2 + 5x + 6}$

Namun, perlu diperhatikan bahwa pada baris source code, ada pemanggilan ‘smp.diff’ yang seharusnya diikuti oleh ekspresi matematika yang ingin diturunkan terhadap variabel tertentu, tetapi dalam source code kami, tidak ada ekspresi yang diturunkan. Jadi, potongan kode ‘smp.diff’ diikuti oleh ekspresi yang ingin diturunkan dan variabel yang ingin diturunkan, misalnya ‘smp.diff(x\*\*2 + x - 6, x)’ untuk menghitung turunan dari fungsi  $x^2 + x - 6$  terhadap variabel x.

### 3.3 Demonstrasi Perhitungan Limit dengan Python

Untuk memastikan keakuratan pada program Python yang menggunakan pustaka NumPy dan SymPy untuk melakukan perhitungan limit fungsi. Kami menggunakan soal berbeda untuk memastikan apakah program Python tersebut bisa mengerjakannya.

```

Demonstrasi Limit via Python.ipynb
Demonstrasi Limit via Python.ipynb > import numpy as np
+ Code + Markdown | ▶ Run All ☰ Clear All Outputs | ☰ Outline ...

smp.Limit((x**2+3*x-18)/(x**2-3*x), x,3, dir='+-')
[6]
...

$$\lim_{x \rightarrow 3} \left( \frac{x^2 + 3x - 18}{x^2 - 3x} \right)$$

smp.limit((x**2+3*x-18)/(x**2-3*x), x,3, dir='+-')
[7]
... 3
    
```

Gambar 3. Source Code Python NumPy

```

Demonstrasi Limit via Python.ipynb
Demonstrasi Limit via Python.ipynb > import numpy as np
+ Code + Markdown | ▶ Run All ☰ Clear All Outputs | ☰ Outline ...

import numpy as np
import sympy as smp
from sympy import *
[1]

x,y = symbols('x,y')
[4]

smp.diff
[5]
... <function sympy.core.function.diff(f, *symbols, **kwargs)>
    
```

Gambar 4. Source Code Python NumPy

Pada percobaan kedua yaitu program yang diberikan sama dengan sebelumnya menggunakan pustaka NumPy dan SymPy untuk melakukan perhitungan diferensial dan perhitungan limit dari sebuah fungsi. Di bagian awal program, pustaka NumPy dan SymPy diimpor menggunakan sintaks `import numpy as np` dan `import sympy as smp`. Kemudian, `symbols` dari sympy digunakan untuk mendefinisikan simbol-simbol variabel 'x' dan 'y'.

Kemudian, perhitungan diferensial dengan menggunakan `smp.diff` dari sympy adalah sintaks yang mungkin dimaksud untuk menghitung turunan dari suatu fungsi. Namun, dalam program ini, tidak ada fungsi yang dihitung turunannya.

Selanjutnya, terdapat penggunaan fungsi limit. Pada bagian berikut:  $\lim_{x \rightarrow 3} \frac{x^2 + 3x - 18}{x^2 + 3x}$

Ini berarti program ingin menghitung limit dari fungsi 'x' mendekati 3 dari kedua arah, yaitu dari positif (+) dan negatif (-). Kemudian, terdapat sintaks yang merupakan pemanggilan fungsi `limit` dari sympy untuk menghitung limit dari fungsi yang sama saat 'x' mendekati 3 dari kedua arah, positif dan negatif.

Penggunaan bahasa pemrograman Python membawa konsistensi pada alur kerja dengan memungkinkan peneliti dan praktisi untuk menggabungkan perhitungan numerik dan simbolik dalam satu platform dengan mudah. Sebaliknya, menggunakan metode tradisional dapat mengharuskan transisi antara alat atau bahasa pemrograman yang berbeda, yang dapat menyulitkan integrasi dan melambatkan proses analisis matematika. Python juga mendukung otomatisasi perhitungan dan penanganan data yang besar, menjadikannya lebih scalable daripada metode tradisional yang mungkin kurang efisien dalam menangani kompleksitas perhitungan matematika.

Perbandingan ini juga mengundang diskusi terkait kecepatan eksekusi dan efisiensi dalam pemanfaatan sumber daya komputasi. Python, terutama dengan pustaka khususnya, mampu memberikan kinerja yang memadai untuk sebagian besar aplikasi perhitungan turunan. Namun, dalam konteks kebutuhan dan tingkat kompleksitas perhitungan yang berbeda, metode tradisional atau alat lain mungkin lebih sesuai untuk situasi tertentu.

## 4. KESIMPULAN

Secara keseluruhan dari pembahasan diatas, keunggulan Python dalam perhitungan materi kalkulus yaitu limit terdapat pada kemudahan membaca sintaksnya yang ekspresif, menyediakan kejelasan dan pemahaman yang lebih baik terhadap algoritma perhitungan. Dukungan yang ada pada program Python dari pustaka khusus seperti NumPy dan SymPy meningkatkan kemampuan Python dalam menangani perhitungan numerik dan simbolik secara efisien. NumPy memberikan kinerja numerik yang optimal melalui vektorisasi, sementara SymPy memfasilitasi perhitungan simbolik dengan merepresentasikan dan memanipulasi ekspresi matematika secara simbolik.

Integrasi yang sinergis antara NumPy dan SymPy menciptakan lingkungan kerja yang konsisten dan efisien, memungkinkan penanganan perhitungan turunan numerik dan simbolik dalam satu platform. Keunggulan Python tidak hanya terletak pada kapabilitas teknisnya, tetapi juga didukung oleh komunitas yang besar dan aktif, yang terus mengembangkan beragam alat dan solusi terbaik dalam perhitungan matematika komputasional.

Kesimpulannya, Python tidak hanya menjadi bahasa pemrograman yang populer dan mudah dipelajari, tetapi juga merupakan pilihan yang kuat dan serbaguna dalam menangani tugas matematika yang melibatkan perhitungan limit. Dengan kombinasi kejelasan sintaks, dukungan pustaka khusus, dan partisipasi aktif dari komunitas, Python membuka peluang untuk efisiensi dan inovasi dalam ranah perhitungan matematika komputasional.

## REFERENCES

- VanderPlas, Jake. Python data science handbook: Essential tools for working with data. " O'Reilly Media, Inc.", 2016.
- Harris, Charles R., et al. "Array programming with NumPy." Nature 585.7825 (2020): 357-362.
- Ascher, David, et al. "Numerical python." (2001).Meurer, Aaron, et al. "SymPy: symbolic computing in Python." PeerJ Computer Science 3 (2017): e103.
- Id, Ibnu Daqil. Machine Learning: Teori, Studi Kasus dan Implementasi Menggunakan Python. Vol. 1. Unri Press, 2021.
- Hua, N., and B. A. Leu. "Exploring the Power of SymPy: A Comprehensive Study on Symbolic Mathematics in Python."