

## MENENTUKAN SUPREMUM DAN INFIMUM HIMPUNAN DENGAN PYTHON

Dita Lovianna Purba<sup>1</sup>, Dwita Lestari Saragih<sup>2</sup>, Eka Christy Haloho<sup>3</sup>, Rifka Elianti Br  
Kaban<sup>4</sup>, Tri Andri Hutapea<sup>5</sup>

Program Studi Pendidikan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Negeri Medan, Indonesia<sup>1,2,3,4,5</sup>  
e-mail: [saragihdwita50@gmail.com](mailto:saragihdwita50@gmail.com)

### ABSTRAK

Analisis real merupakan cabang matematika yang membahas bilangan real dan fungsi terkait, di mana supremum (batas atas terkecil) dan infimum (batas bawah terbesar) menjadi konsep fundamental. Meskipun esensial dalam memahami limit, integral, dan deret, konsep ini kerap dianggap abstrak dan sulit dipahami mahasiswa. Penelitian ini bertujuan untuk menyederhanakan pemahaman konsep supremum dan infimum melalui pendekatan komputasi dengan bahasa pemrograman Python. Dengan metode kuantitatif, dikembangkan algoritma untuk menentukan supremum dan infimum dari berbagai jenis himpunan, termasuk himpunan diskrit, interval, dan ketidaksetaraan. Program memanfaatkan pustaka regex untuk parsing input dan struktur data set() untuk menghindari duplikasi. Hasil menunjukkan bahwa program mampu mengidentifikasi supremum dan infimum secara akurat, bahkan untuk kasus khusus seperti himpunan kosong atau batas tak hingga. Namun, masih terdapat keterbatasan dalam menangani ekspresi matematika kompleks dan input dengan batas tak hingga. Penelitian ini menunjukkan bahwa integrasi konsep matematika dengan teknologi komputasi dapat menjadi alternatif pembelajaran yang inovatif dan aplikatif dalam analisis real.

**Kata Kunci:** *Supremum, Infimum, Analisis Real, Python, Komputasi Matematika*

### ABSTRACT

Real analysis is a branch of mathematics that deals with real numbers and related functions, where supremum (smallest upper bound) and infimum (largest lower bound) are fundamental concepts. Although essential in understanding limits, integrals, and series, these concepts are often considered abstract and difficult for students to understand. This research aims to simplify the understanding of the supremum and infimum concepts through a computational approach with the Python programming language. Using quantitative methods, algorithms are developed to determine the supremum and infimum of various types of sets, including discrete sets, intervals, and inequalities. The program utilizes the regex library for input parsing and the set() data structure to avoid duplication. Results show that the program is able to identify the supremum and infimum accurately, even for special cases such as the empty set or infinite limits. However, there are still limitations in handling complex mathematical expressions and inputs with infinite limits. This research shows that the integration of mathematical concepts with computational technology can be an innovative and applicable learning alternative in real analysis.

**Keywords:** *Supremum, Infimum, Real Analysis, Python, Mathematical Computation*

### PENDAHULUAN

Analisis real merupakan cabang matematika yang berfokus pada studi bilangan real dan fungsi-fungsi terkait. Konsep-konsep seperti limit, kontinuitas, integral, dan deret konvergen menjadi landasan utama dalam analisis real. Dengan menggunakan pendekatan formal berbasis himpunan dan logika, analisis real memberikan kerangka kerja yang ketat dalam memahami struktur bilangan real serta sifat-sifatnya. Seperti dijelaskan oleh Abbott (2015), pemahaman konsep analisis real membutuhkan pendekatan konseptual yang sistematis agar tidak hanya bersifat mekanistik. Keakuratan dan ketelitian dalam analisis real sangat penting dalam pengembangan teori matematika lanjut serta aplikasinya di berbagai bidang ilmu, termasuk fisika, ekonomi, dan rekayasa (Bartle & Sherbet, 2011).

Salah satu aspek mendasar dalam analisis real adalah konsep supremum dan infimum, yang berkaitan erat dengan sifat keterurutan dalam himpunan bilangan real. Supremum, atau batas atas terkecil, dan infimum, atau batas bawah terbesar, merupakan dua konsep penting yang membantu dalam memahami keberadaan batas suatu himpunan. Pemahaman konsep ini sangat berguna dalam berbagai studi matematika, terutama dalam teori limit dan kekonvergenan. Mashadi (2018) menyatakan bahwa penyampaian konsep supremum dan infimum melalui ilustrasi visual dan contoh numerik dapat meningkatkan pemahaman mahasiswa secara signifikan. Oleh karena itu, sebelum mendalami penerapannya, penting untuk memahami definisi serta sifat dasar dari supremum dan infimum (Rudin, 1976). Dan diperjelas lagi oleh Apostol (1975) bahwa pemahaman mendalam tentang supremum dan infimum sangat penting untuk analisis struktur keterurutan bilangan real.

Konsep supremum dan infimum merupakan dua elemen fundamental dalam analisis real yang berperan penting dalam memahami sifat himpunan bilangan riil. Supremum atau batas atas terkecil dan infimum atau batas bawah terbesar digunakan dalam berbagai kajian matematika, termasuk teori limit, integral, dan deret. Pemahaman yang kuat mengenai kedua konsep ini menjadi landasan bagi mahasiswa maupun peneliti dalam mengembangkan teori yang lebih kompleks dalam analisis real. Oleh karena itu, kajian ini bertujuan untuk memberikan pemahaman mendalam tentang supremum dan infimum, baik secara teoritis maupun aplikatif (Supriadi, 2018).

Perkembangan teknologi komputasi telah membuka peluang baru dalam mempelajari dan mengaplikasikan konsep matematika, termasuk supremum dan infimum. Dengan menggunakan bahasa pemrograman seperti Python, kita dapat menentukan supremum dan infimum suatu himpunan secara efisien dan akurat. Python adalah salah satu bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido van Rossum dan resmi diperkenalkan ke publik pada tahun 1991. Bahasa ini didesain agar mudah dibaca dan ditulis, sehingga para programmer bisa menuangkan ide atau logika pemrograman hanya dalam beberapa baris kode saja. Dibandingkan dengan bahasa seperti C++ atau Java, Python menawarkan cara penulisan yang lebih ringkas. Python juga fleksibel karena mendukung berbagai gaya pemrograman, seperti pemrograman berorientasi objek, prosedural, hingga fungsional (Lutz, 2013).

Dengan tampilan kode yang sederhana dan mudah dipahami, Python menjadi sangat diminati, baik di kalangan akademisi maupun praktisi industri. Bahasa ini juga punya banyak

pustaka atau library bawaan yang bisa langsung digunakan untuk berbagai kebutuhan, mulai dari analisis data, kecerdasan buatan, pengembangan situs web, hingga otomatisasi tugas-tugas komputer. Karena sifatnya yang ramah bagi pemula, Python sering dijadikan sebagai bahasa pertama yang diajarkan di bangku kuliah (Downey, 2015). Penelitian oleh Smith & Johnson (2020) dalam jurnal *Computational Mathematics* menunjukkan bahwa penggunaan Python untuk menentukan supremum dan infimum tidak hanya mempercepat proses komputasi tetapi juga meningkatkan pemahaman konseptual melalui visualisasi data. Tools seperti Matplotlib dapat digunakan untuk menggambarkan himpunan dan batas-batasnya, sehingga memudahkan identifikasi supremum dan infimum. Selain itu, aplikasi Python dalam pembelajaran mesin dan optimasi juga memanfaatkan konsep supremum dan infimum. Misalnya, dalam algoritma clustering, supremum digunakan untuk menentukan batas atas jarak antar-cluster, sementara infimum membantu dalam menemukan solusi optimal pada masalah minimasi (Goodfellow et al., 2016).

Sebagai bahasa pemrograman yang bersifat open-source dan didukung oleh komunitas global, python terus berkembang dan menawarkan fleksibilitas tinggi bagi para peneliti dan pendidik. Menurut Van Rossum dan Drake (2009), desain sintaksis Python yang sederhana dan mudah dipahami menjadikannya bahasa yang ideal untuk mengajarkan konsep matematika kepada pemula, sekaligus cukup kuat untuk analisis tingkat lanjut. Hal ini menjadikan Python tidak hanya sebagai alat bantu teknis, tetapi juga sebagai jembatan antara konsep teoritis dan implementasi praktis dalam dunia pendidikan maupun penelitian.

Di era digital saat ini, pembelajaran matematika tidak lagi terbatas pada pendekatan konvensional, melainkan sudah mulai beralih ke pendekatan berbasis teknologi. Hal ini terutama penting untuk topik-topik abstrak seperti supremum dan infimum, yang sering kali sulit dipahami mahasiswa secara intuitif. Banyak mahasiswa kesulitan membayangkan batas atas dan bawah dalam bentuk numerik dan visual, sehingga penggunaan alat bantu python dapat menjembatani kesenjangan antara teori praktik. Selain itu, Python sebagai bahasa pemrograman yang bersifat open-source dan memiliki komunitas luas, menawarkan kemudahan akses serta fleksibilitas yang tinggi dalam menyusun simulasi matematis. Oleh karena itu, pemanfaatan Python dalam pembelajaran konsep-konsep fundamental analisis real seperti supremum dan infimum menjadi sangat relevan dalam konteks pendidikan masa kini, terlebih untuk mempersiapkan mahasiswa menghadapi tantangan di bidang sains data, kecerdasan buatan, dan teknologi analitik lainnya.

Penelitian ini bertujuan untuk menguraikan konsep supremum dan infimum secara sistematis serta mengeksplorasi aplikasi praktisnya dalam analisis real. Kajian ini tidak hanya membahas definisi dan sifat-sifat dasar supremum dan infimum, tetapi juga memberikan contoh konkret serta pembuktian matematis yang mendukung pemahaman konseptual. Dengan pendekatan yang lebih komprehensif, diharapkan penelitian ini dapat memberikan kontribusi bagi pengembangan analisis real serta menjadi referensi bagi mahasiswa, akademisi, dan peneliti yang mendalami matematika.

## **METODE PENELITIAN**



Penelitian ini menggunakan pendekatan kuantitatif untuk menyelesaikan permasalahan supremum dan infimum dalam suatu himpunan dengan bantuan Python. Pendekatan ini dipilih karena melibatkan perhitungan numerik, pemodelan matematis, serta analisis hasil yang dapat diukur secara objektif. Menurut Hidayah et al. (2024), penelitian kuantitatif dalam konteks supremum dan infimum bertujuan untuk mengidentifikasi pola batas atas dan batas bawah suatu himpunan melalui komputasi yang akurat. Oleh karena itu, dalam penelitian ini, supremum dan infimum dianalisis secara sistematis dengan menerapkan algoritma pemrograman dalam Python guna memperoleh hasil yang lebih efisien dibandingkan metode manual. Menurut Tao (2006), penggunaan pendekatan berbasis himpunan dan logika formal memudahkan pembuktian dan validasi program yang memodelkan konsep matematika abstrak. Tahapan penelitian diawali dengan identifikasi masalah, di mana supremum dan infimum suatu himpunan didefinisikan secara matematis. Himpunan yang digunakan dalam penelitian ini dapat berupa bilangan real, bilangan bulat, atau fungsi tertentu yang memiliki batas atas dan batas bawah. Setelah itu, dilakukan pemodelan matematis untuk merepresentasikan himpunan dalam bentuk yang dapat diproses oleh Python. Pemodelan ini penting untuk mengubah konsep abstrak supremum dan infimum menjadi bentuk algoritmik sehingga dapat diterapkan dalam komputasi secara otomatis dan sistematis.

Setelah pemodelan dilakukan, tahap berikutnya adalah implementasi program menggunakan bahasa pemrograman Python dengan pemanfaatan pustaka `re` untuk pencocokan pola input serta struktur data seperti `set()` untuk menghindari duplikasi elemen dalam himpunan eksplisit. Algoritma yang dikembangkan memungkinkan program untuk menerima input berupa himpunan angka atau fungsi, kemudian menghitung supremum dan infimum secara efisien. Program juga dirancang untuk menampilkan hasil perhitungan dengan cara yang mudah dipahami dan sesuai dengan konsep teoritis supremum dan infimum.

Data yang diperoleh dari hasil uji coba dianalisis secara kualitatif untuk mengidentifikasi pola kesalahan, keterbatasan algoritma, serta potensi perbaikan yang dapat dilakukan pada implementasi program. Evaluasi program juga mencakup aspek efisiensi dalam menangani himpunan besar dan kemampuan program dalam memberikan pesan kesalahan yang jelas bagi pengguna (Van Rossum & Drake, 2009). Dengan demikian, penelitian ini tidak hanya mengevaluasi keakuratan hasil yang diberikan oleh program, tetapi juga menilai keandalan dan kemudahan penggunaan dalam berbagai konteks aplikasi matematika.

## HASIL DAN PEMBAHASAN

### Hasil

Program ini dikembangkan dengan mengintegrasikan konsep matematika *supremum* dan *infimum* dari teori himpunan dan analisis real ke dalam logika pemrograman. Berdasarkan kajian pustaka, supremum didefinisikan sebagai *batas atas terkecil* dari suatu himpunan, sedangkan infimum adalah *batas bawah terbesar*-nya (Bartle & Sherbet, 2011). Implementasi program ini dirancang untuk mereplikasi proses analisis manual tersebut secara komputasional, dengan tiga pendekatan utama: himpunan diskrit, interval, dan ketidaksetaraan.



Pada awal program, didefinisikan beberapa fungsi utama yang berperan dalam memproses input dan menentukan supremum serta infimum. Fungsi `parse_inequality` (`input_str`) bertugas untuk mengenali dan mengekstrak informasi dari input berupa ketidaksetaraan (misalnya,  $x > 3$  atau  $2 \leq x < 7$ ) menggunakan ekspresi reguler (regex). Fungsi ini mengidentifikasi batas bawah dan batas atas serta menentukan apakah batas-batas tersebut inklusif (menggunakan  $\leq$  atau  $\geq$ ) atau eksklusif (menggunakan  $<$  atau  $>$ ). Selanjutnya, fungsi `print_sup_inf` (`lower`, `upper`, `lower_inclusive`, `upper_inclusive`) digunakan untuk menganalisis himpunan yang dihasilkan dan menentukan supremum serta infimumnya.

Dalam teori supremum dan infimum, supremum (least upper bound) adalah batas atas terkecil dari suatu himpunan, sedangkan infimum (greatest lower bound) adalah batas bawah terbesar. Program ini menerapkan konsep tersebut dengan mengevaluasi apakah suatu himpunan memiliki batas atas atau bawah, dan jika ada, menentukan nilai supremum dan infimum berdasarkan sifat inklusif atau eksklusif dari batas-batasnya. Jika himpunan mencakup semua bilangan real, maka supremum dan infimum tidak terdefinisi (tak hingga). Jika himpunan terbatas, supremum adalah nilai terbesar dalam himpunan atau batas atasnya jika inklusif, sedangkan infimum adalah nilai terkecil atau batas bawahnya jika inklusif. Dengan demikian, program ini mengotomatisasi analisis himpunan dalam konteks supremum dan infimum serta memberikan interpretasi matematis yang sesuai.

```
1 import re
2
3 def parse_inequality(input_str):
4     input_str = input_str.strip()
5
6     # Pola regex untuk mencocokkan ketidaksetaraan
7     inequality_pattern = re.compile(r'^\s*\s*([<=>])\s*(-?\d+\.\d*)\s*$')
8     between_pattern = re.compile(r'^\s*(-?\d+\.\d*)\s*([<=>])\s*(-?\d+\.\d*)\s*$')
9
10    # Cek apakah input adalah ketidaksetaraan tunggal (contoh: x > 3)
11    inequality_match = inequality_pattern.match(input_str)
12    if inequality_match:
13        op = inequality_match.group(1)
14        num = float(inequality_match.group(2))
15
16        if op == '>':
17            return (num, float('inf'), False, False)
18        elif op == '>=':
19            return (num, float('inf'), True, False)
20        elif op == '<':
21            return (float('-inf'), num, False, False)
22        elif op == '<=':
23            return (float('-inf'), num, False, True)
24        elif op == '==':
25            return (num, num, True, True)
26
27    # Cek apakah input adalah ketidaksetaraan ganda (contoh: 2 < x < 7)
28    between_match = between_pattern.match(input_str)
29    if between_match:
30        lower_num = float(between_match.group(1))
```

```
31 lower_op = between_match.group(2)
32 upper_op = between_match.group(3)
33 upper_num = float(between_match.group(4))
34
35 # Validasi operasi
36 if (lower_op not in ['<', '<=']) or (upper_op not in ['<', '<=']):
37     return None
38
39 lower_inclusive = (lower_op == '<=')
40 upper_inclusive = (upper_op == '<=')
41
42 return (lower_num, upper_num, lower_inclusive, upper_inclusive)
43
44 return None
45
46 def print_sup_inf(lower, upper, lower_inclusive, upper_inclusive):
47     if lower == float('-inf') and upper == float('inf'):
48         print("Himpunan semua bilangan real.")
49         print("Supremum: tidak ada (tak hingga)")
50         print("Infimum: tidak ada (minus tak hingga)")
51     elif lower == float('-inf'):
52         print(f"Himpunan: x {'<' if upper_inclusive else '<'} {upper}")
53         print(f"Supremum: {upper}")
54         print("Infimum: tidak ada (minus tak hingga)")
55     elif upper == float('inf'):
56         print(f"Himpunan: x {'>' if lower_inclusive else '>'} {lower}")
57         print("Supremum: tidak ada (tak hingga)")
58         print(f"Infimum: {lower}")
59     else:
60         if lower == upper:
61             if lower_inclusive and upper_inclusive:
62                 print(f"Himpunan: {{ {lower} }}")
63                 print(f"Supremum: {lower}")
64                 print(f"Infimum: {lower}")
65             else:
66                 print("Himpunan kosong")
67                 print("Supremum: tidak ada")
68                 print("Infimum: tidak ada")
69         else:
70             print(f"Himpunan: {lower} {'<' if lower_inclusive else '<'} x {'<' if upper_inclusive else '<'} {upper}")
71             print(f"Supremum: {upper}")
72             print(f"Infimum: {lower}")
73
74 input_str = input("Masukkan himpunan (contoh: x>3, x<=5, 2<x<7, atau {1,5}): ").strip()
75
76 # Cek apakah input adalah ketidaksamaan
77 inequality = parse_inequality(input_str)
78 if inequality is not None:
79     lower, upper, lower_inclusive, upper_inclusive = inequality
80     print_sup_inf(lower, upper, lower_inclusive, upper_inclusive)
81 else:
82     # Proses sebagai interval atau daftar angka
83     if (input_str.startswith '[' or input_str.startswith '(') and (input_str.endswith ']' or input_str.endswith ')'):
84         # Proses sebagai interval
85         lower_inclusive = input_str.startswith '[')
86         upper_inclusive = input_str.endswith ']'
87         content = input_str[1:-1].strip()
88         parts = content.split(',')
89
```

```
90     if len(parts) != 2:
91         print("Format interval tidak valid. Harus memiliki dua bagian, dipisahkan koma.")
92         exit()
93
94     lower_str, upper_str = parts[0].strip(), parts[1].strip()
95
96     # Parsing batas bawah
97     try:
98         if lower_str.lower() == 'inf':
99             lower = float('inf')
100        elif lower_str.lower() == '-inf':
101            lower = float('-inf')
102        else:
103            lower = float(lower_str)
104    except ValueError:
105        print("Batas bawah tidak valid:", lower_str)
106        exit()
107
108    # Parsing batas atas
109    try:
110        if upper_str.lower() == 'inf':
111            upper = float('inf')
112        elif upper_str.lower() == '-inf':
113            upper = float('-inf')
114        else:
115            upper = float(upper_str)
116    except ValueError:
117        print("Batas atas tidak valid:", upper_str)
118        exit()
```

```
120 # Cek validitas interval
121 if lower > upper:
122     print("Interval kosong karena batas bawah lebih besar dari batas atas.")
123     print("Supremum dan infimum tidak ada.")
124 elif lower == upper:
125     if lower_inclusive and upper_inclusive:
126         print("Supremum adalah", upper)
127         print("Infimum adalah", lower)
128     else:
129         print("Interval kosong karena batas sama tetapi tidak inklusif di kedua sisi.")
130         print("Supremum dan infimum tidak ada.")
131 else:
132     print_sup_inf(lower, upper, lower_inclusive, upper_inclusive)
133 else:
134     # Proses sebagai daftar angka
135     elements = input_str.split(',')
136     numbers = []
137
138     for e in elements:
139         stripped = e.strip()
140         if stripped:
141             try:
142                 num = float(stripped)
143                 numbers.append(num)
144             except ValueError:
145                 print("Elemen tidak valid:", stripped)
146                 exit()
147
148     if not numbers:
149         print("Himpunan kosong. Supremum dan infimum tidak ada.")
150     else:
151         unique_numbers = list(set(numbers))
152         sup = max(unique_numbers)
153         inf = min(unique_numbers)
154         print("Supremum adalah", sup)
155         print("Infimum adalah", inf)
```

Gambar 1. Program supremum dan infimum

Untuk menguji keakuratan program dalam menentukan supremum dan infimum, dilakukan beberapa uji coba dengan berbagai jenis input. Berikut adalah beberapa hasil yang diperoleh:

1. Input Ketidaksetaraan ( $x > 3$ )

```
Masukkan himpunan (contoh:  $x > 3$ ,  $x \leq 5$ ,  $2 < x < 7$ , atau  $[1, 5)$ ):
x > 3
Himpunan: x > 3.0
Supremum: tidak ada (tak hingga)
Infimum: 3.0
```

Gambar 2. Hasil eksekusi program  $x > 3$

Program mengenali batas bawah 3 sebagai infimum (karena  $x$  lebih besar dari 3) dan tidak memiliki batas atas, sehingga supremum tidak terdefinisi.

2.  $2 < x < 7$

```
Masukkan himpunan (contoh:  $x > 3$ ,  $x \leq 5$ ,  $2 < x < 7$ , atau  $[1, 5)$ ):  
2 < x < 7  
Himpunan: 2.0 < x < 7.0  
Supremum: 7.0  
Infimum: 2.0
```

Gambar 3. Hasil eksekusi program  $2 < x < 7$

3. Input Interval  $[1, 5)$

```
Masukkan himpunan (contoh:  $x > 3$ ,  $x \leq 5$ ,  $2 < x < 7$ , atau  $[1, 5)$ ):  
[1, 5)  
Himpunan: 1.0 <= x < 5.0  
Supremum: 5.0  
Infimum: 1.0
```

Gambar 4. Hasil eksekusi program  $[1, 5)$

Interval  $[1, 5)$  berarti himpunan mencakup angka mulai dari 1 (inklusif) hingga mendekati 5 (eksklusif). Maka, infimum adalah 1, dan supremum adalah 5, tetapi tidak termasuk dalam himpunan.

4.  $(3, 3)$

```
Masukkan himpunan (contoh:  $x > 3$ ,  $x \leq 5$ ,  $2 < x < 7$ , atau  $[1, 5)$ ):  
(3, 3)  
Interval kosong karena batas sama tetapi tidak inklusif di kedua sisi.  
Supremum dan infimum tidak ada.
```

Gambar 5. Hasil eksekusi program  $(3, 3)$

Karena batas bawah dan atas sama tetapi tidak inklusif ( $()$ ), maka tidak ada angka yang termasuk dalam himpunan, sehingga supremum dan infimum tidak ada.

### Pembahasan

Hasil eksekusi menunjukkan bahwa program bekerja dengan baik dalam mengenali berbagai format input dan menentukan supremum serta infimum dengan benar. Program ini dirancang untuk menangani berbagai bentuk himpunan yang sering muncul dalam analisis matematika real, baik yang berbentuk ekspresi ketidaksetaraan, interval, maupun daftar angka eksplisit.

Salah satu keunggulan utama dari program ini adalah kemampuannya dalam menginterpretasikan berbagai format input dengan akurat. Misalnya, input berupa

ketidaksetaraan seperti  $x > 3$  dapat langsung dikenali sebagai himpunan bilangan yang lebih besar dari 3, dengan infimum 3 dan supremum tidak terdefinisi (tak hingga). Begitu pula dengan bentuk interval, seperti  $[1,5)$ , yang dianalisis dengan mempertimbangkan sifat inklusif atau eksklusif dari batas-batasnya. Program juga mampu mengenali himpunan eksplisit berupa daftar angka, menghilangkan duplikasi, serta menentukan supremum dan infimum dengan pendekatan berbasis komputasi numerik.

Selain menangani kasus standar, program ini juga menunjukkan fleksibilitas dalam menghadapi skenario khusus, pada kasus  $(3,3)$ , program mengenali bahwa himpunan tersebut kosong karena kedua batas tidak inklusif dan nilainya sama, sehingga supremum dan infimum tidak terdefinisi. Kemampuan program dalam menangani situasi semacam ini menunjukkan bahwa implementasi logikanya sudah cukup matang untuk berbagai kemungkinan input pengguna.

Program ini juga memiliki keunggulan dalam membedakan antara supremum yang benar-benar termasuk dalam himpunan dengan supremum yang hanya merupakan batas atas dari himpunan. Sebagai contoh, dalam himpunan  $[2, 7)$ , supremum adalah 7, tetapi angka tersebut tidak termasuk dalam himpunan karena batasnya eksklusif. Hal ini menunjukkan bahwa program tidak hanya mencari nilai maksimum atau minimum dalam daftar angka, tetapi juga mempertimbangkan karakteristik matematis dari batasan yang diberikan dalam input.

Keakuratan dan efisiensi program ini menjadikannya alat yang bermanfaat dalam studi matematika real, terutama dalam memahami konsep supremum dan infimum secara komputasional. Sejalan dengan hal tersebut, Pugh (2002) menjelaskan bahwa konsep supremum dan infimum tidak hanya bersifat teoritis, tetapi juga menjadi dasar penting dalam penerapan analisis real untuk menjelaskan keberadaan batas dan sifat kekonvergenan dalam himpunan bilangan. Program ini dapat digunakan sebagai alat bantu dalam pendidikan matematika, penelitian, maupun aplikasi praktis lainnya yang memerlukan analisis batas atas dan bawah suatu himpunan bilangan hal ini sejalan dengan hasil penelitian yang dilakukan oleh Hidayah, dkk (2024) menunjukkan bahwa Python merupakan alat yang efektif dan efisien dalam menyelesaikan permasalahan supremum dan infimum suatu himpunan. Dengan pendekatan kuantitatif, peneliti membangun algoritma dalam Python untuk mengidentifikasi sifat batas atas dan bawah dari himpunan, serta menghitung nilai supremum dan infimumnya secara otomatis. Selain itu, dengan pengembangan lebih lanjut, program ini dapat diperluas untuk menangani ekspresi yang lebih kompleks, seperti ketidaksetaraan yang melibatkan variabel dalam bentuk polinomial atau fungsi non-linear, sehingga cakupan penerapannya semakin luas. Dalam konteks optimal, Luenberger (1997) menunjukkan bahwa supremum dan infimum juga penting untuk menentukan batas Solusi optimal dalam ruang vector.

Dengan demikian, program ini tidak hanya sekadar menentukan supremum dan infimum dari himpunan bilangan, tetapi juga memberikan wawasan yang lebih dalam mengenai sifat dan struktur himpunan tersebut. Hal ini menunjukkan potensi penerapan algoritma dalam memahami konsep matematika secara lebih interaktif dan aplikatif.

Meskipun program ini telah mampu menentukan supremum dan infimum dengan baik dalam berbagai kasus standar, terdapat beberapa kelemahan yang perlu diperhatikan dalam

implementasinya. Salah satu kelemahan utama adalah ketidaksesuaian dalam menangani input yang melibatkan batas tak hingga, seperti  $(-\infty, 4]$ . Hal ini kemungkinan disebabkan oleh cara program memproses simbol  $\infty$  atau  $-\infty$ , baik dalam tahap parsing maupun validasi input. Secara khusus, jika program mengandalkan fungsi `float()` untuk mengubah input string menjadi angka, maka perlu ada perlakuan khusus agar  $-\infty$  dapat dikenali sebagai `float('-inf')` dan  $\infty$  sebagai `float('inf')`. Namun, jika program tidak menangani kasus ini dengan eksplisit, input yang melibatkan batas tak hingga akan dianggap tidak valid.

Selain itu, kelemahan lain dari program ini adalah keterbatasannya dalam menangani ekspresi ketidaksetaraan yang lebih kompleks, seperti  $x^2 < 4$  atau  $x + y > 5$ , yang memerlukan pendekatan pemrosesan simbolik untuk menyelesaikan. Saat ini, program hanya mampu menangani bentuk dasar ketidaksetaraan dan interval satu variabel dalam format yang sangat terbatas. Oleh karena itu, jika input yang diberikan berupa ekspresi matematika yang lebih kompleks, program akan gagal dalam menginterpretasi dan memberikan hasil yang benar.

Kelemahan lainnya adalah kurangnya fitur penanganan kesalahan yang lebih informatif bagi pengguna. Jika pengguna memasukkan input yang tidak sesuai format, program saat ini hanya menampilkan pesan error umum tanpa menjelaskan dengan detail kesalahan apa yang terjadi. Hal ini dapat menyulitkan pengguna dalam memahami format yang benar. Misalnya, jika seseorang memasukkan interval dengan format salah seperti  $(3, 3]$ , program seharusnya memberikan informasi lebih spesifik bahwa interval tersebut sebenarnya kosong karena batas bawah lebih besar atau sama dengan batas atas tetapi tidak inklusif di kedua sisi. Dengan adanya validasi dan umpan balik yang lebih jelas, program dapat menjadi lebih ramah pengguna dan mengurangi potensi kesalahan input.

Dari sisi efisiensi, program juga masih dapat dioptimalkan lebih lanjut, terutama dalam menangani himpunan eksplisit yang terdiri dari banyak angka. Saat ini, program menggunakan `set()` untuk menghilangkan duplikasi sebelum mencari supremum dan infimum. Meskipun pendekatan ini cukup efisien untuk kumpulan angka kecil, pada skala yang lebih besar, program mungkin mengalami keterbatasan kinerja jika tidak dioptimalkan lebih lanjut. Sebagai contoh, dalam kasus daftar angka yang sangat besar, algoritma pencarian supremum dan infimum bisa diperbaiki dengan pendekatan yang lebih efisien seperti penggunaan heap atau struktur data yang lebih optimal.

Dengan demikian, meskipun program ini telah memberikan solusi dasar dalam menentukan supremum dan infimum, masih terdapat beberapa kelemahan yang perlu diperbaiki agar lebih fleksibel, efisien, dan mudah digunakan. Dengan pengembangan lebih lanjut, program ini dapat diperluas untuk menangani kasus yang lebih kompleks, memberikan umpan balik yang lebih baik, dan meningkatkan performa dalam pemrosesan data skala besar.

## KESIMPULAN

Penelitian ini berhasil mengimplementasikan konsep supremum dan infimum dalam analisis real menggunakan bahasa pemrograman Python. Program yang dikembangkan mampu menentukan batas atas terkecil dan batas bawah terbesar suatu himpunan dengan akurat untuk berbagai format input, seperti himpunan diskrit, interval, dan ketidaksetaraan sederhana. Hasil



uji coba menunjukkan bahwa program bekerja efektif dalam kasus-kasus standar, termasuk identifikasi himpunan kosong dan batas tak hingga. Namun, terdapat beberapa keterbatasan, seperti ketidakmampuan menangani ekspresi matematika kompleks dan kurangnya fitur penanganan kesalahan yang informatif. Ke depan, pengembangan program dapat difokuskan pada peningkatan fleksibilitas untuk menangani ekspresi non-linear, optimasi kinerja untuk himpunan besar, serta penyempurnaan antarmuka pengguna. Penelitian ini menegaskan potensi integrasi antara matematika dan komputasi dalam mempermudah pemahaman konsep analisis real, sekaligus membuka peluang untuk penelitian lanjutan yang lebih mendalam.

#### DAFTAR PUSTAKA

- Abbott, S. (2015). *Understanding analysis* (2nd ed.). Springer.
- Apostol, T. M. (1974). *Mathematical analysis* (2nd ed.). Addison-Wesley.
- Bartle, R. G., & Sherbert, D. R. (2011). *Introduction to real analysis*. Wiley.
- Downey, A. (2015). *Think Python: How to Think Like a Computer Scientist* (2nd ed.). O'Reilly Media.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Hidayah, N., Permana, R., & Yuliana, D. (2024). Penerapan metode kuantitatif dalam analisis batas himpunan bilangan real. *Jurnal Matematika dan Komputasi Terapan*, 12(1), 33–45.
- Luenberger, D. G. (1997). *Optimization by vector space methods*. Wiley.
- Lutz, M. (2013). *Learning Python* (5th ed.). O'Reilly Media.
- Mashadi, M. (2018). *Analisis real*. Universitas Riau.
- Pugh, C. C. (2002). *Real mathematical analysis*. Springer.
- Rudin, W. (1976). *Principles of mathematical analysis* (3rd ed.). McGraw-Hill.
- Supriadi. (2018). Komposisi bahan ajar konsep analisis real "Supremum dan Infimum". *Sainstek: Jurnal Sains dan Teknologi*, 9(2), 151–157.
- Smith, A., & Johnson, B. (2020). Computational approaches to supremum and infimum in real analysis. *Computational Mathematics*, 15(3), 45–60.
- Tao, T. (2006). *Analysis I* (3rd ed.). Hindustan Book Agency.
- Van Rossum, G., & Drake, F. L. (2009). *The Python language reference manual*. Network Theory Ltd.