Proximal: Jurnal Penelitian Matematika dan Pendidikan Matematika
ISSN 26158132 (cetak)
ISSN 26157667 (online)

MENJELAJAH KONVERGENSI BARISAN LEWAT PYTHON: INTEGRASI ANALISIS MATEMATIS DAN KOMPUTASIONAL

Putri Nazwa[1], Rehana[2], Yarti Melvina Rambe[3], Yesnita Aprilia[4]
Jurusan Matematika, FMIPA[1,2,3,4]
Universitas Negeri Medan[1,2,3,4]
Email: putrinzwaa12@gmail.com[1], rhna1804@gmail.com[2], yartiirambe@gmail.com[3], yesnitaa@gmail.com[4]

Coressponding Author: Putri Nazwa email: putrinzwaa12@gmail.com

Abstrak. Konsep konvergensi barisan merupakan salah satu topik fundamental dalam Analisis Real yang sering kali menimbulkan kendala konseptual dan prosedural bagi pembelajar. Penelitian ini bertujuan untuk mengembangkan serta menganalisis sebuah program berbasis *Python* yang dirancang untuk mengevaluasi konvergensi barisan secara sistematis melalui integrasi pendekatan analitik dan numerik. Metode yang digunakan dalam penelitian ini adalah penelitian pengembangan dengan pendekatan deskriptif-komputasional, yang mencakup perancangan, implementasi, dan pengujian program. Program ini mengadopsi definisi limit, kriteria *Cauchy*, dan estimasi limit berbasis representasi simbolik untuk mengidentifikasi sifat konvergen atau divergen dari suatu barisan. Hasil pengujian menunjukkan bahwa program mampu mengevaluasi berbagai bentuk barisan dengan tingkat akurasi tinggi dan ketahanan terhadap keragaman struktur ekspresi. Keunggulan utama terletak pada efisiensi pemrosesan serta kemampuannya dalam menginterpretasi ekspresi matematis kompleks secara simbolik. Meskipun demikian, keterbatasan masih dijumpai, terutama terkait ketergantungan pada format sintaks yang tepat dan ketiadaan visualisasi grafis dinamis. Simpulan dari penelitian ini menegaskan bahwa program yang dikembangkan berpotensi menjadi alat bantu inovatif dalam mendukung pemahaman konsep konvergensi barisan. Disarankan pengembangan lanjutan melalui integrasi antarmuka visual interaktif dan perluasan kapabilitas komputasi simbolik agar program ini semakin optimal sebagai media pembelajaran Analisis Real.
Kata Kunci: Analisis Real, Konvergensi Barisan, Limit, Python.

Abstract. The concept of convergence of a sequence is one of the fundamental topics in Real Analysis that often poses conceptual and procedural obstacles for learners. This research aims to develop and analyze a Python-based program designed to systematically evaluate the convergence of a sequence through the integration of analytical and numerical approaches. The method used in this research is development research with a descriptive-computational approach, which includes designing, implementing, and testing the program. The program adopts the limit definition, Cauchy criterion, and symbolic representation-based limit estimation to identify the convergent or divergent nature of a line. The test results show that the program is able to evaluate various line shapes with a high degree of accuracy and robustness to expression structure diversity. The main advantages lie in its processing efficiency as well as its ability to interpret complex mathematical expressions symbolically. However, limitations were still encountered, mainly related to the dependence on the exact syntax format and the absence of dynamic graphical visualization. The conclusion of this study confirms that the developed program has the potential to be an innovative tool in supporting the understanding of the concept of convergence of lines. Further development through the integration of interactive visual interfaces and the expansion of symbolic computing capabilities are recommended to optimize the program as a learning tool for Real Analysis.
Keywords: Convergence of Rows, Limits, Python, Real Analysis.

A. Pendahuluan

Matematika memainkan peran fundamental dalam pengembangan ilmu pengetahuan dan teknologi yang tidak hanya terbatas pada perhitungan angka, tetapi mencakup analisis logis dan pemecahan masalah kompleks. Salah satu cabang kritis dalam matematika adalah analisis real, yang fokus pada studi bilangan real dan sifat-sifat fungsinya, menyediakan

CC BY SA

landasan bagi konsep-konsep lanjutan seperti limit, kontinuitas, dan integral yang digunakan dalam berbagai bidang ilmiah seperti fisika, ekonomi, dan teknik (Isnani et al., 2021).

Dalam analisis real, konsep konvergensi barisan memiliki signifikansi teoritis yang mendalam karena menjadi dasar bagi berbagai studi lanjut, seperti teori deret, analisis fungsi, dan pemodelan matematika. Mengacu pada definisi dari Bartle & Sherbert (2011), konvergensi barisan bilangan real $\{a_n\}$ dikatakan konvergen jika barisan bilangan real tersebut memiliki limit tertentu (L) yang akan didekati barisan tertentu ketika nilai n menuju tak hingga. Secara matematis yaitu: $\lim_{n\to\infty} a_n = L$. Definisi ini membutuhkan pemahaman mendalam tentang nilai mutlak dan sifat-sifatnya, termasuk sifat non-negatif ($|x| \geq 0$ untuk semua $x \in \mathbb{R}$), identitas perkalian ($|x| = 0$ jika dan hanya jika $x = 0$), dan pertidaksamaan segitiga ($|a.b| = |a|.|b|$). Secara matematis, suatu barisan bilangan real $\{a_n\}$ dikatakan konvergen ke L jika untuk setiap $\varepsilon > 0$, terdapat $\mathbb{N}$ sehingga $\forall\, n \geq \mathbb{N}$ maka $|a_n - L| < \varepsilon$. Konsep ini memerlukan kemampuan analitis yang tinggi, dengan pengujian dan pembuktian sifat konvergensi melalui berbagai metode seperti uji rasio, uji akar, analisis limit simbolik, serta verifikasi numerik (Angin et al., 2024).

Kemampuan pemecahan masalah adalah keterampilan individu dalam menganalisis informasi, menyusun alternatif solusi, dan memilih penyelesaian yang paling efektif, yang terdiri dari lima tahap kritis, yaitu mengidentifikasi, merumuskan, mengeksplorasi strategi, mengimplementasikan, dan mengevaluasi hasil. Sementara itu, kemampuan teknologi informasi dan komunikasi (TIK) mencakup pemanfaatan TIK sebagai sarana informasi, media pengolahan, basis literasi, peningkatan kreativitas, sarana kolaborasi, dan komunikasi. Kemampuan ini mencakup akses dan evaluasi informasi, pemahaman serta validasi pesan, pemanfaatan teknologi digital, kreativitas dalam menyampaikan ide, kolaborasi melalui aplikasi, serta komunikasi yang cepat dan valid, sehingga mendukung efektivitas dalam memperoleh dan menyebarkan informasi (Hutapea et al., 2020).

Konsep konvergensi barisan merupakan fondasi utama dalam Analisis Real karena menjadi dasar dalam memahami limit, kontinuitas, dan deret tak hingga. Namun, sifatnya yang sangat abstrak menjadikannya tantangan konseptual bagi mahasiswa, terutama karena definisi formalnya melibatkan pendekatan limit, nilai mutlak, dan parameter ε, yang menuntut pemahaman mendalam terhadap sifat-sifat bilangan real serta logika formal yang tinggi. Penelitian oleh Angin et al. (2024), Rahmadani et al. (2025) menunjukkan bahwa banyak mahasiswa mengalami kesulitan dalam memahami konsep konvergensi. Hal ini sering kali mengakibatkan kesalahan dalam pembuktian analitis dan evaluasi numerik. Kompleksitas tersebut mendorong perlunya pendekatan inovatif, seperti pemrograman berbasis visualisasi, guna membantu pemahaman yang lebih intuitif dan aplikatif. Salah satu solusi yang menjanjikan adalah pemanfaatan teknologi pemrograman, khususnya *Python*. Sebagai bahasa pemrograman tingkat tinggi yang dikembangkan oleh *Guido Van Rossum* pada tahun 1991, *Python* telah menjadi alat yang sangat populer dalam berbagai bidang, termasuk matematika (Alfarizi et al., 2023). Penelitian oleh Apriani et al. (2025) menunjukkan bahwa penggunaan Python dalam pendidikan matematika dapat meningkatkan pemahaman konsep-konsep abstrak, termasuk konvergensi barisan, melalui visualisasi dan simulasi yang interaktif.

Python menawarkan kemampuan unik dalam membantu menganalisis dan memvisualisasikan konsep-konsep matematika yang kompleks. Melalui pustaka seperti *NumPy*, *SymPy*, dan Matplotlib, dapat dilakukan simulasi barisan, perhitungan nilai limit secara numerik, dan visualisasi proses konvergensi. Pendekatan ini tidak hanya membantu pemahaman konseptual tetapi juga mengembangkan keterampilan pemrograman yang sangat diperlukan di era digital saat ini (Saharuddin & Prihatmono, 2022).

Program yang dikembangkan dalam penelitian ini dirancang untuk menentukan sifat konvergensi suatu barisan secara algoritmik dengan pendekatan eksplisit, uji rasio, dan evaluasi numerik yang sesuai dengan teori analisis matematika. Jika suatu barisan terbukti konvergen, program ini diharapkan dapat menentukan nilai limitnya dengan akurasi tinggi. Sementara itu, dalam kasus di mana metode analitik tidak memberikan kesimpulan yang jelas, pendekatan numerik dapat digunakan untuk memberikan indikasi kuat terhadap perilaku limit. Dengan pendekatan ini, penelitian berfokus pada pengembangan program berbasis komputasi yang mampu menganalisis konvergensi barisan secara otomatis, menerapkan berbagai metode dalam menentukan limit, serta memvalidasi efektivitas algoritma yang digunakan melalui teori analisis matematika. Selain itu, program ini juga dirancang untuk memberikan pemahaman intuitif kepada pengguna melalui visualisasi pola nilai barisan serta mengeksplorasi keterbatasan metode yang diterapkan, sehingga dapat memberikan wawasan lebih dalam terhadap keakuratan dan efisiensi pendekatan yang digunakan.

Beberapa penelitian terbaru telah mengeksplorasi pemanfaatan teknologi pemrograman, khususnya *Python*, dalam menyelesaikan permasalahan matematika yang kompleks. Penelitian oleh Dasawarsa et al. (2023) membahas penerapan *Python* dalam perhitungan limit fungsi, dengan menggunakan pustaka seperti *NumPy* dan *SciPy* untuk melakukan perhitungan numerik dan simbolik secara efisien dan akurat. Meskipun fokus penelitian ini pada perhitungan limit, pendekatan yang digunakan menunjukkan potensi *Python* dalam menganalisis konsep-konsep matematika yang kompleks. Selain itu, penelitian oleh Saragih et al. (2024) menganalisis perhitungan turunan fungsi dua peubah dan visualisasi grafik 3D menggunakan *Python*. Dengan memanfaatkan pustaka seperti *NumPy*, *SymPy*, dan *Matplotlib*, penelitian ini menunjukkan bagaimana *Python* dapat digunakan untuk melakukan perhitungan matematika dan visualisasi yang kompleks, sehingga memberikan pemahaman yang lebih mendalam terhadap konsep turunan fungsi multivariabel. Meskipun penelitian ini tidak secara langsung membahas konvergensi barisan, pendekatan serupa dapat diterapkan untuk menganalisis dan memvisualisasikan proses konvergensi.

Penelitian-penelitian tersebut mendukung hipotesis bahwa penggunaan teknologi pemrograman seperti *Python* dapat membantu dalam menganalisis dan memahami konsep-konsep matematika yang kompleks, termasuk konvergensi barisan, secara algoritmik sesuai dengan teori analisis matematika. Integrasi teknologi ini tidak hanya meningkatkan pemahaman konseptual tetapi juga mengembangkan keterampilan pemrograman yang esensial di era digital.

B. Metodologi Penelitian

Metode penelitian yang digunakan dalam penelitian ini adalah penelitian pengembangan (*development research*) yang bertujuan untuk merancang dan mengembangkan perangkat lunak berbasis *Python* guna menganalisis konvergensi suatu barisan dalam analisis real. Pendekatan yang digunakan adalah kuantitatif dan komputasional dengan metode eksperimen, yang melibatkan penerapan berbagai teknik analisis matematis dan numerik dalam perangkat lunak yang dikembangkan. Subjek dalam penelitian ini adalah berbagai jenis barisan bilangan real yang dianalisis berdasarkan karakteristik konvergensinya, sedangkan objek penelitian adalah perangkat lunak berbasis *Python* yang dikembangkan untuk menguji konvergensi barisan menggunakan pendekatan numerik dan simbolik. Pengumpulan data dilakukan melalui beberapa tahapan, yaitu kajian literatur, implementasi algoritma, simulasi dan eksperimen, serta validasi dan analisis hasil. Kajian literatur dilakukan dengan mempelajari teori konvergensi barisan yang terdapat dalam buku analisis real, khususnya dari Bartle & Sherbert (2011). Implementasi algoritma dilakukan menggunakan bahasa pemrograman *Python* dengan pustaka seperti *SymPy* untuk pendekatan simbolik dan *NumPy* untuk pendekatan numerik. Selanjutnya, perangkat lunak diuji dengan berbagai jenis barisan yang memiliki sifat konvergen maupun divergen. Evaluasi numerik dilakukan dengan metode Kriteria *Cauchy*, uji rasio, dan uji akar, sedangkan evaluasi simbolik dilakukan dengan

CC BY SA

perhitungan limit menggunakan metode analitik. Validasi dilakukan dengan membandingkan hasil perangkat lunak dengan teori analisis real yang telah teruji dalam literatur.

Teknik analisis data dalam penelitian ini mencakup analisis numerik, analisis simbolik, dan evaluasi kinerja perangkat lunak. Analisis numerik dilakukan dengan mengevaluasi konvergensi barisan berdasarkan selisih antar suku untuk memastikan apakah memenuhi Kriteria *Cauchy*, serta dengan menerapkan uji rasio dan uji akar guna mengidentifikasi sifat konvergensi. Analisis simbolik dilakukan menggunakan pustaka *SymPy* untuk mengevaluasi limit barisan secara analitik, serta membandingkan hasil perhitungan numerik dengan hasil limit simbolik guna validasi. Evaluasi kinerja perangkat lunak dilakukan dengan mengukur tingkat akurasi perangkat lunak dalam menentukan sifat konvergensi barisan dan meninjau efektivitas metode yang diterapkan. Instrumen penelitian yang digunakan dalam penelitian ini meliputi perangkat lunak *Python* yang dikembangkan untuk implementasi algoritma analisis konvergensi, kumpulan barisan uji yang sifat konvergensinya telah diketahui dari literatur, serta dokumentasi hasil analisis yang digunakan untuk membandingkan hasil perangkat lunak dengan teori analisis real. Keabsahan data dalam penelitian ini diuji melalui triangulasi metode, validasi dengan literatur, serta pengujian berulang. Triangulasi metode dilakukan dengan menggunakan pendekatan numerik dan simbolik untuk memastikan hasil yang diperoleh konsisten. Validasi dengan literatur dilakukan dengan membandingkan hasil perangkat lunak dengan contoh-contoh dalam buku analisis real, sementara pengujian berulang dilakukan dengan berbagai jenis barisan untuk memastikan stabilitas hasil.

Pelaksanaan penelitian ini terdiri dari beberapa tahap utama, yaitu perencanaan, pengembangan, pengujian, serta analisis dan pelaporan. Tahap perencanaan melibatkan studi literatur dan pemahaman teori konvergensi barisan, serta perancangan algoritma dan model perangkat lunak. Pada tahap pengembangan, implementasi kode dilakukan menggunakan *Python* dan pustaka pendukung, serta mengintegrasikan metode numerik dan simbolik dalam perangkat lunak. Tahap pengujian dilakukan dengan mensimulasikan berbagai jenis barisan dan mengevaluasi hasilnya, serta membandingkan hasil tersebut dengan teori untuk validasi. Akhirnya, tahap analisis dan pelaporan dilakukan dengan mengevaluasi keakuratan hasil perangkat lunak dan menyusun laporan hasil penelitian.

C. Hasil Penelitian dan Pembahasan

Penelitian ini bertujuan mengembangkan sebuah program berbasis Python untuk memeriksa konvergensi barisan tak hingga dengan menggabungkan pendekatan simbolik dan numerik. Program dirancang untuk mengevaluasi konvergensi berdasarkan rumus eksplisit suku ke-n, serta mengacu pada konsep limit yang dijelaskan dalam *Introduction to Real Analysis* (Bartle & Sherbert, 2011). Melalui pendekatan ini, program tidak hanya menghitung limit secara numerik, tetapi juga memeriksa selisih absolut antar suku barisan guna menguji kecenderungan menuju suatu nilai limit.

Implementasi ini merupakan bagian integral dari upaya pengembangan perangkat lunak sederhana yang dapat digunakan sebagai alat bantu dalam analisis konvergensi, baik dalam konteks pembelajaran maupun eksplorasi mandiri.

Definisi 3.1.3 Suatu barisan $\mathrm{A} = (\mathrm{a_n})$ dikatakan konvergen ke $\mathrm{L} \in \mathbb{R}$, atau L dikatakan limit dari $(\mathrm{a_n})$, jika untuk setiap $\varepsilon > 0$ terdapat bilangan asli $\mathrm{N}(\varepsilon)$ sehingga untuk semua $\mathrm{n} \geq \mathrm{N}(\varepsilon)$, suku-suku $\mathrm{a_n}$ memenuhi $|\mathrm{a_n} - \mathrm{L}| < \varepsilon$. Jika suatu barisan mempunyai limit, maka barisan tersebut konvergen, dan jika tida mempunyai limit, maka barisan tersebut divergen.

Untuk merepresentasikan definisi tersebut secara komputasional, program menghitung nilai limit secara numerik dan memeriksa kedekatan lima suku terakhir barisan terhadap limit yang diperkirakan. Strategi ini memberikan estimasi empiris terhadap konvergensi barisan tanpa menentukan nilai *N* secara eksplisit.

CC BY SA

Sebagai sarana bantu dalam analisis konvergensi barisan, program ini dirancang untuk menerima masukan berupa formula eksplisit suku ke-n dari barisan bilangan real. Program mengintegrasikan berbagai pendekatan simbolik dan numerik yang relevan dalam evaluasi konvergensi. Beberapa metode yang diakomodasi dalam sistem mencakup perhitungan limit simbolik, penerapan uji rasio dan uji akar, serta pendekatan numerik berbasis perhitungan nilai suku-suku barisan. Di samping itu, aspek teoritis seperti kriteria *Cauchy* dan karakteristik osilasi atau pertumbuhan barisan juga menjadi bagian dari pertimbangan analitis dalam program ini. Dengan penggabungan berbagai pendekatan tersebut, sistem ini berfungsi sebagai alat analisis konvergensi yang komprehensif dan otomatis, yang bertujuan menghasilkan simpulan mengenai sifat konvergensi barisan yang dianalisis. Program sebagai berikut:

```
import math
import re
import numpy as np
from sympy import symbols, sympify, limit, oo, N,
lambdify

def nilai_mutlak(x):
    """Menghitung nilai mutlak dari x."""
    return abs(x)
def evaluasi_formula(formula, n):
    """
    Mengevaluasi formula untuk nilai n tertentu
dengan pendekatan yang lebih aman
    dan mendukung lebih banyak fungsi matematika.
    """
    try:
        n_sym = symbols('n')
        expr = sympify(formula)
        func = lambdify(n_sym, expr,
modules=['numpy', 'sympy'])
        return float(func(n))
    except Exception:
        try:
            expr = re.sub(r'\bn\b', f'({n})',
formula)
            safe_dict = {
                'abs': abs, 'sqrt': math.sqrt,
'exp': math.exp,
                'log': math.log, 'log10':
math.log10, 'log2': math.log2,
                'sin': math.sin, 'cos': math.cos,
'tan': math.tan,
                'asin': math.asin, 'acos':
math.acos, 'atan': math.atan,
                'sinh': math.sinh, 'cosh':
math.cosh, 'tanh': math.tanh,
                'pi': math.pi, 'e': math.e, 'pow':
pow,
                'floor': math.floor, 'ceil':
math.ceil,
                'factorial': math.factorial
            }
            return eval(expr, {"_builtins_": {}},
safe_dict)
        except Exception as e:
            try:
                expr = sympify(formula.replace('n',
str(n)))
                return float(expr.evalf())
            except Exception:
                return None

def ratio_test(formula, n_start=10, n_samples=30):
    """
    Melakukan uji rasio untuk menentukan
konvergensi barisan.
    Menghitung lim(n→∞) |a_{n+1}/a_n|
    """
    try:
        n = symbols('n')
        expr = sympify(formula)
        ratio_expr = abs(expr.subs(n, n+1) /
expr.subs(n, n))
        ratio_limit = limit(ratio_expr, n, oo)
        try:
            ratio_value = float(N(ratio_limit))
            if ratio_value < 1:
                return "konvergen", ratio_value
            elif ratio_value > 1:
                return "divergen", ratio_value
            else:
                return "tidak konklusif",
ratio_value
        except:
            pass
    except:
        pass
    ratios = []
    for n in range(n_start, n_start + n_samples):
        an = evaluasi_formula(formula, n)
        an_plus_1 = evaluasi_formula(formula, n +
1)
        if an is None or an_plus_1 is None or an ==
0:
            continue
        ratio = nilai_mutlak(an_plus_1 / an)
        ratios.append(ratio)
    if not ratios:
        return None, None
    last_ratios = ratios[-10:] if len(ratios) >= 10
else ratios
    avg_ratio = sum(last_ratios) / len(last_ratios)
    is_stable = all(nilai_mutlak(r - avg_ratio) <
0.01 for r in last_ratios)
    if is_stable:
        if avg_ratio < 0.99:
            return "konvergen", avg_ratio
        elif avg_ratio > 1.01:
            return "divergen", avg_ratio
    return "tidak konklusif", avg_ratio

def root_test(formula, n_start=10, n_samples=30):
    """
    Melakukan uji akar untuk menentukan konvergensi
barisan.
    Menghitung lim(n→∞) |a_n|^(1/n)
    """
    try:
        n = symbols('n')
        expr = sympify(formula)

        root_expr = abs(expr) ** (1/n)
        root_limit = limit(root_expr, n, oo)
        try:
            root_value = float(N(root_limit))
            if root_value < 1:
                return "konvergen", root_value
            elif root_value > 1:
                return "divergen", root_value
            else:
                return "tidak konklusif",
root_value
        except:
            pass
    except:
        pass
    roots = []
    for n in range(n_start, n_start + n_samples):
        an = evaluasi_formula(formula, n)
        if an is None:
            continue
        root = nilai_mutlak(an) ** (1/n)
        roots.append(root)
    if not roots:
        return None, None
    last_roots = roots[-10:] if len(roots) >= 10
else roots
    avg_root = sum(last_roots) / len(last_roots)
    is_stable = all(nilai_mutlak(r - avg_root) <
0.01 for r in last_roots)
    if is_stable:
        if avg_root < 0.99:
            return "konvergen", avg_root
        elif avg_root > 1.01:
            return "divergen", avg_root
    return "tidak konklusif", avg_root

def hitung_limit_simbolik(formula):
    """
    Mencoba menghitung limit barisan secara
simbolik menggunakan sympy.
    """
    try:
        n = symbols('n')
        expr = sympify(formula)
        limit_value = limit(expr, n, oo)
        try:
            return float(N(limit_value))
        except:
            return None
    except:
        return None

def cek_limit_numerik(suku, threshold=1e-6):
    """
    Memeriksa konvergensi barisan berdasarkan
perilaku numerik.
    """
    if len(suku) < 20:
        return None, False
    last_values = suku[-20:]
    diffs = [nilai_mutlak(last_values[i] -
last_values[i-1]) for i in range(1,
len(last_values))]
    if all(d < threshold for d in diffs[-5:]):
        limit_value = sum(last_values[-5:]) / 5
        return limit_value, True
    increasing = all(last_values[i] >=
last_values[i-1] for i in range(1,
len(last_values)))
    decreasing = all(last_values[i] <=
last_values[i-1] for i in range(1,
len(last_values)))
    if (increasing and last_values[-1] > 1e6) or
(decreasing and last_values[-1] < -1e6):
        return None, False
    return None, None

def uji_cauchy(suku, epsilon=1e-6):
    """
    Menerapkan uji Cauchy untuk menentukan
konvergensi barisan.
    Barisan konvergen jika untuk semua ε>0, ada N
sehingga untuk semua m,n>N, |a_m - a_n| < ε
    """
    if len(suku) < 20:
        return False
    n = len(suku)
    blocks = [suku[i:i+10] for i in range(0, n-10,
10)]
    if not blocks:
        return False
    max_dists = []

    for block in blocks:
        max_dist = max(nilai_mutlak(block[i] -
block[j]) for i in range(len(block)) for j in
range(i+1, len(block)))
        max_dists.append(max_dist)
    return len(max_dists) >= 3 and all(max_dists[i]
>= max_dists[i+1] for i in range(len(max_dists)-1))
and max_dists[-1] < epsilon

def cek_konvergensi(formula):
    """
    Memeriksa konvergensi barisan yang
didefinisikan oleh formula menggunakan
    berbagai kriteria dan pendekatan.
    """
    limit_value = hitung_limit_simbolik(formula)
    ratio_result, ratio_value = ratio_test(formula)
    root_result, root_value = root_test(formula)

    suku = []
    for n in range(1, 150):
        nilai = evaluasi_formula(formula, n)
        if nilai is not None and not
math.isnan(nilai) and not math.isinf(nilai):
            suku.append(nilai)
    if len(suku) < 10:
        return False, None, "Formula tidak valid
atau tidak menghasilkan cukup data"

    numerik_limit, numerik_konvergen =
cek_limit_numerik(suku)
    cauchy_konvergen = uji_cauchy(suku)

    konvergen = False
    result_limit = None
    alasan = ""
    if ratio_result == "konvergen":
        konvergen = True
        alasan = f"Uji rasio menunjukkan
konvergensi (rasio = {ratio_value:.4f} < 1)"
        result_limit = limit_value if limit_value
is not None else numerik_limit
    elif root_result == "konvergen":
        konvergen = True
        alasan = f"Uji akar menunjukkan konvergensi
(akar = {root_value:.4f} < 1)"
        result_limit = limit_value if limit_value
is not None else numerik_limit
    elif ratio_result == "divergen":
        konvergen = False
        alasan = f"Uji rasio menunjukkan divergensi
(rasio = {ratio_value:.4f} > 1)"
    elif root_result == "divergen":
        konvergen = False
        alasan = f"Uji akar menunjukkan divergensi
(akar = {root_value:.4f} > 1)"
    elif limit_value is not None:
        konvergen = True
        result_limit = limit_value
        alasan = f"Limit barisan dihitung secara
simbolik = {limit_value}"
    elif numerik_konvergen is True:
        konvergen = True
        result_limit = numerik_limit
        alasan = f"Barisan tampak konvergen numerik
ke {numerik_limit:.6f}"
    elif numerik_konvergen is False:
```

CC BY SA

```
        konvergen = False
        alasan = "Barisan tampak divergen
berdasarkan analisis numerik"
    elif cauchy_konvergen:
        konvergen = True
        result_limit = sum(suku[-10:]) / 10
        alasan = f"Barisan memenuhi kriteria
Cauchy, perkiraan limit = {result_limit:.6f}"
    else:
        differences = [suku[i] - suku[i-1] for i in
range(1, len(suku))]
        sign_changes = sum(1 for i in range(1,
len(differences)) if differences[i] *
differences[i-1] < 0)
        if sign_changes > len(differences) * 0.4:
            last_values = suku[-20:]
            avg_value = sum(last_values) /
len(last_values)

            amplitudes = [nilai_mutlak(v -
avg_value) for v in last_values]
            decreasing_amplitude =
all(amplitudes[i] >= amplitudes[i+5] for i in
range(len(amplitudes)-5))
            if decreasing_amplitude:
                konvergen = True
                result_limit = avg_value
                alasan = f"Barisan berosilasi
dengan amplitudo menurun, perkiraan limit =
{result_limit:.6f}"
            else:
                konvergen = False
                alasan = "Barisan berosilasi dan
tampak tidak konvergen"
        elif all(nilai_mutlak(suku[i]) > 1e10 for i
in range(len(suku)-5, len(suku))):
            konvergen = False
            alasan = "Barisan divergen (nilai
membesar tanpa batas)"
```

```
        else:
            konvergen = False
            alasan = "Tidak dapat menentukan
konvergensi dengan pasti menggunakan metode yang
tersedia"
    return konvergen, result_limit, alasan

def main():
    print("PROGRAM PEMERIKSAAN KONVERGENSI
BARISAN")
    print("========================================"
)
    print("Masukkan formula barisan dalam bentuk
ekspresi dengan 'n'")
    while True:
        formula = input("\nMasukkan formula barisan
a_n (atau 'exit' untuk keluar): ")
        if formula.lower() == 'exit':
            break
        if not re.search(r'\bn\b', formula):
            print("Error: Formula harus mengandung
variabel 'n'")
            continue
        valid = False
        for i in range(1, 10):
            if evaluasi_formula(formula, i) is not
None:
                valid = True
                break
        if not valid:
            print("Formula tidak valid. Silakan
coba lagi.")
            continue
        print("\nMemeriksa konvergensi barisan...")
        konvergen, limit, alasan =
cek_konvergensi(formula)
        print("\nHasil Pemeriksaan:")
        print("------------------")
        print(f"Formula: {formula}")
```

```
        if konvergen:
            if limit is not None:
                print(f"Kesimpulan: Barisan
KONVERGEN ke {limit}")
            else:
                print("Kesimpulan: Barisan
KONVERGEN tetapi nilai limitnya tidak dapat
ditentukan")
        else:
            print("Kesimpulan: Barisan TIDAK
KONVERGEN")
        print(f"Alasan: {alasan}")

        print("\nBeberapa suku barisan:")
        print(f"{'n':>5} {'a_n':>18}")
        for n in [1, 2, 3, 4, 5, 10, 20, 50, 100]:
            an = evaluasi_formula(formula, n)
            if an is not None:
                if abs(an) > 1e12:
                    print(f"{n:>5} {an:>18.6e}")
                else:
                    print(f"{n:>5} {an:>18.10f}")
        if konvergen and limit is not None:
            print("\nVerifikasi konvergensi:")
            print(f"{'n':>5} {'a_n':>18} {'|a_n -
L|':>18}")
            for n in [50, 100, 500, 1000]:
                an = evaluasi_formula(formula, n)
                if an is not None:
                    selisih = nilai_mutlak(an -
limit)
                    if abs(an) > 1e12:
                        print(f"{n:>5} {an:>18.6e}
{selisih:>18.6e}")
                    else:
                        print(f"{n:>5} {an:>18.10f}
{selisih:>18.10f}")
if _name_ == "_main_":
    main()
```

Program yang dikembangkan dirancang secara modular menggunakan pendekatan terstruktur, yang terdiri atas sejumlah fungsi utama yang merepresentasikan tahapan-tahapan analitis dalam proses evaluasi konvergensi barisan bilangan real. Modularitas ini memberikan fleksibilitas tinggi dalam pengembangan dan pemeliharaan sistem, serta memungkinkan integrasi efisien antara metode simbolik dan numerik dalam proses analisis.

Fungsi `evaluasi_formula(formula, n)` merupakan komponen awal yang digunakan untuk mengevaluasi nilai suku ke-n berdasarkan formula eksplisit yang diberikan oleh pengguna. Proses ini mengandalkan pustaka *SymPy* untuk pemrosesan simbolik, memungkinkan program menginterpretasikan dan menghitung ekspresi matematika secara simbolik. Apabila ekspresi tidak dapat diproses secara simbolik akibat kompleksitas bentuk atau keterbatasan pustaka, maka sistem secara otomatis menggunakan pendekatan alternatif melalui fungsi `eval()`, yang dibatasi oleh konteks keamanan guna menjamin integritas dan perlindungan terhadap eksekusi kode yang tidak diinginkan.

Pendekatan analitis untuk evaluasi konvergensi diimplementasikan melalui dua metode utama, yakni uji rasio dan uji akar. Fungsi `ratio_test(formula)` mengimplementasikan uji rasio dengan menghitung limit dari nilai absolut perbandingan dua suku berurutan dalam barisan.

9.2.4 Uji Rasio Misalkan $A := \boldsymbol{a_n}$ adalah barisan bilangan real bukan nol. (a) Jika terdapat $r \in \mathbb{R}$, di mana $0 < r < 1$ dan $N \in \mathbb{N}$ sehingga $\left|\frac{\boldsymbol{a_{n+1}}}{\boldsymbol{a_n}}\right| \leq r$ untuk $n \geq N$, maka deret $\sum \boldsymbol{a_n}$ konvergen mutlak. (b) Jika terdapat $N \in \mathbb{N}$ sehingga $\left|\frac{\boldsymbol{a_{n+1}}}{\boldsymbol{a_n}}\right| \geq 1$ untuk $n \geq N$, maka deret $\sum \boldsymbol{a_n}$ divergen.

Sementara itu, fungsi `root_test(formula)` menerapkan uji akar, yang menghitung limit dari akar pangkat-n dari nilai absolut suatu suku.

9.2.2 Uji Akar Misalkan $A := \boldsymbol{a_n}$ adalah barisan di $\mathbb{R}$. (a) Jika terdapat $r \in \mathbb{R}$, di mana $r < 1$ dan $N \in \mathbb{N}$ sehingga $|\boldsymbol{a_n}|^{1/n} \leq r$ untuk $n \geq N$, maka deret $\sum \boldsymbol{a_n}$ konvergen mutlak. (b) Jika terdapat $N \in \mathbb{N}$ sehingga $|\boldsymbol{a_n}|^{1/n} \geq 1$ untuk $n \geq N$, maka deret $\sum \boldsymbol{a_n}$ divergen.

Kedua uji tersebut diupayakan untuk diselesaikan secara simbolik dengan *SymPy*. Namun, apabila hasil tidak dapat diperoleh secara simbolik, maka pendekatan numerik digunakan sebagai estimasi berdasarkan data suku-suku barisan.

CC BY SA

Proximal: Jurnal Penelitian Matematika dan Pendidikan Matematika
ISSN 26158132 (cetak)
ISSN 26157667 (online)

Untuk mendukung pendekatan numerik secara lebih kuat, fungsi `cek_limit_numerik(suku)` dikembangkan guna mengevaluasi kestabilan nilai pada bagian akhir barisan. Fungsi ini memeriksa apakah nilai-nilai suku terakhir menunjukkan kecenderungan mendekati suatu nilai tetap, sebagai dasar estimasi empiris terhadap keberadaan limit barisan tersebut.

Sebagai pendekatan alternatif yang tidak memerlukan pengetahuan terhadap nilai limit, fungsi `uji_cauchy(suku)` mengimplementasikan prinsip kriteria *Cauchy*.

Definisi 3.5.1 Suatu barisan $A = (a_n)$ dari bilangan-bilangan real dikatakan barisan *Chaucy* jika untuk setiap $\varepsilon > 0$ terdapat bilangan asli $H(\varepsilon)$ sedemikian sehingga untuk semua bilangan asli $n, m \geq H(\varepsilon)$, suku-suku $\boldsymbol{a_n}, \boldsymbol{a_m}$ memenuhi $|\boldsymbol{a_n} - \boldsymbol{a_m}| < \varepsilon$

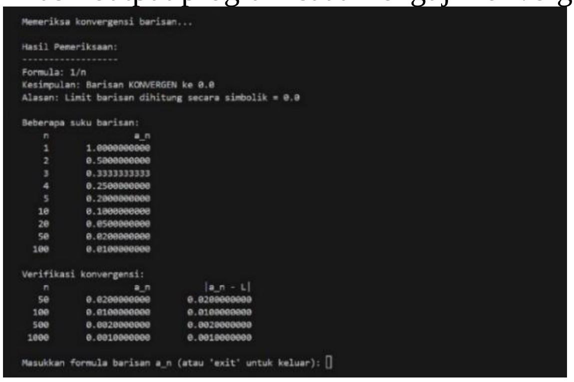
Implementasi dalam program dilakukan dengan memeriksa selisih antara suku-suku dalam blok nilai tertentu, untuk mengamati apakah selisih tersebut menunjukkan kecenderungan menuju nol.

Seluruh pendekatan simbolik, numerik, dan kriteria *Cauchy* tersebut terintegrasi dalam fungsi `cek_konvergensi(formula)` yang menjadi inti sistem evaluasi. Fungsi ini memadukan hasil dari evaluasi limit simbolik, uji rasio, uji akar, serta hasil dari pendekatan numerik dan kriteria *Cauchy*. Selain itu, fungsi ini juga mengevaluasi karakteristik tambahan dari barisan seperti pola osilasi (fluktuasi periodik atau acak) dan pertumbuhan nilai barisan (meningkat, menurun, atau tetap), guna memperoleh pemahaman menyeluruh terhadap perilaku barisan yang dianalisis. Berdasarkan keseluruhan hasil analisis, sistem menyimpulkan apakah suatu barisan bersifat konvergen, divergen, atau belum dapat ditentukan secara pasti.

Fungsi `main()` dalam program bertindak sebagai antarmuka berbasis teks yang bersifat interaktif. Melalui antarmuka ini, pengguna dapat memberikan *input* berupa formula eksplisit dari barisan yang ingin dianalisis. Setelah menerima masukan, program menjalankan serangkaian proses evaluasi dan analisis secara otomatis, dan menyajikan hasil simpulan dalam format terstruktur dan informatif. Sebagai pelengkap, program juga menampilkan sejumlah suku awal dari barisan tersebut untuk memberikan ilustrasi numerik terhadap perilaku awal barisan.

Secara keseluruhan, alur kerja program dimulai dari penerimaan *input* formula, dilanjutkan dengan evaluasi nilai suku-suku barisan baik secara simbolik maupun numerik, yang kemudian digunakan dalam berbagai uji konvergensi. Hasil dari masing-masing uji, baik simbolik, numerik, maupun berdasarkan kriteria Cauchy, digabungkan bersama dengan analisis osilasi dan tren pertumbuhan, untuk memberikan simpulan akhir. Pendekatan terintegrasi ini menegaskan keterpaduan antara teori analisis real dan implementasi komputasional, serta menunjukkan potensi program sebagai sarana bantu dalam pembelajaran maupun eksplorasi mandiri terhadap konsep konvergensi barisan.

Untuk mengilustrasikan kinerja dan keluaran dari program yang dikembangkan, berikut disajikan beberapa contoh hasil output program saat menguji konvergensi suatu barisan.

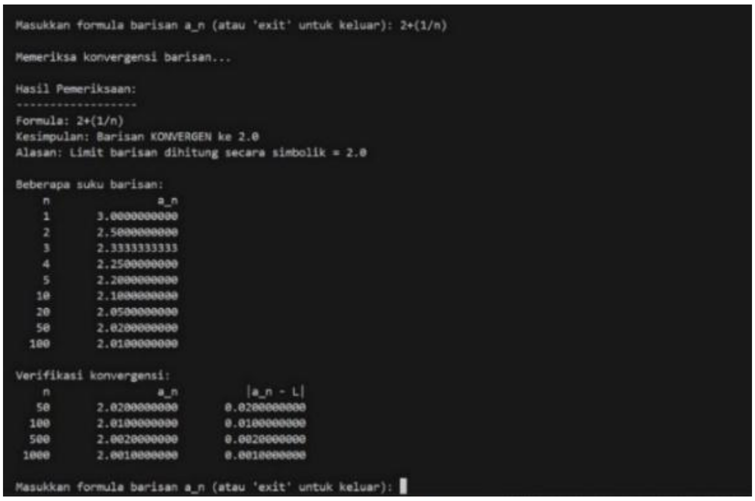


Gambar 1 Hasil *output* program untuk barisan $\boldsymbol{a_n} = \frac{\boldsymbol{1}}{\boldsymbol{n}}$.

CC BY SA

Program menerima input berupa formula eksplisit barisan, kemudian melakukan evaluasi limit secara simbolik untuk menentukan konvergensi. Dalam Gambar 1, program menghasilkan kesimpulan bahwa barisan konvergen ke 0, berdasarkan hasil limit yang dihitung menggunakan pendekatan simbolik. Selanjutnya, ditampilkan beberapa suku awal barisan guna memperlihatkan perilaku numerik dari a_n seiring bertambahnya nilai n. Bagian akhir output menunjukkan proses verifikasi konvergensi melalui evaluasi nilai mutlak selisih $|a_n - L|$ untuk n besar, yang semakin mendekati nol. Hal ini mengindikasikan bahwa program tidak hanya mengandalkan analisis simbolik, tetapi juga mengimplementasikan pendekatan numerik untuk memperkuat kesimpulan tentang sifat konvergensi suatu barisan.

Sebagai penyesuaian terhadap hasil program, konvergensi barisan $a_n = \frac{1}{n}$ dapat dibuktikan secara manual menggunakan definisi limit barisan. Berdasarkan definisi tersebut, barisan (a_n) dikatakan konvergen ke suatu bilangan real L jika untuk setiap $\varepsilon > 0$, terdapat bilangan bulat positif N sedemikian sehingga untuk setiap $n > N$, berlaku:$|a_n - L| < \varepsilon$. Untuk barisan ini, diketahui bahwa $L = 0$, sehingga: $\left|\frac{1}{n} - 0\right| = \frac{1}{n} < \varepsilon$. pertidaksamaan tersebut ekuivalen dengan: $n > \frac{1}{\varepsilon}$. Dengan demikian, cukup dipilih $N \in \mathbb{N}$ sedemikian sehingga $N > \frac{1}{\varepsilon}$, maka untuk setiap $n \geq N$, diperoleh $\frac{1}{n} < \varepsilon$. Hal ini membuktikan bahwa barisan $a_n = \frac{1}{n}$ konvergen ke 0. Pembuktian ini menunjukkan bahwa hasil program telah sesuai dengan formal konvergensi barisan secara analitik.



Gambar 2 Hasil *output* program untuk barisan $\boldsymbol{a_n = 2 + \frac{1}{n}}$.

Berdasarkan hasil output pada Gambar 2, sistem secara simbolik menghitung bahwa barisan tersebut konvergen menuju nilai limit $L = 2.0$. Penyesuaian terhadap hasil program juga dapat ditunjukkan melalui pembuktian manual menggunakan definisi limit. Misalkan: $a_n = 2 + \frac{1}{n}$. Untuk membuktikan bahwa $\lim_{n\to\infty} a_n = 2$, dapat menggunakan definisi limit barisan. Diberikan $\varepsilon > 0$, untuk menentukan $N \in \mathbb{N}$ sedemikian sehingga untuk setiap $n > N$, berlaku: $|a_n - 2| = \left|2 + \frac{1}{n} - 2\right| = \frac{1}{n} < \varepsilon$. pertidaksamaan ini identik dengan pembuktian pada barisan sebelumnya, sehingga cukup memilih $n > N$, diperoleh $|a_n - 2| < \varepsilon$, yang menunjukkan bahwa barisan a_n konvergen ke 2. Hal ini sesuai dengan teori dasar limit, di mana penambahan suatu bilangan tetap terhadap suku barisan yang konvergen tidak mempengaruhi sifat konvergensi, melainkan hanya menggeser nilai limitnya. Secara umum, jika $\lim_{n\to\infty} b_n = L$, maka $\lim_{n\to\infty}(b_n + c) = L + c$, $2 + \frac{1}{n}$ nilai limit bergeser dari 0 menjadi 2 akibat penambahan bilangan konstan 2, sedangkan sifat konvergen tetap dipertahakan. Dengan demikian, barisan a_n tidak hanya konvergen, tetapi juga merupakan transformasi

CC BY SA

https://doi.org/10.30605/proximal.v8i2.5732
Volume 8 Nomor 2, Tahun 2025

linier dari barisan $\frac{1}{n}$ yang telah dibahas sebelumnya. Untuk memperkuat hasil analisis simbolik tersebut, program menampilkan sejumlah suku awal dari barisan. Dari representasi numerik tersebut, tampak bahwa nilai a_n mendekati 2 secara bertahap seiring bertambahnya n. Hal ini semakin memperkuat konsistensi antara hasil program dan perhitungan manual berdasarkan definisi formal.

Program evaluasi konvergensi barisan yang dikembangkan menunjukkan sejumlah keunggulan yang menandai integrasi efektif antara pendekatan simbolik dan numerik dalam analisis matematis. Melalui pemanfaatan perhitungan limit secara simbolik serta verifikasi numerik terhadap nilai-nilai suku pada indeks besar, program ini mampu menghasilkan analisis yang tidak hanya akurat, tetapi juga komprehensif. Kombinasi metode ini memperkuat keandalan evaluasi dan memperluas jangkauan diagnostik terhadap perilaku konvergensi barisan. Fleksibilitas sistem dalam menerima berbagai bentuk ekspresi matematika, termasuk fungsi trigonometri, logaritma, eksponensial, maupun bentuk kompleks lainnya, menjadi nilai tambah yang signifikan. Kemampuan ini memungkinkan program untuk diaplikasikan dalam spektrum analisis yang lebih luas, termasuk pada barisan-barisan yang muncul dalam konteks lanjutan analisis real. Selain itu, program juga menunjukkan keandalan dalam menangani barisan osilatif atau yang tidak memiliki limit eksplisit, dengan cara mengamati pola perubahan dan fluktuasi amplitudo guna menilai kecenderungan konvergensi secara lebih mendalam.

Dari aspek pedagogis, program ini dirancang dengan pendekatan interaktif dan edukatif. Antarmuka berbasis terminal yang sederhana namun informatif, serta penyajian hasil yang terstruktur secara sistematis, mendukung penggunaan dalam konteks pembelajaran. Mahasiswa maupun pendidik dapat memanfaatkannya sebagai media eksplorasi konseptual, sehingga memperkuat pemahaman terhadap konsep limit dan konvergensi secara aplikatif.

Namun demikian, terdapat sejumlah keterbatasan yang perlu dicermati sebagai pijakan untuk pengembangan lanjutan. Salah satu tantangan utama terletak pada ketergantungan terhadap format sintaks ekspresi yang harus sesuai dengan standar pemrosesan simbolik. Kesalahan dalam penulisan formula dapat menyebabkan kegagalan eksekusi atau interpretasi yang keliru. Di samping itu, pada barisan yang sangat kompleks atau bersifat piecewise, program masih memiliki keterbatasan presisi numerik dan kedalaman analisis.

Dari segi tampilan, absennya visualisasi grafis menjadi salah satu aspek yang dapat ditingkatkan, mengingat representasi visual seperti grafik suku atau pola konvergensi dapat membantu memperjelas pemahaman pengguna. Selain itu, program belum menyediakan penjelasan konseptual otomatis, sehingga pengguna yang belum menguasai teori limit mungkin memerlukan pendampingan tambahan untuk menginterpretasi hasil secara tepat.

Secara keseluruhan, program ini menawarkan kontribusi yang substansial dalam ranah pembelajaran dan eksplorasi matematika, khususnya dalam memahami konvergensi barisan. Keunggulan yang dimiliki menunjukkan potensi besar dalam implementasi pendidikan berbasis teknologi, sementara keterbatasan yang ada membuka ruang untuk inovasi berkelanjutan guna menjadikannya lebih inklusif, adaptif, dan berdaya guna di berbagai konteks akademik.

D. Kesimpulan

Penelitian ini menunjukkan bahwa analisis konvergensi barisan dengan konsep nilai mutlak dapat diselesaikan secara lebih sistematis menggunakan pendekatan numerik dan simbolik. Implementasi *Python* sebagai alat bantu analisis memberikan hasil yang akurat dalam menentukan sifat konvergensi suatu barisan, terutama melalui penggunaan kriteria *Cauchy*, uji rasio, dan uji akar. Kombinasi pendekatan numerik dan simbolik memastikan keakuratan hasil, sehingga dapat menjadi solusi inovatif dalam menyelesaikan permasalahan konvergensi barisan. Selain itu, penggunaan teknologi komputasi dalam analisis matematika

https://doi.org/10.30605/proximal.v8i2.5732
Volume 8 Nomor 2, Tahun 2025

terbukti meningkatkan efisiensi serta ketelitian perhitungan, yang pada akhirnya mendukung pemahaman konsep secara lebih mendalam dan sistematis.

Pengembangan lebih lanjut dapat dilakukan dengan menambahkan fitur visualisasi yang lebih interaktif serta mengintegrasikan metode yang lebih kompleks guna memperluas cakupan analisis. Dengan adanya pengolahan data yang lebih canggih, penelitian ini dapat diperluas ke ranah aplikasi yang lebih luas, seperti model prediksi berbasis analisis deret. Selain itu, integrasi algoritma yang lebih canggih dalam perangkat lunak berbasis *Python* dapat meningkatkan akurasi dan efektivitas analisis konvergensi, sehingga mampu memberikan kontribusi yang lebih besar dalam kajian matematika terapan maupun murni.

DAFTAR PUSTAKA

Alfarizi, M. R., Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning. *Karimah Tauhid*, *2*(1), 1–6. https://doi.org/10.30997/karimahtauhid.v2i1.7518

Angin, C. P., Tambunan, C. P., Lubis, R. H., Siregar, U. M., & Piliang, Y. K. (2024). Analisis Kemampuan Mahasiswa Matematika FMIPA Unimed dalam Menyelesaikan Permasalahan Konvergensi dan Divergensi Barisan Bilangan Real dengan Berbantuan Software MATLAB. *Jurnal Matematika, Ilmu Pengetahuan Alam, Kebumian Dan Angkasa*, *2*(6), 76–86.https://doi.org/10.62383/algoritma.v2i6.281

Apriani, D. E., Siburian, G., Situmorang, R., Simamora, S. A., & Hutapea, T. A. (2025). Menyelesaikan Permasalahan Supremum dan Infimum Suatu Himpunan dengan Menggunakan Python. *Jurnal Media Akademik (JMA)*, *3*(3), 1-14. https://doi.org/10.62281/v3i3.1683

Dasawarsa, A. R., Sucipto, A., Sofyan, A., Saptaji, M., & Rosyani, P. (2023). Analisis Penerapan Python Dalam Perhitungan Limit. *Jurnal Matematika*, *Algoritma dan Sains*, *1*(1), 31–35. https://ojs.jurnalmahasiswa.com/ojs/index.php/alkhawarizmi/article/view/202

Hutapea, T. A., Sinambela, P. N., & Adlin, D. (2020). Ability of Problem Solving Students Based on Information and Communication Technology. *Journal of Physics: Conference Series*, 1-5. doi:10.1088/1742-6596/1485/1/012052

Isnani, Waluya, S. B., Rochmad, Dwijanto, & Asih, R. S. (2021). Analisis Miskonsepsi Mahasiswa pada Matakuliah Analisis Real. *Prosiding Seminar Nasional Matematika, 4*, 235-238. https://journal.unnes.ac.id/sju/index.php/prisma/

Rahmadani, N., Aulia, C. N., Laia, L. H., Maigani, & Simanullang, M. C. (2025). Studi Kesulitan Mahasiswa dalam Mempelajari Deret Tak Hingga: Studi Kasus di Jurusan Matematika UNIMED. *Lebesgue: Jurnal Ilmiah Pendidikan Matematika*, *Matematika dan Statistika*, *6*(1), 428-433. doi:10.46306/lb.v6i1.930

Saharuddin, & Prihatmono, M. W. (2022). Pengenalan dan pelatihan dasar bahasa pemrograman Python pada siswa/i SMA Negeri 3 Makassar. *Jurnal Pengabdian Masyarakat Berkemajuan,*, *6*(4), 2233-2237. https://doi.org/10.31764/jpmb.v6i4.10569

Saragih, V. Y., Kusfa, B. D., & Fitria, R. I. (2024). Penerapan Python dalam Perhitungan Turunan Fungsi Dua Peubah dan Visualisasi Grafik 3D. *Jurnal Arjuna : Publikasi Ilmu Pendidikan, Bahasa dan Matematika*, *3*(1), 76–83. https://doi.org/10.61132/arjuna.v3i1.141.