



Penggunaan *Python* dalam Pengerjaan Induksi Matematika

Salsabila Arvi^{1*}, Ikrimah Sabina Triadi², Zahra Putri³, Rhamanda Ardiansyah Lubis⁴,
Fitriyani Fitriyani⁵

¹⁻⁵Universitas Negeri Medan, Indonesia

Jl. W. Iskandar Pasar V Medan Esatate Kab. Deli Serdang

Korespondensi penulis: arvi@mhs.unimed.ac.id*

Abstract. *Mathematics is a science that is structured deductively and systematically. Proof in Mathematics is important because it can enable critical thinking logically, and the truth of a hypothesis can be tested. Mathematical induction is a proof method that has 2 steps, namely basis and induction. With advances in technology today, there are many applications that can make this proof easier, such as Python. This research uses quantitative and qualitative approaches to prove the effectiveness of using Python compared to manual proof. The results show that Python not only speeds up work but also minimizes errors that could occur if done manually. With this research we recommend further exploitation of mathematical induction in other programming applications.*

Keywords: *Mathematical Induction, Python, Numpy*

Abstrak. Matematika adalah ilmu pengetahuan yang tersusun secara deduktif dan sistematis. Pembuktian dalam Matematika penting karena dapat membuat berpikir kritis secara logis, serta kebenaran suatu hipotesis dapat teruji. Induksi Matematika adalah suatu metode pembuktian yang memiliki 2 langkah yaitu basis dan induksi. Dengan kemajuan teknologi zaman sekarang banyak aplikasi yang dapat memudahkan pembuktian ini seperti *python*. Penelitian ini menggunakan pendekatan kuantitatif dan kualitatif untuk membuktikan keefektifan penggunaan *python* dibandingkan pembuktian manual. Hasil menunjukkan *python* tidak hanya mempercepat pengerjaan tetapi juga meminimalisir kesalahan yang dapat terjadi jika dikerjakan secara manual. Dengan penelitian ini kami merekomendasikan untuk eksploitasi lebih lanjut mengenai induksi matematika dalam aplikasi-aplikasi pemrograman lain.

Kata kunci: Induksi Matematika, *Python*, *Numpy*

1. LATAR BELAKANG

Matematika adalah ilmu pengetahuan yang tersusun secara deduktif aksiomatik yang berarti Matematika diawali dengan istilah yang belum didefinisikan (Utomo & Huda, 2020). Salah satu ciri utama Matematika adalah deduktif, yaitu kebenaran pernyataan atau konsep merupakan akibat logis dari kebenaran sebelumnya sehingga kaitan antar pernyataan dan konsep bersifat konsisten (Ernawati, 2020). Tarhadi dan pujiastuti (Firmasari, 2019) mengungkapkan pembuktian dalam Matematika penting karena dapat membuat kita berpikir kritis secara logis dan sistematis, serta kebenaran suatu hipotesis dapat teruji.

Induksi Matematika adalah suatu metode pembuktian yang sering digunakan untuk menunjukkan suatu pernyataan benar untuk semua bilangan asli. Metode pembuktian ini melibatkan 2 langkah yaitu basis dan induksi. Basis adalah pembuktian untuk bilangan terkecil seperti $n = 1$ adalah benar, sedangkan Langkah induksi jika suatu pernyataan benar untuk $n = k$ maka pernyataan tersebut juga harus benar untuk $n = k + 1$.

Dengan berkembangnya teknologi saat ini, banyak aplikasi-aplikasi pemrograman yang dapat memberi manfaat dalam pembelajaran dan penerapan konsep-konsep Matematika termasuk induksi Matematika seperti penggunaan aplikasi Python. Dalam pemrograman, Python dengan berbagai pustaka matematis seperti SymPy memungkinkan pengguna untuk melakukan perhitungan simbolik dan melakukan pembuktian dengan lebih efisien dan mudah, sehingga dapat berguna untuk berbagai contoh pembuktian induksi.

Python sangat populer dalam persoalan matematika salah satunya adalah Induksi matematika. Hal ini juga yang menjadi alasan kami menggunakan aplikasi pemrograman Python yang mudah dipelajari, Fleksibel, memiliki *Library* yang kuat seperti NumPy, dan tentunya dapat diakses oleh semua orang.

2. KAJIAN TEORITIS

Induksi Matematika

Induksi matematika merupakan sebuah teknik yang digunakan untuk membuktikan kebenaran sebuah pernyataan terkait bilangan asli. Induksi matematika memiliki bentuk formal yang dinyatakan dalam prinsip berikut (Bartle & Sherbert, 2010).

Prinsip Induksi Matematika | Misalkan S adalah himpunan bagian dari \mathbb{N} (bilangan asli) yang memiliki dua sifat berikut.

1. Angka 1 termasuk dalam S ($1 \in S$)
2. Untuk $k \in \mathbb{N}$, jika $k \in S$, maka $k + 1 \in S$

Maka, dapat disimpulkan bahwa $S = \mathbb{N}$.

Ketika kedua sifat terpenuhi, maka dapat disimpulkan bahwa $S = \mathbb{N}$. Pernyataan pertama disebut sebagai *basic step/base*, sedangkan pernyataan kedua disebut sebagai *induction hypothesis*

Prinsip Induksi Matematika Kedua | Misalkan $n_0 \in \mathbb{N}$ dan misalkan $P(n)$ menyatakan bahwa untuk setiap $n \geq n_0$, maka pertimbangkan kedua kondisi berikut.

- (1) Pernyataan $P(n_0)$ benar.
- (2) Untuk semua $k \geq n_0$, $P(k)$ benar mengimplikasikan bahwa $P(k + 1)$ benar.

Maka, $P(n)$ benar untuk semua $n \geq n_0$.

Prinsip kedua sendiri memiliki bentuk lain yang ekuivalen, yaitu:

Prinsip Induksi Kuat | Misalkan S adalah himpunan bagian dari \mathbb{N} sehingga

1. $1 \in S$.
2. Untuk setiap $k \in \mathbb{N}$, jika $\{1, 2, \dots, k\} \subseteq S$, maka $k + 1 \in S$.

Maka, $S = \mathbb{N}$.

Dalam kehidupan sehari-hari, konsep induksi matematika dapat dinyatakan dalam efek domino. Pada kasus ini, agar domino $(n + 1)$ jatuh, domino n harus jatuh terlebih dahulu, barulah implikasi “Jika domino n jatuh, maka domino $(n + 1)$ jatuh” dapat terjadi. Maka, efek domino ini dapat dijelaskan dalam kasus induksi matematika, di mana harus benar jatuh domino pertama, agar domino $(n + 1)$ juga jatuh (Thariq & Apviano, 2020).

Python

Python adalah salah satu bahasa pemrograman yang umum digunakan oleh banyak *developer* untuk mengembangkan berbagai aplikasi dan simulasi. Python diciptakan pada tahun 1990 oleh Guido van Rossum di Belanda. Sintaks yang lebih sederhana dan ringkas juga pustaka yang luas menjadi poin lebih Python, sehingga Python kerap digunakan bahkan dalam bidang pendidikan (Romzi & Kurniawan, 2020). Python menyiapkan banyak pustaka (*library*) yang dapat digunakan, seperti NumPy, SymPy, dan lainnya yang dapat digunakan sesuai kebutuhan (Ua, dkk., 2023).

Induksi Matematika dengan Python

Pembuktian induksi matematika merupakan teknik fundamental dalam matematika yang digunakan untuk membuktikan pernyataan yang berlaku untuk semua bilangan bulat positif. Dalam konteks pemrograman, Python menawarkan berbagai alat dan pustaka yang dapat digunakan untuk mempermudah dan mengotomatisasi proses ini. Pustaka seperti SymPy memungkinkan pengguna untuk melakukan perhitungan simbolik dan manipulasi ekspresi matematika dengan efisien, menjadikannya sangat berguna untuk pembuktian induksi.

Dalam pembuktian induksi matematika, kadang diperlukan validasi atau verifikasi langkah-langkah basis dan langkah induksi. SymPy dapat digunakan untuk menyelesaikan perhitungan simbolik yang diperlukan dalam langkah-langkah ini. Misalnya, dalam pembuktian jumlah deret aritmatika, SymPy digunakan untuk membuktikan bahwa jumlah dari $(1 + 2 + \dots + n)$ sesuai dengan rumus $\frac{n(n+1)}{2}$, dan verifikasi ini dapat dilakukan dengan memasukkan ekspresi simbolik ke dalam kode Python.

Penerapan Python dalam pembuktian induksi melibatkan penggunaan pustaka komputasi untuk melakukan perhitungan dan verifikasi secara otomatis. Dengan menggunakan NumPy dan Matplotlib, pengguna dapat menangani perhitungan numerik dan membuat visualisasi untuk memeriksa hasil pembuktian. NumPy sangat berguna dalam evaluasi ekspresi numerik yang sering muncul dalam pembuktian induksi, sedangkan Matplotlib dapat digunakan untuk menggambarkan pola atau hasil pembuktian secara grafis.

Sebagai contoh, dalam pembuktian induksi untuk deret bilangan, Python dapat digunakan untuk menulis skrip yang mengotomatiskan langkah basis dan langkah induksi. Kode Python dapat memvalidasi apakah hasil yang diperoleh sesuai dengan rumus yang diberikan dan memberikan solusi yang tepat dalam bentuk simbolik atau numerik. Hal ini tidak hanya mempercepat proses pembuktian tetapi juga meningkatkan akurasi hasil dengan meminimalkan potensi kesalahan manusia.

Penelitian terdahulu menunjukkan bahwa penggunaan Python dalam pembuktian matematika, termasuk induksi, memberikan manfaat signifikan dalam hal efisiensi dan keakuratan. Misalnya, SymPy memungkinkan pengguna untuk melakukan manipulasi aljabar yang kompleks dan pembuktian secara simbolik, sedangkan NumPy dan Matplotlib memperluas kemampuan Python dalam menangani data numerik dan visualisasi. Kombinasi alat-alat ini mendukung pengembangan skrip yang mampu menyelesaikan dan memverifikasi pembuktian induksi dengan lebih cepat dan efektif.

Secara keseluruhan, penggunaan Python dalam pembuktian induksi matematika memanfaatkan kekuatan pemrograman untuk menyederhanakan dan mengotomatisasi proses pembuktian yang kompleks. Dengan alat yang tepat, Python dapat mempermudah verifikasi langkah-langkah induksi dan memberikan hasil yang akurat, menjadikannya alat yang berharga dalam analisis matematis dan pendidikan.

3. METODE PENELITIAN

Penelitian ini menggunakan penelitian aplikatif dan eksperimental yang mengabungkan konsep induksi matematika dengan aplikasi pemrograman menggunakan python. Tujuannya adalah untuk mengevaluasi efektivitas python dalam membantu pembuktian matematika. Penelitian ini juga menggunakan pendekatan kuantitatif dan kualitatif. Penelitian kuantitatif akan berfokus pada jumlah dan ukuran, sedangkan kualitatif akan mengacu pada konsep, definisi, karakteristik, dan hal lain yang berkaitan dengan deskripsi (Firmansyah, dkk., 2021).

Pada pendekatan kuantitatif yang dilihat adalah jumlah langkah yang diperlukan python dalam penyelesaian pembuktian induksi. Sedangkan pada pendekatan kualitatif akan diperhatikan hasil dan keefektifan penggunaan python dibandingkan pembuktian manual.

Langkah awal penelitian dimulai dengan kajian literatur untuk memahami konsep induksi matematika serta penerapannya dalam komputasi menggunakan python. Proses pembuktian ini mengikuti langkah-langkah induksi matematika di mana akan dilakukan pengujian dengan menjalankan skrip python untuk memastikan validitas python. Tahapan selanjutnya dilakukan pengukuran dan analisis hasil. Waktu eksekusi python tentunya

mempermudah dan mempercepat proses. Dokumentasi hasil penelitian mencakup kode, hasil eksekusi dan diskusi mengenai temuan.

4. HASIL DAN PEMBAHASAN

Hasil dan pembahasan menyajikan hasil dari implementasi pembuktian induksi matematika menggunakan bahasa pemrograman python serta analisis terhadap hasil tersebut. Hasil terbagi dua, yaitu hasil eksekusi program dan pembahasan tentang keefektifan python dalam penyelesaian masalah induksi matematika.

Hasil Eksekusi Program

Setelah mengimplementasikan langkah-langkah pembuktian induksi matematika dalam Python, hasil eksekusi menunjukkan bahwa program berhasil menyelesaikan beberapa contoh kasus induksi, seperti pembuktian deret aritmatika dan pembuktian sifat bilangan genap. Contoh kode yang digunakan untuk menyelesaikan pembuktian induksi ini berhasil menunjukkan bahwa langkah-langkah induksi (basis dan langkah induktif) terpenuhi.

Sebagai contoh, untuk pembuktian jumlah deret aritmatika $1 + 2 + \dots + n = \frac{n(n+1)}{2}$ hasil eksekusi menunjukkan bahwa langkah basis pada $n = 1$ valid, dan langkah induksi untuk $n = k$ dan $n = k + 1$ juga memenuhi persamaan yang diberikan. Dengan menggunakan pustaka *SymPy*, program dapat memecahkan perhitungan simbolik ini secara otomatis dan menghasilkan solusi yang akurat. Hasil ini diverifikasi dengan pembuktian manual untuk memastikan keakuratan.

Selain itu, waktu eksekusi program untuk berbagai nilai n diukur untuk mengetahui efisiensi program dalam menyelesaikan masalah. Berdasarkan hasil uji coba, program dapat menyelesaikan pembuktian induksi dalam waktu yang sangat singkat untuk nilai n yang cukup besar, yang menunjukkan bahwa Python efektif dalam menangani pembuktian matematika secara komputasional.



```

1 import sympy as sp
2
3 def sum_natural_numbers(n):
4     return sum(range(1, n + 1))
5
6 def formula_natural_numbers(n):
7     return n * (n + 1) // 2
8
9 def check_induction_for_n(n):
10    print(f"Memeriksa untuk n = {n}")
11    actual_sum = sum_natural_numbers(n)
12    formula_sum = formula_natural_numbers(n)
13    print(f"Jumlah aktual dari 1 hingga {n} = {actual_sum}")
14    print(f"Jumlah dari rumus untuk n = {n} = {formula_sum}")
15
16    if actual_sum == formula_sum:
17        print(f"Rumus benar untuk n = {n}.")
18    else:
19        print(f"Rumus salah untuk n = {n}.")
20
21 if n > 0:
22    generate_graph(n)

```

Gambar 1.

```

23 def generate_graph(n):
24     x_values = list(range(1, n + 1))
25     sum_values = [sum_natural_numbers(i) for i in x_values]
26     formula_values = [formula_natural_numbers(i) for i in x_values]
27
28     plt.figure(figsize=(10, 6))
29     plt.plot(x_values, sum_values, label="Sum Natural Numbers (1 + 2 + ... + n)", marker="o", linestyle="--", color="blue")
30     plt.plot(x_values, formula_values, label="Formula (n*(n+1)/2)", marker="o", linestyle="--", color="green")
31     plt.title("Perbandingan Hasil Pengalasan dan Rumus untuk n hingga (n)")
32     plt.xlabel("n")
33     plt.ylabel("sum(n)")
34     plt.legend()
35     plt.grid(True)
36
37     plt.show()
38
39 if __name__ == "__main__":
40     try:
41         n = int(input("Masukkan nilai n untuk pembuktian rumus: "))
42
43         if n <= 0:
44             print("Silakan masukkan bilangan bulat positif.")
45         else:
46             check_induction_for_n(n)
47
48     except ValueError:
49         print("Input tidak valid. Silakan masukkan bilangan bulat positif.")
50

```

Gambar 2.

Pembahasan

```

1 import matplotlib.pyplot as plt
2
3 def sum_natural_numbers(n):
4     """Menghitung jumlah bilangan 1 hingga n."""
5     return sum(range(1, n + 1))
6
7 def formula_natural_numbers(n):
8     return n * (n + 1) // 2
9
10 def check_induction(n):
11     basis_induction = (sum_natural_numbers(1) == formula_natural_numbers(1))
12     induction_step = all((sum_natural_numbers(i) == formula_natural_numbers(i)) for i in range(1, n + 1))
13     return basis_induction and induction_step
14
15 def visualize_comparison(n):
16     x_values = list(range(1, n + 1))
17     actual_sums = [sum_natural_numbers(i) for i in x_values]
18     formula_sums = [formula_natural_numbers(i) for i in x_values]
19
20     plt.plot(x_values, actual_sums, label="Actual Sum", color="blue")
21     plt.plot(x_values, formula_sums, label="Formula Sum", linestyle="--", color="red")
22     plt.xlabel("n")
23     plt.ylabel("sum")
24     plt.title("Perbandingan Hasil Pengalasan dan Rumus (n*(n+1)/2) hingga n={n}")
25     plt.grid(True)
26     plt.show()
27

```

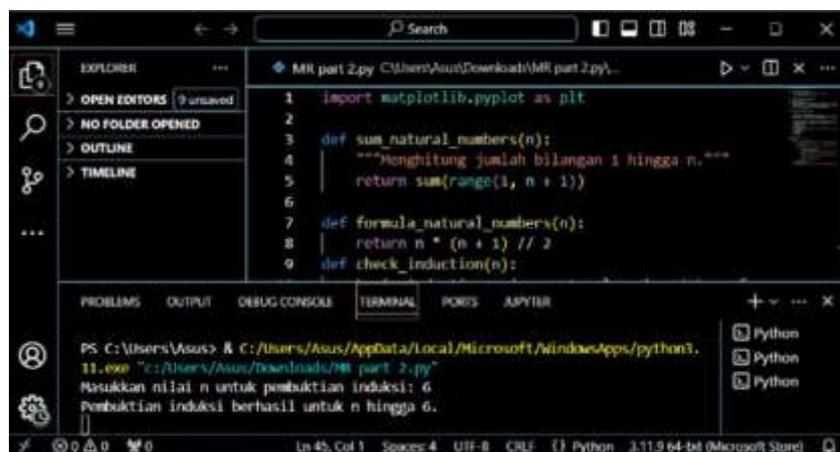
Gambar 3.

```

29 # Main Program
30 if __name__ == "__main__":
31     try:
32         n = int(input("Masukkan nilai n untuk pembuktian induksi: "))
33         if n <= 0:
34             print("Silakan masukkan bilangan bulat positif yang lebih besar dari 0.")
35         else:
36             if check_induction(n):
37                 print("Pembuktian induksi berhasil untuk n hingga (n).")
38             else:
39                 print("Pembuktian induksi gagal untuk n hingga (n).")
40
41             visualize_comparison(n)
42
43     except ValueError:
44         print("Input tidak valid. Silakan masukkan bilangan bulat positif.")
45

```

Gambar 4.



Gambar 5.



Gambar 6. Grafik perbandingan jumlah bilangan dan rumus

Hasil yang diperoleh dari eksekusi program memperlihatkan bahwa Python dapat digunakan secara efektif untuk membantu proses pembuktian induksi matematika. Penerapan *SymPy* untuk perhitungan simbolik sangat memudahkan dalam menyelesaikan permasalahan yang melibatkan pembuktian induktif. Program dapat menjalankan perhitungan otomatis dengan cepat, sehingga mengurangi potensi kesalahan yang mungkin terjadi jika dilakukan secara manual. Hal ini menjadi salah satu keunggulan utama Python dalam menyelesaikan masalah matematika kompleks.

Dibandingkan dengan metode manual, Python mampu memberikan hasil yang lebih efisien, baik dalam hal waktu maupun upaya yang diperlukan. Dalam pembuktian manual, setiap langkah harus diuraikan dengan cermat dan membutuhkan ketelitian tinggi. Dengan Python, proses ini dapat dilakukan secara otomatis, dan hasilnya dapat langsung dihasilkan dalam bentuk simbolik maupun numerik.

Namun, terdapat beberapa tantangan yang muncul dalam penggunaan Python, terutama dalam hal memahami sintaks dan pustaka yang digunakan. Pengguna yang belum familiar dengan Python mungkin memerlukan waktu untuk belajar dan memahami cara kerja pustaka seperti *SymPy*. Meski demikian, setelah dikuasai, Python memberikan fleksibilitas dan efisiensi yang sangat membantu dalam menyelesaikan masalah induksi matematika.

Secara keseluruhan, hasil dari penelitian ini menunjukkan bahwa Python adalah alat yang sangat berguna untuk pembuktian induksi matematika, terutama untuk masalah-masalah yang lebih kompleks atau membutuhkan perhitungan simbolik. Program ini tidak hanya mempercepat proses pembuktian, tetapi juga memungkinkan verifikasi yang lebih mudah dan akurat.

Perbandingan dengan Metode Manual

Untuk menilai keefektifan Python, hasil dari pembuktian dengan Python dibandingkan dengan pembuktian manual. Python jelas memberikan keunggulan dalam hal kecepatan dan kemudahan pengoperasian, terutama saat berhadapan dengan deret yang lebih panjang atau

perhitungan yang lebih kompleks. Dalam pembuktian manual, terdapat potensi kesalahan pada perhitungan atau langkah-langkah induksi, sedangkan dengan Python, proses ini diotomatisasi sehingga meminimalkan kesalahan.

5. KESIMPULAN DAN SARAN

Berdasarkan riset yang telah dilakukan, kesimpulan yang dapat kami ambil adalah:

1. Kita dapat mengonfirmasi tentang bilangan asli (\mathbb{R}) dapat divalidasi dengan menggunakan teknik induksi matematika.
2. Python diciptakan pada tahun 1990 oleh Guido van Rossum di Belanda. Python adalah bahasa pemrograman yang sangat serbaguna, mudah dipelajari, dan populer yang berfungsi untuk pengembangan web, analisis data, kecerdasan buatan, pembuatan permainan, pengembangan aplikasi desktop, dan pemrograman jaringan. Sintaks yang lebih sederhana dan ringkas juga pustaka yang luas menjadi poin lebih Python
3. Penggunaan Python dalam bukti induksi matematika memanfaatkan kemampuan pemrograman untuk menyederhanakan dan mengotomatiskan prosedur pembuktian yang kompleks, itulah kenapa masalah induksi matematika juga cocok saat menggunakan pemrograman Python. Aplikasi pemrograman Python juga dapat menjadi alat yang berguna dengan mengingatnya Python dalam analisis dan instruksi matematika dengan menyederhanakan proses verifikasi tahapan induksi dan menghasilkan hasil yang tepat jika dilengkapi dengan sumber daya yang tepat. Hal ini juga yang menjadi alasan kami menggunakan aplikasi pemrograman Python yang mudah dipelajari, Fleksibel, memiliki Library yang kuat seperti NumPy, dan tentunya dapat diakses oleh semua orang.

Selain dari hal itu, dalam hal induksi matematika, penggunaan manual serta aplikasi pemrograman Python tidak berbeda. Python tidak diragukan lagi memiliki Efisiensi dan Kinerja yang lebih baik, terutama pada saat menangani seri atau perhitungan yang lebih kompleks dan fleksibel. Pembuktian manual memungkinkan kesalahan/ketidaktepatan dalam perhitungan maupun dalam proses induksi, sedangkan Python mengotomatiskan prosedur ini untuk mengurangi kesalahan.

Untuk saran yang dapat kami berikan yaitu mungkin untuk lebih dalam lagi memahami bagaimana pemanfaatan Python dalam memecahkan permasalahan lainnya, dapat menggunakan sumber lain untuk mendapatkan pemahaman tentang pemanfaatan pemecahan masalah menggunakan Python. Tetapi, alangkah lebih

baik jika menggunakan aplikasi pemrograman dasar lain seperti Free Pascal untuk mengenal lebih tentang bahasa program supaya saat menggunakan Python menjadi lebih maksimal. Karena *Library* yang kuat akan sia - sia apabila tidak mengenalnya lebih jauh. Selain daripada itu, kami mengharapkan penelitian ini dapat menjadi sumber referensi dalam memahami Python dalam pembuktian induksi matematika serta untuk permasalahan yang lebih jauh.

DAFTAR REFERENSI

- Bartle, R. G., & Sherbert, D. R. (2010). *Introduction to real analysis* (4th ed.). Urbana: John Wiley & Sons.
- Dristian, U., Qodar, A., Sulandra, I. M., & Anwar, L. (2023). Analisis proses pemahaman relasional mahasiswa ditinjau dari kesulitan dalam menyelesaikan soal induksi matematika. *Jurnal Cendekia: Jurnal Pendidikan Matematika*, 7(2), 1354-1365.
- Ernawati, E., & Ilhamuddin, I. (2020). Deskripsi kesalahan siswa dalam menyelesaikan soal matematika pokok bahasan induksi matematika. *Delta-Pi: Jurnal Matematika dan Pendidikan Matematika*, 9(2).
- Firmansyah, M., Masrun, & Yudha, I. D. K. (2021). Esensi perbedaan metode kualitatif dan kuantitatif. *Elastisitas: Jurnal Ekonomi Pembangunan*, 3(2), 156-159.
- Firmasari, S. (2019). *Journal of Medives*, 3(1).
- Firmasari, S., & Sulaiman, H. (2019). Kemampuan pembuktian matematis mahasiswa menggunakan induksi matematika. *Journal of Medives: Journal of Mathematics Education IKIP Veteran Semarang*, 3(1), 1-9.
- Gunawan, P. H. (2022). *Logika matematika untuk analisis algoritma*. Penerbit Andi.
- Kurniawan, R. I., Rosjanuardi, R., & Albania, I. N. (2022). Analisis kesulitan siswa dalam menyelesaikan soal induksi matematika. *AKSIOMA: Jurnal Program Studi Pendidikan Matematika*, 11(4), 3777-3789.
- Romzi, M., & Kurniawan, B. (2020). Pembelajaran pemrograman Python dengan pendekatan logika algoritma. *JTIM: Jurnal Teknik Informatika Mahakarya*, 3(2), 37-44.
- Thariq, A., & Apviano, D. (2020). Efek domino pada induksi matematika. *Journal on Mathematics Education Research*, 1(2), 60-68. <https://doi.org/10.17509/jmer.v1i2.14861>
- Ua, A. M. T. I., Lestari, D., Marpaung, E. S. K., Ong, J., Savinka, M., Nurhaliza, P., & Ningsih, R. Y. (2023). Penggunaan bahasa pemrograman Python dalam analisis faktor penyebab kanker paru-paru. *JUPTI: Jurnal Publikasi Teknik Informatika*, 2(2), 88-99.
- Utomo, D. P., & Huda, M. (2020). Pemahaman relasional analisis proses pembuktian menggunakan induksi matematika. *Bildung*, 1(1), 1-65.