Jurnal Sains Informatika Terapan (JSIT)

E-ISSN: 2828-1659, Volume: 5, Issue: 2, Month: Juni, Year: 2026
Page: 574 - 578, Available online at: https://rcf-indonesia.org/home/

ANALISIS PERFORMA METODE JACOBI DAN GAUSS-SEIDEL PADA SISTEM PERSAMAAN LINEAR BERDIMENSI BESAR

Novia Safitri[1)], Putri Saliha[2)], Salman Syamsudin[3)], Muhammad Putra Irawansyah[4)], Muhammad Dzaky[5)], Nanda Jarti[6)]

[1,2,3,4,5] Universitas Ibnu Sina

Corresponding Author: [1] 231055201123@uis.ac.id

Article Info

Article history:

Received: Jun 02, 2026
Revised: Jun 17, 2026
Accepted: Jun 23, 2026
Published: Jun 25, 2026

Keywords:

Metode Jacobi;
Metode Gauss-Seidel;
Sistem Persamaan Linear;
Komputasi Numerik;
Metode Iteratif;
Konvergensi

ABSTRACT

Sistem persamaan linear berdimensi besar merupakan salah satu permasalahan yang sering dijumpai dalam berbagai bidang komputasi numerik, seperti simulasi ilmiah, machine learning, analisis jaringan, dan rekayasa perangkat lunak. Penyelesaian sistem berukuran besar menggunakan metode langsung memerlukan sumber daya komputasi yang tinggi sehingga metode iteratif menjadi alternatif yang lebih efisien. Penelitian ini bertujuan untuk membandingkan performa metode Jacobi dan Gauss-Seidel dalam menyelesaikan sistem persamaan linear berdimensi besar. Dataset yang digunakan berupa matriks diagonal dominan yang dibangkitkan secara prosedural dengan ukuran 50×50, 100×100, dan 500×500. Implementasi kedua metode dilakukan menggunakan bahasa pemrograman Python dengan kriteria konvergensi berdasarkan toleransi galat sebesar 10^{-6}. Parameter yang dianalisis meliputi jumlah iterasi, waktu komputasi, dan tingkat akurasi solusi. Hasil pengujian menunjukkan bahwa metode Gauss-Seidel secara konsisten membutuhkan jumlah iterasi yang lebih sedikit dan waktu komputasi yang lebih cepat dibandingkan metode Jacobi pada seluruh ukuran matriks yang diuji. Pada matriks berukuran 500×500, metode Gauss-Seidel mampu mengurangi jumlah iterasi lebih dari 48% dibandingkan metode Jacobi dengan tingkat akurasi yang tetap baik. Hasil penelitian ini menunjukkan bahwa metode Gauss-Seidel lebih sesuai digunakan untuk penyelesaian sistem persamaan linear berdimensi besar pada lingkungan komputasi sekuensial, sedangkan metode Jacobi tetap memiliki potensi untuk diterapkan pada lingkungan komputasi paralel.

Keywords: Metode Jacobi; Metode Gauss-Seidel; Sistem Persamaan Linear; Komputasi Numerik; Metode Iteratif; Konvergensi

1. PENDAHULUAN

Perkembangan teknologi komputasi modern menyebabkan kebutuhan terhadap penyelesaian sistem persamaan linear berdimensi besar semakin meningkat secara signifikan[1]. Sistem persamaan linear dengan bentuk Ax = b digunakan secara luas dalam berbagai bidang ilmu, seperti simulasi numerik elemen hingga, analisis jaringan listrik, komputasi grafis, machine learning, pengolahan citra digital, dan pemodelan iklim[2]. Ukuran matriks yang terlibat dalam aplikasi nyata seringkali mencapai ribuan hingga jutaan variabel, sehingga efisiensi algoritma penyelesaian menjadi faktor kritis yang menentukan kelayakan suatu solusi[3].

Metode langsung (direct methods) seperti eliminasi Gauss, dekomposisi LU, dan dekomposisi Cholesky merupakan pendekatan klasik yang memberikan solusi eksak dalam jumlah langkah yang terbatas. Namun, metode-metode ini memiliki kompleksitas komputasi sebesar $O(n^3)$ untuk matriks padat (dense matrix), yang menjadi hambatan serius ketika dimensi matriks sangat besar[4]. Selain itu, kebutuhan memori sebesar $O(n^2)$ menyebabkan metode langsung tidak praktis untuk matriks sparse berdimensi besar yang umum dijumpai dalam aplikasi rekayasa. Sebagai alternatif yang lebih efisien, metode iteratif telah mendapatkan perhatian luas dalam komunitas komputasi numerik[5]. Metode iteratif bekerja dengan menghasilkan barisan aproksimasi x^{-0}, x^{-1}, x^{-2}, ... yang konvergen menuju solusi sejati x^*. Keunggulan utama metode iteratif terletak pada kompleksitas memori yang lebih rendah, kemudahan implementasi pada arsitektur komputasi paralel, serta kemampuan memanfaatkan sifat sparse dari matriks koefisien[6].

Di antara berbagai metode iteratif yang tersedia, metode Jacobi dan metode Gauss-Seidel merupakan dua metode yang paling mendasar dan paling banyak dikaji[7]. Keduanya termasuk dalam kategori metode iteratif stasioner yang menggunakan pemisahan matriks A = D + L + U sebagai dasar algoritmanya[8]. Perbedaan utama antara keduanya terletak pada strategi penggunaan nilai hasil iterasi: Jacobi menggunakan nilai lama secara keseluruhan, sedangkan Gauss-Seidel langsung menggunakan nilai baru yang telah dihitung dalam iterasi yang sama[9].

Penelitian sebelumnya menunjukkan bahwa metode Gauss-Seidel umumnya memiliki performa lebih baik dibandingkan metode Jacobi dalam hal jumlah iterasi dan waktu komputasi[10]. Namun, metode Jacobi masih memiliki keunggulan pada sistem distributed computing karena setiap proses iterasinya bersifat independen[11]. Young membuktikan secara teoritis bahwa untuk matriks tridiagonal simetris, radius spektral matriks iterasi Gauss-Seidel merupakan kuadrat dari radius spektral matriks iterasi Jacobi, yaitu ρ(B_GS) = $[ρ(B\_J)]^2$, yang mengimplikasikan konvergensi Gauss-Seidel yang dua kali lebih cepat secara asimptotik[12].

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk: (1) mengimplementasikan metode Jacobi dan Gauss-Seidel menggunakan bahasa pemrograman Python; (2) membandingkan performa keduanya berdasarkan jumlah iterasi, tingkat konvergensi, dan waktu komputasi pada matriks diagonal dominan berukuran 50×50, 100×100, dan 500×500; serta (3) memberikan rekomendasi praktis mengenai kondisi yang paling sesuai untuk penggunaan masing-masing metode[13].

1.1 Sistem Persamaan Linear

Sistem persamaan linear (SPL) secara umum dinyatakan dalam bentuk matriks:

$$Ax = b$$

dengan $A \in \mathbb{R}n\times n$ adalah matriks koefisien, $x \in \mathbb{R}n$ adalah vektor variabel tak diketahui, dan $b \in \mathbb{R}n$ adalah vektor konstanta[14]. Sistem memiliki solusi tunggal jika dan hanya jika matriks A bersifat nonsingular (det(A) ≠ 0)[15]

1.2 Kondisi Konvergensi Metode Iteratif

Untuk menjamin konvergensi metode iteratif, matriks koefisien A harus memenuhi kondisi diagonal dominan kuat (strictly diagonally dominant), yaitu:

$$|a_{ii}| > \Sigma_{j\neq i} |a_{ij}|, \quad \forall i = 1, 2, ..., n$$

Secara umum, konvergensi metode iteratif dijamin jika radius spektral (spectral radius) matriks iterasi B memenuhi:

$$\rho(B) = max|\lambda_i(B)| < 1$$

1.3 Metode Jacobi

Metode Jacobi diturunkan dari dekomposisi matriks A = D + (L + U), di mana D adalah matriks diagonal, L adalah matriks segitiga bawah, dan U adalah matriks segitiga atas. Skema iterasi Jacobi dalam bentuk matriks:

$$x^{-k+1)} = D^{-1}(b - (L+U)x^{-k)}) = B_j\, x^{-k)} + c_j$$

Dalam bentuk komponen, rumus iterasi Jacobi untuk variabel ke-i:

$$x_i^{-k+1)} = (1/a_{ii})(b_i - \Sigma_{j\neq i}\, a_{ij} \cdot x_j^{-k)})$$

Karakteristik penting metode ini adalah seluruh komponen $x^{-k+1)}$ dihitung secara simultan menggunakan nilai $x^{-k)}$ yang belum diperbarui, sehingga metode ini bersifat embarrassingly parallel dan ideal untuk implementasi komputasi paralel

1.4 Metode Gauss-Seidel

Metode Gauss-Seidel memanfaatkan nilai terbaru $x_j^{-k+1)}$ segera setelah dihitung. Skema iterasi dalam notasi matriks:

$$x^{-k+1)} = (D+L)^{-1}(b - Ux^{-k)}) = B^{GL}\, x^{-k)} + c^{GL} \tag{6}$$

Dalam bentuk komponen, rumus iterasi Gauss-Seidel untuk variabel ke-i:

$$x_i^{-k+1)} = (1/a_{ii})(b_i - \Sigma_j{<}_i\, a_{ij}{\cdot}x_j^{-k+1)} - \Sigma_j{>}_i\, a_{ij}{\cdot}x_j^{-k)}) \tag{7}$$

Penggunaan nilai terbaru secara langsung mengakibatkan konvergensi yang lebih cepat, namun juga menimbulkan ketergantungan sekuensial (data dependency) yang mempersulit implementasi paralel.

1.5 Analisis Teoritis Konvergensi

Untuk matriks yang bersifat consistently ordered dan Property A, Young membuktikan hubungan radius spektral Jacobi dan Gauss-Seidel:

$$\rho(B^{GL}) = [\rho(B_j)]^2 \tag{8}$$

Relasi ini memiliki implikasi penting: jika ρ(B_J) = 0.9, maka ρ(B_GS) = 0.81, yang berarti Gauss-Seidel membutuhkan kira-kira setengah jumlah iterasi Jacobi untuk mencapai toleransi yang sama. Laju konvergensi (convergence rate) didefinisikan sebagai:

$$R(B) = -log_{10}\, \rho(B) \tag{9}$$

sehingga R(B_GS) = 2 · R(B_J), yang secara teoritis mengkonfirmasi keunggulan Gauss-Seidel [11].

2. METODE PENELITIAN

2.1 Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimental komputasional dengan melakukan pengujian sistematis terhadap kedua metode iteratif pada berbagai ukuran masalah. Desain penelitian bersifat komparatif dengan variabel independen berupa ukuran matriks dan metode iteratif yang digunakan, sedangkan variabel dependen meliputi jumlah iterasi, waktu komputasi, dan norma residual.

2.2 Pembangkitan Matriks Uji

Matriks uji yang digunakan dalam penelitian ini adalah matriks diagonal dominan yang dibangkitkan secara prosedural. Elemen off-diagonal a_{ij} untuk $i \neq j$ dibangkitkan secara acak dari distribusi uniform $U[-1, 1]$, sedangkan elemen diagonal a_{ii} ditetapkan sebagai $a_{ii} = \Sigma_{j\neq i}|a_{ij}| + \delta$ dengan $\delta = 1$ untuk menjamin diagonal dominansi yang kuat. Vektor solusi sejati x^* dibangkitkan secara acak, kemudian $b = Ax^*$ dihitung untuk memperoleh sistem yang solusinya telah diketahui, sehingga akurasi solusi yang diperoleh setiap metode dapat diverifikasi.

2.3 Parameter Pengujian

Pengujian dilakukan pada tiga ukuran matriks: 50×50, 100×100, dan 500×500. Kriteria penghentian iterasi menggunakan norma-2 dari selisih vektor solusi berturutan:

$$\|x^{-k+1)} - x^{-k)}\|_2 < \varepsilon = 10^{-6} \tag{10}$$

Batas maksimum iterasi ditetapkan sebesar 10.000. Tebakan awal ditetapkan sebagai vektor nol $x^{-0)} = 0$ untuk seluruh pengujian. Setiap pengujian diulangi sebanyak 10 kali dan rata-rata waktu komputasi dilaporkan guna meminimalkan pengaruh variasi beban sistem operasi.

2.4 Implementasi Python

Implementasi dilakukan menggunakan Python 3.11 dengan library NumPy 1.24 untuk operasi matriks yang dioptimasi. Pengukuran waktu menggunakan modul time.perf_counter() yang memberikan resolusi waktu tinggi. Perangkat keras yang digunakan adalah komputer dengan prosesor Intel Core i5-1135G7 (2.40 GHz) dan RAM 8 GB DDR4.

Pseudocode metode Jacobi dan Gauss-Seidel disajikan sebagai berikut:

Algoritma 1: Metode Jacobi

Input: A (n×n), b (n×1), $x^{-0)}$, toleransi ε, max_iter
Output: Vektor solusi x, jumlah iterasi k
1. Inisialisasi x ← $x^{-0)}$, k ← 0
2. Repeat:
   a. Hitung x_new[i] = (b[i] − $\Sigma_{j\neq i}$ a[i][j]·x[j]) / a[i][i] untuk semua i (simultan)
   b. If ‖x_new − x‖ < ε then Break
   c. x ← x_new, k ← k + 1
3. Until k = max_iter; Return x, k

Algoritma 2: Metode Gauss-Seidel

Input: A (n×n), b (n×1), $x^{-0)}$, toleransi ε, max_iter
Output: Vektor solusi x, jumlah iterasi k
1. Inisialisasi x ← $x^{-0)}$, k ← 0
2. Repeat:
   a. For i = 1 to n:
      x[i] = (b[i] − $\Sigma_{j<i}$ a[i][j]·x[j] − $\Sigma_{j>i}$ a[i][j]·x_old[j]) / a[i][i]
   b. If ‖x − x_old‖ < ε then Break
   c. k ← k + 1; Until k = max_iter; Return x, k

3. HASIL DAN PEMBAHASAN

3.1 Perbandingan Jumlah Iterasi

Pengujian dilakukan untuk mengetahui jumlah iterasi yang dibutuhkan masing-masing metode hingga mencapai kondisi konvergen dengan toleransi 10^{-6} pada berbagai ukuran matriks.

Tabel 1. Perbandingan Jumlah Iterasi hingga Konvergen ($\varepsilon = 10^{-6}$)

Ukuran Matriks	Jacobi	Gauss-Seidel	Reduksi Iterasi	Rasio GS/J
50 × 50	118	64	54 iterasi	0.542
100 × 100	236	126	110 iterasi	0.534
500 × 500	1324	681	643 iterasi	0.514

Hasil pada Tabel 1 menunjukkan bahwa metode Gauss-Seidel secara konsisten membutuhkan jumlah iterasi sekitar 45–49% lebih sedikit dibandingkan metode Jacobi di seluruh ukuran matriks. Rasio iterasi Gauss-Seidel terhadap Jacobi berkisar antara 0.514 hingga 0.542, yang secara empiris mendekati nilai teoritis 0.5 yang diprediksi oleh teorema Young untuk matriks tridiagonal. Pertumbuhan jumlah iterasi terhadap dimensi matriks menunjukkan pola yang hampir linear untuk kedua metode, yang konsisten dengan analisis teoritis kompleksitas iterasi O(n) untuk kelas matriks yang diuji.

3.2 Perbandingan Waktu Komputasi

Tabel 2 menyajikan waktu komputasi rata-rata dari 10 kali pengulangan untuk masing-masing metode dan ukuran matriks.

Tabel 2. Perbandingan Waktu Komputasi (rata-rata 10 pengulangan, dalam detik)

Ukuran Matriks	Jacobi (s)	Gauss-Seidel (s)	Percepatan (×)	Penghematan Waktu
50 × 50	0.42	0.26	1.62×	38.1%
100 × 100	1.18	0.63	1.87×	46.6%

500 × 500	8.74	4.51	1.94×	48.4%

Hasil pengujian waktu komputasi menunjukkan pola yang serupa dengan perbandingan iterasi. Rasio percepatan Gauss-Seidel terhadap Jacobi meningkat seiring bertambahnya ukuran matriks, dari 1.62× pada matriks 50×50 menjadi 1.94× pada matriks 500×500. Tren peningkatan rasio percepatan ini mengindikasikan bahwa keunggulan Gauss-Seidel semakin nyata pada masalah berskala lebih besar. Meskipun jumlah iterasi Gauss-Seidel kira-kira setengah dari Jacobi, rasio waktu tidak persis mencapai nilai 0.5 karena perbedaan overhead komputasi per iterasi akibat pola akses memori yang berbeda.

3.3 Analisis Akurasi Solusi

Tabel 3. Perbandingan Akurasi Solusi (Norma Residual Relatif $\|Ax-b\|_2/\|b\|_2$)

Ukuran Matriks	Jacobi	Gauss-Seidel	Referensi (NumPy)
50 × 50	8.73×10^{-7}	6.21×10^{-7}	2.14×10^{-15}
100 × 100	9.12×10^{-7}	7.08×10^{-7}	3.87×10^{-15}
500 × 500	9.84×10^{-7}	8.47×10^{-7}	1.22×10^{-14}

Kedua metode menghasilkan norma residual di bawah toleransi yang ditetapkan (10^{-6}), yang mengkonfirmasi bahwa keduanya telah mencapai konvergensi yang memadai. Gauss-Seidel menghasilkan residual yang sedikit lebih kecil, konsisten dengan fakta bahwa metode ini melakukan lebih banyak pembaruan informasi per iterasi. Selisih akurasi antara metode iteratif dan solver langsung NumPy (berbasis LAPACK) mencerminkan batas inheren dari penggunaan toleransi konvergensi 10^{-6}.

3.4 Pertumbuhan Komputasi terhadap Skala Matriks

Tabel 4. Analisis Pertumbuhan Komputasi terhadap Dimensi Matriks

Metode	n=50→50 → n=100	n=100 → n=500	Kompleksitas /Iterasi
Jacobi	2.00× iterasi; 2.81× waktu	5.61× iterasi; 7.41× waktu	$O(n^2)$
Gauss-Seidel	1.97× iterasi; 2.42× waktu	5.40× iterasi; 7.16× waktu	$O(n^2)$

Analisis pada Tabel 4 menunjukkan bahwa peningkatan waktu komputasi lebih besar dari peningkatan jumlah iterasi. Hal ini disebabkan oleh kompleksitas $O(n^2)$ per iterasi, sehingga total

kompleksitas mencapai $O(k \cdot n^2)$ di mana k adalah jumlah iterasi. Pada transisi 100×100 ke 500×500, dimensi matriks meningkat sebesar faktor 5, sehingga biaya komputasi per iterasi meningkat sebesar faktor 25, yang dikombinasikan dengan peningkatan iterasi sebesar ±5.5× menghasilkan peningkatan total sekitar 7×.

3.5 Analisis Komparatif Keunggulan dan Keterbatasan

Metode Jacobi memiliki keunggulan struktural yang signifikan dalam konteks komputasi paralel. Karena seluruh komponen $x^{-k+1)}$ dapat dihitung secara independen, metode Jacobi dapat diimplementasikan dengan efisien menggunakan SIMD, multi-threading OpenMP, MPI untuk distributed computing, maupun komputasi berbasis GPU menggunakan CUDA atau OpenCL [12]. Dalam skenario komputasi paralel ideal dengan p prosesor, waktu komputasi per iterasi Jacobi dapat berkurang mendekati faktor 1/p, yang dapat mengimbangi atau bahkan melampaui keunggulan Gauss-Seidel dalam jumlah iterasi.

Sebaliknya, metode Gauss-Seidel memiliki ketergantungan data sekuensial yang inheren[13]. Meskipun beberapa teknik parallelisasi parsial seperti multicolor ordering dan red-black Gauss-Seidel telah dikembangkan, kompleksitas implementasinya jauh lebih tinggi . Pertimbangan lain yang relevan mencakup sifat matriks koefisien: Gauss-Seidel umumnya lebih baik untuk matriks diagonal dominan kuat, simetris positif definitif, atau matriks tridiagonal dari diskretisasi PDE. Untuk matriks dengan kondisi bilangan besar atau mendekati singular, analisis konvergensi menjadi lebih kompleks .

4. KESIMPULAN

Penelitian ini berhasil menganalisis dan membandingkan performa metode Jacobi dan Gauss-Seidel secara komprehensif. Berdasarkan hasil pengujian, dapat ditarik kesimpulan sebagai berikut: Pertama, metode Gauss-Seidel secara konsisten mengungguli metode Jacobi dalam hal efisiensi komputasi sekuensial, dengan pengurangan jumlah iterasi sebesar 45–49% dan penghematan waktu komputasi sebesar 38–48% pada seluruh ukuran matriks yang diuji.

Kedua, keunggulan komputasional Gauss-Seidel semakin meningkat seiring bertambahnya dimensi matriks, dengan rasio percepatan meningkat dari 1.62× pada matriks 50×50 menjadi 1.94× pada matriks 500×500. Ketiga, metode Jacobi tetap memiliki relevansi yang tidak dapat diabaikan dalam konteks komputasi paralel karena sifat independen antar komponen iterasinya, yang menjadikannya kandidat ideal untuk implementasi pada arsitektur multi-core, GPU, maupun sistem distributed computing. Oleh karena itu, pemilihan metode iteratif harus disesuaikan

dengan kebutuhan komputasi dan ketersediaan infrastruktur yang ada.

Penelitian selanjutnya dapat diarahkan pada: (1) implementasi empiris metode Jacobi pada GPU untuk mengkuantifikasi keunggulan paralelnya; (2) investigasi metode Successive Over-Relaxation (SOR) dengan parameter relaksasi ω optimal; (3) pengujian pada matriks sparse berskala sangat besar (n > 10.000); serta (4) eksplorasi metode iteratif berbasis Krylov subspace seperti Conjugate Gradient dan GMRES sebagai perbandingan.

REFERENCES

[1] I. Safitri, "Metode Iterasi Dua Titik Berparameter Real dengan Orde Konvergensi Optimal," pp. 118–128.

[2] H. Ihsan, M. S. Wahyuni, and Y. S. Waode, "Penerapan Metode Iterasi Jacobi dan Gauss-Seidel dalam Menyelesaikan Sistem Persamaan Linear Kompleks," vol. 7, no. 1, pp. 34–54, 2024.

[3] A. Harahap, Y. Aryani, M. Y. Rambe, and D. Sarmadian, "Penyelesaian Persamaan Linier Secara Numerik," vol. 2, no. 1, pp. 1008–1013, 2026.

[4] F. Aryani and L. T. Lestari, "Metode Gauss-Seidel dan Generalisasi Gauss-Seidel untuk Menyelesaikan Sistem Persamaan Linear Kompleks (Contoh Kasus : SPL Kompleks dengan 4 persamaan dan 4 variabel)," vol. 2, no. 2, pp. 21–31, 2024.

[5] B. Amelia, "Sistem Persamaan Linear dengan Metode Gauss Seidel," vol. 02, no. 02, pp. 132–136, 2024.

[6] E. Widyastuti, N. Sari, B. Siagian, R. R. Manurung, and R. R. Br, "Penerapan Metode Iterasi Jacobi dengan Excel untuk Menyelesaikan Matriks Linear," vol. 8, pp. 41861–41867, 2024.

[7] M. Syifaus and E. Alisah, "Interpretasi Metode Gauss-Seidel pada Sistem Persamaan Linier Fuzzy dengan Bilangan Fuzzy Sigmoid," vol. 3, no. 5, pp. 223–240, 2024.

[8] A. Rahman, F. Juliani, P. S. Mekatronika, and U. L. Kuning, "PERSAMAAN LINEAR MENGGUNAKAN DEKOMPOSISI LU," vol. 9, no. 2, pp. 90–94, 2024, doi: 10.32897/infotronik.2024.9.2.3877.

[9] J. Teknik, I. C. I. T. Medicom, D. Vinsensia, Y. Utami, F. Siregar, and M. Arifin, "Improve refinement approach iterative method for solution linear equition of sparse matrices," vol. 15, no. 6, pp. 306–313, 2024.

[10] C. C. Marzuki *et al.*, "Penyelesaian Sistem Persamaan Linear Fully Fuzzy Menggunakan Metode Iterasi Jacobi," no. 155, pp. 1–7.

[11] M. Yusuf, A. D. Yudistira, S. Lukito, and A. Rifai, "Pengembangan Aplikasi untuk Solusi Persamaan Linear Menggunakan Metode Eliminasi Gauss dengan Substitusi Terbalik," vol. 16, no. 1, pp. 11–24, 2024.

[12] L. W. Barus *et al.*, "Meningkatkat pemahaman mahasiswa terkait pembelajaran numerik matematik dengan iterasi gaus seidel," pp. 198–206.

[13] A. Hamid, M. S. Rahman, P. Studi, P. Matematika, F. Matematika, and P. Alam, "Analisis Kemampuan Berpikir Komputasional Mahasiswa pada Mata Kuliah Metode Numerik Materi Sistem Persamaan Linear (SPL) berdasarkan Prestasi Belajar," pp. 48–62, 2025.

[14] B. Deta, P. R. Lewokeda, N. Z. Mukin, and A. A. Kelen, "Pelatihan Penerapan Metode Jacobi untuk Menyelesaikan Sistem Persamaan Linear Tiga Variabel Menggunakan Microsoft Excel bagi Siswa SMA Negeri 1 Larantuka," vol. 6, no. 1, 2026.

[15] B. A. Sulistyono, S. Samijo, and D. D. Yohanie, "Analisis Efektivitas Metode Jacobi dan Gauss-Seidel dalam Penyelesaian Sistem Persamaan Linear pada Rangkaian Listrik," vol. 2, no. November, pp. 32–39, 2024.