SAINSTECH
Jurnal Penelitian dan Pengkajian Sains dan Teknologi
ISTN
Volume 36 No.1
Maret 2026

Perbandingan Efisiensi Algoritma, Bubble Sort, Merge Sort, dan Quick Sort dalam Pengolahan Data Konsumsi Listrik Rumah Tangga

Akbar Alif Haikal[1], Triwanti Andini Hutasoit[2], Claudia Agatha Br. Tarigan[3] Jonathan Rio Gultom[4] , Adidtya Perdana[5]

[1,2,3,4,5]Universitas Negeri Medan, Jalan Willem Iskandar, Pasar V Medan Estate, Kecamatan Percut Sei Tuan, Kabupaten Deli Serdang, Sumatera Utara, 20221.
Program Studi S1 Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam
e-mail: [1]akbaralifhaikal@gmail.com, [2]triwantiandinihutasoit@gmail.com, [3]claudiaagatha5@gmail.com, [4]jonathanriogultom@gmail.com, [5]adidtya@unimed.ac.id

Abstrak

Proses pengurutan data numerik merupakan tahapan penting dalam sistem analisis data, termasuk dalam pengelolaan informasi konsumsi listrik rumah tangga. Setiap algoritma sorting memiliki karakteristik kompleksitas waktu yang berbeda, sehingga dapat memengaruhi performa komputasi pada berbagai ukuran dataset. Penelitian ini bertujuan untuk membandingkan efisiensi algoritma Bubble Sort, Merge Sort, dan Quick Sort dalam mengolah data konsumsi listrik rumah tangga. Penelitian dilakukan menggunakan pendekatan eksperimen kuantitatif dengan mengimplementasikan ketiga algoritma menggunakan bahasa pemrograman Python. Dataset yang digunakan berupa data numerik dengan variasi ukuran 50, 500, dan 5000 data. Parameter utama yang diukur adalah waktu eksekusi masing-masing algoritma. Hasil penelitian menunjukkan bahwa Bubble Sort memiliki waktu eksekusi paling tinggi, terutama pada dataset besar, dengan waktu mencapai 2.525164 detik pada 5000 data. Sebaliknya, Quick Sort menunjukkan performa terbaik dengan waktu tercepat sebesar 0.004337 detik, diikuti oleh Merge Sort sebesar 0.012139 detik. Temuan ini sejalan dengan teori kompleksitas waktu, di mana Bubble Sort memiliki kompleksitas $O(n^2)$, sedangkan Merge Sort dan Quick Sort memiliki kompleksitas rata-rata O(n log n). Penelitian ini menegaskan bahwa pemilihan algoritma yang tepat sangat berpengaruh terhadap efisiensi pengolahan data, khususnya pada dataset berukuran besar dalam konteks pengolahan data energi.

Kata Kunci: algoritma sorting, Bubble Sort, Merge Sort, Quick Sort, kompleksitas waktu, notasi Big-O

Abstract

The process of numerical data sorting is a fundamental stage in data analysis systems, including in the management of household electricity consumption data. Each sorting algorithm has different time complexity characteristics, which can significantly affect computational performance when applied to datasets of varying sizes. This study aims to compare the efficiency of Bubble Sort, Merge Sort, and Quick Sort in processing household electricity consumption data. This research employs a quantitative experimental approach by implementing the three algorithms using the Python programming language. The dataset consists of numerical values with varying sizes of 50, 500, and 5000 data points. The main parameter measured is the execution time of each algorithm. The results show that Bubble Sort has the highest execution time, especially for larger datasets, reaching 2.525164 seconds for 5000 data points. In contrast, Quick Sort demonstrates the best performance with the fastest execution time of 0.004337 seconds, followed by Merge Sort at 0.012139 seconds. These findings are consistent with theoretical time complexity, where Bubble Sort has a complexity of $O(n^2)$, while Merge Sort and Quick Sort have an average complexity of O(n log n). This study confirms that selecting an appropriate sorting algorithm significantly affects data processing efficiency, particularly for large-scale datasets in the context of energy data management.

Keywords: *sorting algorithm, Bubble Sort, Merge Sort, Quick Sort, time complexity, Big-O notation*

DOI : https://doi.org/10.37277/stch.v36i1.2622

1. Pendahuluan

Perkembangan teknologi informasi telah mendorong peningkatan kebutuhan terhadap pengolahan data yang cepat dan efisien dalam berbagai bidang komputasi. Salah satu proses penting dalam pengolahan data adalah proses pengurutan data (sorting), yaitu proses pengaturan data berdasarkan aturan tertentu sehingga data dapat tersusun secara sistematis dan mudah dianalisis (Mahrozi dan Faisal, 2023). Proses pengurutan menjadi bagian penting dalam berbagai sistem komputasi karena data yang terstruktur akan mempermudah proses pencarian, pengolahan, serta pengambilan keputusan berbasis data.

Algoritma pengurutan telah menjadi topik penelitian yang luas selama beberapa dekade karena kebutuhan akan proses pengurutan data di berbagai domain aplikasi komputasi (Liu, 2024). Seiring dengan meningkatnya volume data digital, kebutuhan terhadap teknik pengolahan data yang efisien juga semakin meningkat. Data yang tidak terorganisir dengan baik dapat menyulitkan proses pencarian serta verifikasi informasi sehingga menurunkan efektivitas pengolahan data dalam suatu sistem (Pujiono et al.,2025).

Pertumbuhan data digital yang sangat pesat juga menuntut algoritma pengurutan yang mampu bekerja secara efisien pada skala data yang besar. Kehadiran big data telah menciptakan volume data yang sangat besar sehingga membutuhkan algoritma pengurutan yang efisien serta dapat diskalakan untuk berbagai kebutuhan komputasi modern (AlDammagh dan Abu-Naser, 2025).Teknik pengurutan yang efisien menjadi semakin penting terutama pada sistem komputasi yang membutuhkan pemrosesan data secara cepat dan akurat (Sundaramoorthy dan Karunanidhi, 2025).

Dalam ilmu komputer terdapat berbagai algoritma sorting yang dapat digunakan untuk mengurutkan data, seperti Bubble Sort, Insertion Sort, Selection Sort, Shell Sort, Quick Sort, Merge Sort, dan Heap Sort (Pujiono et al., 2024). Setiap algoritma memiliki karakteristik serta tingkat efisiensi yang berbeda dalam proses pengolahan data. Oleh karena itu, pemilihan algoritma yang tepat menjadi faktor penting dalam meningkatkan performa sistem komputasi.

Kinerja suatu algoritma pengurutan umumnya dianalisis menggunakan konsep kompleksitas waktu (time complexity), yaitu ukuran yang digunakan untuk mengevaluasi hubungan antara pertambahan jumlah data dengan waktu eksekusi algoritma (Riyadi et al., 2022). Analisis kompleksitas waktu memberikan gambaran mengenai performa algoritma dalam berbagai kondisi eksekusi sehingga dapat digunakan sebagai dasar dalam menentukan efisiensi suatu algoritma (Khaleel, 2025). Meskipun berbagai penelitian telah membahas perbandingan algoritma pengurutan, sebagian besar penelitian tersebut berfokus pada analisis umum algoritma tanpa mengaitkannya dengan konteks data tertentu. Oleh karena itu, penelitian ini dilakukan untuk membandingkan efisiensi algoritma Bubble Sort, Merge Sort, dan Quick Sort melalui pengujian waktu eksekusi pada data konsumsi listrik rumah tangga menggunakan bahasa pemrograman Python.

Berdasarkan uraian latar belakang tersebut, penelitian ini berfokus pada permasalahan mengenai bagaimana perbandingan kinerja algoritma Bubble Sort, Merge Sort, dan Quick Sort dalam mengolah data konsumsi listrik rumah tangga pada berbagai ukuran dataset. Selain itu, penelitian ini juga mengkaji algoritma yang paling efisien berdasarkan waktu eksekusi, serta menganalisis kesesuaian antara hasil pengujian empiris dengan teori kompleksitas waktu menggunakan notasi Big-O.

1.1 Tinjauan Pustaka

Algoritma dan Pengurutan Data

Algoritma merupakan suatu prosedur logis yang digunakan untuk menyelesaikan permasalahan secara sistematis dalam proses komputasi. Perancangan algoritma yang tidak tepat dapat menyebabkan proses eksekusi program menjadi lambat serta membutuhkan penggunaan memori yang lebih besar (Iskandar et al., 2023).Oleh karena itu, pemilihan algoritma yang efisien menjadi salah satu aspek penting dalam pengembangan sistem perangkat lunak.

Algoritma Bubble Sort

Bubble Sort merupakan algoritma pengurutan sederhana yang bekerja dengan cara membandingkan elemen-elemen yang bersebelahan dan menukar posisinya apabila urutan tidak sesuai sehingga elemen terbesar secara bertahap berpindah ke posisi akhir daftar (Sena et al., 2024). Algoritma ini termasuk ke dalam kategori comparison sort karena proses pengurutannya menggunakan operasi perbandingan antar elemen (Panggabean et al., 2023).

Beberapa penelitian menunjukkan bahwa performa Bubble Sort dapat ditingkatkan melalui optimasi seperti mekanisme early termination yang menghentikan proses iterasi ketika data telah berada dalam kondisi terurut (Ramadhan dan Widiono, 2025).Meskipun demikian, algoritma ini memiliki keterbatasan ketika digunakan pada dataset berukuran besar karena jumlah operasi perbandingan yang relatif tinggi (Rizky et al., 2025).

Algoritma Bubble Sort

Merge Sort merupakan algoritma pengurutan yang menggunakan pendekatan divide and conquer dengan membagi array menjadi beberapa bagian yang lebih kecil, kemudian mengurutkan masing-masing bagian secara rekursif sebelum digabungkan kembali menjadi satu array yang terurut (Riki et al., 2024). Pendekatan ini memungkinkan algoritma bekerja secara sistematis dalam menangani data berukuran besar. Namun demikian, algoritma ini memerlukan penggunaan memori tambahan selama proses penggabungan data karena membutuhkan array sementara (Saputra et al., 2025).

Algoritma Quick Sort

Quick Sort merupakan salah satu algoritma pengurutan yang banyak digunakan dalam sistem komputasi karena memiliki performa yang baik dalam menangani dataset berukuran besar (Nasution et al., 2024). Beberapa penelitian juga menunjukkan bahwa Quick Sort memiliki efisiensi komputasi yang tinggi, di mana pengujian terhadap dataset berjumlah 250 data menunjukkan waktu pemrosesan yang relatif cepat (Ali et al., 2025).

Kompleksitas Waktu dan Notasi Big-O

Notasi Big-O merupakan metode yang digunakan untuk menyatakan laju pertumbuhan suatu fungsi berdasarkan ukuran input yang diberikan (Tanjung et al., 2023). Notasi ini digunakan untuk menganalisis efisiensi algoritma dari segi waktu maupun penggunaan ruang memori dalam proses komputasi (Cibro, 2024). Selain itu, notasi Big-O juga digunakan untuk memahami perilaku algoritma secara asimtotik sehingga performa algoritma dapat dianalisis ketika ukuran input menjadi sangat besar (Ilmannafia et al., 2025).

2. Metode Penelitian

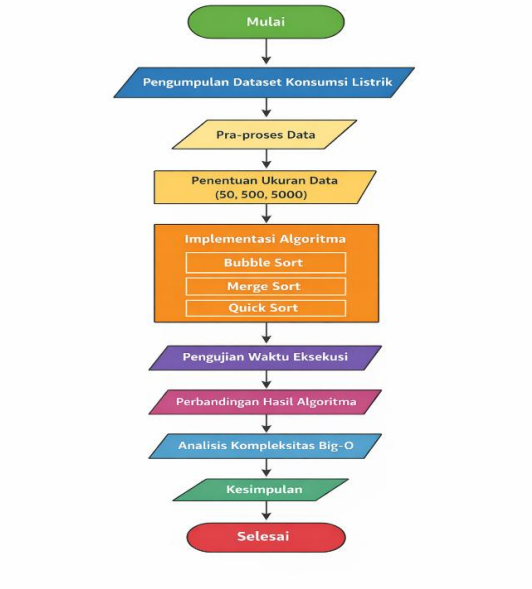
2.1 Rancangan Penelitian

Penelitian ini menggunakan pendekatan eksperimen komputasi dengan metode kuantitatif untuk menganalisis performa algoritma pengurutan data. Eksperimen dilakukan dengan mengimplementasikan tiga algoritma sorting, yaitu Bubble Sort, Merge Sort, dan Quick Sort menggunakan bahasa pemrograman Python.

Setiap algoritma dijalankan pada dataset yang sama sehingga perbedaan performa dapat diamati secara objektif. Fokus utama penelitian adalah mengukur dan membandingkan waktu eksekusi yang dibutuhkan oleh masing-masing algoritma dalam menyelesaikan proses pengurutan

DOI : https://doi.org/10.37277/stch.v36i1.2622

data. Alur tahapan penelitian yang dilakukan dapat dilihat pada



Gambar 1. Alur Metode Penelitian

2.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini berupa data numerik yang merepresentasikan konsumsi listrik rumah tangga. Dataset diperoleh dari sumber data terbuka yang tersedia.

Data konsumsi listrik yang tersedia dalam dataset tersebut kemudian diproses kembali sehingga menghasilkan data numerik yang dapat digunakan sebagai input dalam proses pengujian algoritma pengurutan.

Proses pengumpulan data dilakukan dengan mengunduh dataset dari Kaggle, kemudian memilih data yang relevan dengan kebutuhan penelitian. Selanjutnya, data diproses untuk menghasilkan nilai numerik yang akan digunakan dalam proses eksperimen.

Untuk melihat pengaruh jumlah data terhadap performa algoritma, dataset dibagi ke dalam beberapa variasi ukuran, yaitu 50 data, 500 data, dan 5000 data. Variasi ukuran dataset ini digunakan untuk mengamati bagaimana perubahan jumlah data memengaruhi waktu eksekusi algoritma sorting.

2.3 Analisis Data

2.1.1 Tabel dan Gambar

Analisis data dilakukan dengan mengimplementasikan algoritma Bubble Sort, Merge Sort, dan Quick Sort menggunakan bahasa pemrograman Python. Seluruh proses eksperimen dijalankan pada platform Google Colab yang memungkinkan eksekusi kode Python secara daring.
Setiap algoritma dijalankan pada dataset dengan ukuran yang telah ditentukan, kemudian waktu eksekusinya diukur menggunakan fungsi pengukuran waktu pada Python. Hasil pengukuran waktu komputasi tersebut kemudian dibandingkan untuk mengetahui algoritma yang memiliki performa paling efisien.

Hasil eksperimen selanjutnya dianalisis dengan mengaitkan waktu eksekusi yang diperoleh dengan teori kompleksitas waktu algoritma menggunakan notasi Big-O. Analisis ini dilakukan untuk melihat kesesuaian antara teori kompleksitas algoritma dengan performa implementasinya dalam pengolahan data konsumsi listrik rumah tangga.

Tabel 1. Skenario Pengujian Algoritma

No	Algoritma	Ukuran Dataset	Parameter yang Diukur	Platform
1	Bubble Sort	50	Waktu esksekusi	Google Colab
2	Bubble Sort	500	Waktu esksekusi	Google Colab
3	Bubble Sort	5000	Waktu esksekusi	Google Colab
4	Merge Sort	50	Waktu esksekusi	Google Colab
5	Merge Sort	500	Waktu esksekusi	Google Colab
6	Merge Sort	5000	Waktu esksekusi	Google Colab
7	Quick Sort	50	Waktu esksekusi	Google Colab
8	Quick Sort	500	Waktu esksekusi	Google Colab
9	Quick Sort	5000	Waktu esksekusi	Google Colab

Data yang digunakan dalam penelitian ini berupa data konsumsi listrik rumah tangga yang diperoleh dari dataset terbuka yang tersedia pada platform

Kaggle, yaitu "Individual Household Electric Power Consumption Dataset". Dataset ini berisi informasi konsumsi listrik rumah tangga dalam satuan waktu tertentu yang kemudian diproses menjadi data numerik untuk kebutuhan pengujian algoritma sorting.

Dataset diunduh melalui laman: https://www.kaggle.com/datasets/uciml/electric-power-consumption-data

Pada tahap praproses, data dipilih dan dibersihkan untuk menghilangkan nilai yang tidak valid, kemudian diambil atribut numerik yang relevan untuk proses pengurutan. Data tersebut selanjutnya dibagi ke dalam beberapa variasi ukuran, yaitu 50, 500, dan 5000 data untuk menguji pengaruh skala dataset terhadap waktu eksekusi algoritma.

Seluruh proses eksperimen dilakukan menggunakan platform Google Colab dengan spesifikasi lingkungan komputasi berupa prosesor virtual berbasis Intel Xeon dengan kecepatan sekitar 2.2 GHz, RAM sebesar 12 GB, serta sistem operasi berbasis Linux. Implementasi algoritma dilakukan menggunakan bahasa pemrograman Python versi 3.x dengan bantuan pustaka standar untuk pengukuran waktu eksekusi.

3. Hasil dan Pembahasan

3.1 Hasil Pengujian Waktu Eksekusi Algoritma

Pengujian dalam penelitian ini dilakukan untuk mengetahui dan membandingkan performa tiga algoritma pengurutan, yaitu Bubble Sort, Merge Sort, dan Quick Sort. Pengujian dilakukan menggunakan dataset numerik yang merepresentasikan konsumsi listrik rumah tangga dengan tiga variasi ukuran data, yaitu 50 data, 500 data, dan 5000 data. Parameter utama yang diamati dalam eksperimen ini adalah waktu eksekusi yang dibutuhkan oleh setiap algoritma dalam menyelesaikan proses pengurutan.

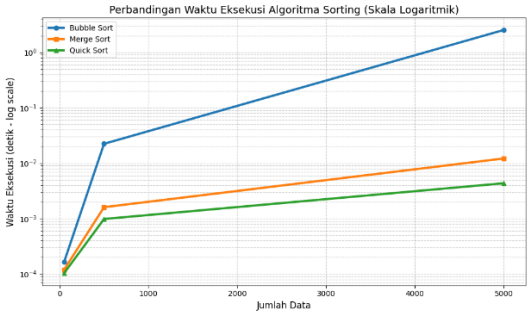
Hasil pengujian waktu eksekusi masing-masing algoritma ditampilkan pada Tabel 2.

Tabel 2. Hasil Pengujian Waktu Eksekusi Algoritma Sorting

Algoritma	50 Data	500 Data	5000 Data
Bubble Sort	0.000164	0.022347	2.525164
Merge Sort	0.000119	0.0001605	0.012139
Quick Sort	0.000103	0.000983	0.004337

Berdasarkan hasil pada Tabel 2, terlihat bahwa waktu eksekusi algoritma cenderung meningkat seiring dengan bertambahnya jumlah data yang diproses. Namun demikian, tingkat peningkatan waktu komputasi pada setiap algoritma menunjukkan pola yang berbeda.

Untuk memberikan gambaran yang lebih jelas mengenai perbandingan performa ketiga algoritma tersebut, hasil pengujian juga divisualisasikan dalam bentuk grafik pada gambar 2.



Gambar 1. Judul Gambar Posisi di Sini

algoritma Bubble Sort, Merge Sort, dan Quick Sort pada berbagai ukuran dataset menggunakan skala logaritmik.

3.2 Analisis Perbandingan Algoritma

Berdasarkan hasil pengujian yang telah dilakukan, terlihat bahwa performa ketiga algoritma menunjukkan perbedaan yang signifikan seiring dengan meningkatnya

ukuran dataset. Pada dataset kecil (50 data), perbedaan waktu eksekusi antar algoritma relatif tidak signifikan. Namun, ketika ukuran dataset meningkat menjadi 500 dan 5000 data, perbedaan performa menjadi semakin jelas.

Algoritma Bubble Sort menunjukkan peningkatan waktu eksekusi yang sangat signifikan. Hal ini disebabkan oleh mekanisme kerjanya yang melakukan perbandingan secara berulang untuk setiap pasangan elemen, sehingga jumlah operasi yang dilakukan meningkat secara kuadratik terhadap ukuran data. Akibatnya, algoritma ini menjadi tidak efisien untuk dataset berukuran besar.

Sebaliknya, Merge Sort dan Quick Sort menunjukkan performa yang jauh lebih stabil. Merge Sort bekerja dengan pendekatan divide and conquer, di mana data dibagi menjadi bagian-bagian kecil, kemudian digabungkan kembali dalam kondisi terurut. Pendekatan ini membuat jumlah operasi lebih terkontrol dan tidak meningkat secara drastis meskipun ukuran data bertambah.

Sementara itu, Quick Sort menunjukkan performa paling cepat pada hasil pengujian. Secara empiris, hal ini disebabkan oleh mekanisme pemilihan pivot yang memungkinkan proses partisi data berlangsung secara efisien. Pada dataset yang digunakan dalam penelitian ini, distribusi data memungkinkan pembagian yang relatif seimbang, sehingga kedalaman rekursi tetap optimal. Kondisi ini menyebabkan jumlah operasi yang dilakukan lebih sedikit dibandingkan Merge Sort, terutama karena Quick Sort tidak memerlukan proses penggabungan array tambahan.

3.3 Analisis Berdasarkan Kompleksitas Big-O

Jika dikaitkan dengan rumusan masalah penelitian, hasil pengujian ini menjawab bahwa algoritma dengan kompleksitas waktu yang lebih rendah cenderung memiliki performa yang lebih baik pada dataset berukuran besar. Bubble Sort dengan kompleksitas $O(n^2)$ menunjukkan peningkatan waktu eksekusi yang sangat signifikan, sedangkan Merge Sort dan Quick Sort dengan kompleksitas rata-rata $O(n \log n)$ menunjukkan peningkatan yang lebih stabil.

Kesesuaian antara hasil empiris dan teori terlihat jelas pada pola pertumbuhan waktu eksekusi. Ketika jumlah data meningkat dari 50 menjadi 5000, waktu eksekusi Bubble Sort meningkat secara drastis, sedangkan peningkatan pada Merge Sort dan Quick Sort relatif lebih kecil. Hal ini menunjukkan bahwa analisis kompleksitas waktu menggunakan notasi Big-O dapat digunakan sebagai prediktor yang cukup akurat terhadap performa algoritma dalam implementasi nyata.

Selain itu, hasil penelitian ini juga menjawab bahwa Quick Sort merupakan algoritma yang paling efisien dalam konteks dataset yang digunakan. Namun demikian, perlu dicatat bahwa performa Quick Sort sangat bergantung pada pemilihan pivot. Pada kondisi terburuk, algoritma ini dapat memiliki kompleksitas $O(n^2)$, meskipun dalam praktiknya kasus tersebut jarang terjadi, terutama pada data yang tidak terurut secara ekstrem.

4. Kesimpulan

Berdasarkan hasil penelitian mengenai perbandingan algoritma Bubble Sort, Merge Sort, dan Quick Sort dalam proses pengurutan data konsumsi listrik rumah tangga, dapat disimpulkan sebagai berikut:

1. Waktu eksekusi algoritma cenderung meningkat seiring dengan bertambahnya jumlah data yang diproses. Hal ini menunjukkan bahwa ukuran dataset memiliki pengaruh terhadap kinerja algoritma pengurutan.

2. Bubble Sort memiliki waktu eksekusi yang paling besar dibandingkan algoritma lainnya, terutama ketika jumlah data semakin banyak, sehingga kurang efisien untuk digunakan pada dataset berukuran besar.
3. Merge Sort dan Quick Sort menunjukkan kinerja yang lebih baik dengan waktu komputasi yang lebih cepat. Hal ini menunjukkan bahwa kedua algoritma tersebut lebih sesuai digunakan untuk proses pengurutan data dalam jumlah yang lebih besar.
4. Penelitian selanjutnya dapat mengembangkan analisis dengan menggunakan variasi dataset yang lebih beragam serta membandingkan algoritma sorting lainnya untuk memperoleh gambaran performa yang lebih luas.

Daftar Pustaka

AlDammagh, A. K., & Abu-Naser, S. S. (2025). "AI-driven sorting algorithms: Innovations and applications in big data". International Journal of Academic Engineering Research, 9(6), 1–10.

Ali, M. I., Fardiarsyah, R. D., Shodik, L., & Dwi, F. Z. (2025)."Analisis komparatif efisiensi memori dan waktu komputasi pada 8 algoritma sorting menggunakan C++." 2(1), 1–17.

Cibro, E. (2024). "Mengidentifikasi dasar algoritma pemrograman. Jurnal Ilmiah Sistem Informasi dan Ilmu Komputer", 4(1), 194–206.

Ilmannafia, A., Yudianti, A. B., & Aini, F. N. (2025). Evaluation theoretical efficiency selection sort by Big-O notation analysis. Journal of Advanced Systems Intelligence and Cybersecurity, 1(1), 1–11.

Iskandar, J., Suhendar, H., & Pamungkas, B. D. (2023). Analisis strategi algoritma sorting menggunakan metode komparatif pada bahasa pemrograman Java dengan Python. G-Tech: Jurnal Teknologi Terapan, 8(1), 104–113.

Khaleel, S. (2025). Comprehensive analysis of time and space complexity in algorithm design. 1(1), 1–5.

Liu, P. (2024). An in-depth study of sorting algorithms. Applied and Computational Engineering, 92(1), 187–195.

Mahrozi, N., & Faisal, M. (2023). Analisis perbandingan kecepatan algoritma Selection Sort dan Bubble Sort. Scientica: Jurnal Ilmiah Sains dan Teknologi, 1(2), 89–98.

Nasution, A. N., Singgam, P., Al-hafiz, A. Y., & Tampubolon, J. D. (2024). Analisis penggunaan algoritma Quick Sort pada pengolahan data. VISA: Journal of Visions and Ideas, 4(3), 1197–1204.

Panggabean, A. B., Htb, R. R., Perina, I., Toro, Y. L., & Syahputra, A. (2023). Implementasi algoritma Bubble Sort pada sistem pelayanan perpustakaan menggunakan Laravel. Sudo Jurnal Teknik Informatika, 2(1), 19–27.

Pujiono, I. P., Rachmawanto, E. H., & Winarsih, N. A. S. (2024). Array sorting algorithm vs algoritma pengurutan tradisional: Analisis efisiensi memori dan waktu. Jurnal Manajemen Informatika (JAMIKA), 15(1), 47–59.

Pujiono, I. P., Trianto, R. B., & Hana, F. M. (2024). Perbandingan efisiensi memori dan waktu komputasi pada 7 algoritma sorting menggunakan bahasa pemrograman Java. Jurnal Sistem Informasi dan Sistem Komputer, 9(2), 218–230.

Ramadhan, R. P., & Widiono, S. (2025). Pengembangan aplikasi pemesanan makanan berbasis Android dengan algoritma Bubble Sort. Jurnal Informatika Teknologi dan Sains (JINTEKS), 7(4), 1722–1727.

Riki, M. F., Hidayah, T. S., & Soeharsono. (2024). Perbandingan algoritma Selection Sort, Shell Sort, dan Merge Sort pada data sampling numerik menggunakan matplotlib. Prosiding Seminar Nasional Sains dan Teknologi, 1(2), 253–265.

Riyadi, A., Naafian, N. R., Ario, K., & Wibowo, T. (2022). Perancangan sistem informasi peminjaman barang inventaris. Indonesian Journal of

Information Technology and Computing, 2(2).

Rizky, M., Barokah, S., Saputra, T., & Sutabri, T. (2025). Perbandingan kinerja algoritma Bubble Sort dan Insertion Sort dalam pengurutan data penjualan UMKM. MIFORTEKH, 5(1).

Saputra, D. A., Ramadhani, M. S., Failandri, M. A., Turmudi, A., & Pujiono, I. P. (2025). Analisis perbandingan algoritma sorting dalam Javascript untuk meningkatkan efisiensi pengurutan produk pada aplikasi e-commerce. Sainstech, 35(2), 1–10.

Sena, M. B., Hanun, R. M., Purnama, I. R., Ardian, M., & Suharsono. (2024). Perbandingan kinerja algoritma sorting dalam pengurutan data menggunakan bahasa Python. Prosiding Seminar Nasional Sains dan Teknologi, 1(2), 310–314.

Sundaramoorthy, S., & Karunanidhi, G. (2025). A systematic analysis on performance and computational complexity of sorting algorithms. Discover Computing, 28.

Tanjung, Y. S. Z. P., Tambunan, Y. S., & Lubis, R. H. (2023). Penerapan metode FIFO dan LIFO dalam menjaga efektivitas persediaan pupuk. Jurnal Ekonomi Bisnis dan Manajemen, 1(1), 1–8.