

Analisis Kinerja Algoritma Bubble Sort, Selection Sort, dan Insertion Sort Menggunakan Python pada Data Mahasiswa

Gustina^{1*}, Ahmad Husna Ahadi², Fattachul Huda Aminuddin³, Muhammad Farhan Syawal⁴

Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Nurdin Hamzah¹²³⁴

E-mail: gus17tina08@gmail.com^{1*}, ahmad_husna@unh.ac.id², fattachulhuda@unh.ac.id³, frhnsywl36@gmail.com⁴

Abstrak

Penelitian ini bertujuan untuk menganalisis kinerja tiga algoritma pengurutan dasar. Bubble Sort, Selection Sort, dan Insertion Sort yang diimplementasikan menggunakan bahasa pemrograman Python 3.12.10. Pengujian dilakukan terhadap lima variasi ukuran dataset berisi data mahasiswa, masing-masing berjumlah 50, 100, 150, 200, dan 250 data. Setiap algoritma diuji pada tiga kondisi urutan data, yaitu acak, hampir terurut, dan terbalik. Hasil pengujian menunjukkan bahwa Insertion Sort memiliki waktu eksekusi paling efisien pada data hampir terurut dengan rata-rata di bawah 1 milidetik, sementara Bubble Sort menjadi algoritma paling lambat, khususnya pada data terbalik dengan waktu eksekusi mencapai lebih dari 30 milidetik. Selection Sort menunjukkan performa stabil di setiap kondisi, namun masih lebih lambat dibandingkan Insertion Sort pada sebagian besar skenario. Secara umum, hasil empiris ini mendukung teori kompleksitas waktu $O(n^2)$ dan memperkuat pembelajaran konsep efisiensi algoritma dalam mata kuliah Algoritma dan Struktur Data pada Program Studi Teknologi Informasi.

Kata kunci: Algoritma Pengurutan; Bubble Sort; Selection Sort; Insertion Sort; Python.

Abstract

This study aims to analyze the performance of three fundamental sorting algorithms—Bubble Sort, Selection Sort, and Insertion Sort—implemented using the Python 3.12.10 programming language. The experiment was conducted using five dataset sizes consisting of student data with 50, 100, 150, 200, and 250 records. Each algorithm was tested under three data order conditions: random, nearly sorted, and reversed. The experimental results indicate that Insertion Sort achieved the most efficient execution time for nearly sorted data, averaging below one millisecond, while Bubble Sort exhibited the slowest performance, particularly on reversed data, with execution times exceeding 30 milliseconds. Selection Sort maintained consistent performance across conditions but remained slower than Insertion Sort in most cases. Overall, the empirical findings support the theoretical $O(n^2)$ time complexity and reinforce the learning of

algorithm efficiency concepts in the Algorithm and Data Structure course of the Information Technology Study Program.

Keywords: *Sorting Algorithm; Bubble Sort; Selection Sort; Insertion Sort; Python.*

Submitted: 2025-11-08. **Revision:** 2025-11-12. **Accepted:** 2025-11-12. **Publish:** 2025-11-13.

PENDAHULUAN

Perkembangan teknologi informasi yang pesat menuntut mahasiswa, terutama di bidang informatika, memiliki kemampuan analitis dan logis untuk memahami konsep algoritma. Algoritma merupakan dasar pemrograman komputer karena menyediakan langkah-langkah sistematis untuk menyelesaikan masalah komputasi (Abdullah et al., 2023)

Salah satu materi fundamental dalam mata kuliah Algoritma dan Struktur Data adalah algoritma pengurutan (sorting), yang penting dalam meningkatkan efisiensi pengolahan data, seperti pengurutan data mahasiswa, pemrosesan data transaksi, dan pencarian informasi berbasis indeks (Bawamenewi et al., 2022; Iskandar et al., 2023)

Tiga algoritma pengurutan dasar yang sering digunakan dalam pembelajaran adalah Bubble Sort, Selection Sort, dan Insertion Sort karena strukturnya sederhana dan mudah dipahami (Saputri Saputri & Yahfizham Yahfizham, 2023). Efisiensi waktu eksekusi tiap algoritma berbeda tergantung kondisi data yang diproses, seperti data acak, hampir terurut, atau terbalik (Hassan et al., 2025)

Pemanfaatan Python dalam eksperimen memberikan kemudahan karena sintaksnya sederhana, mendukung modul pengukuran waktu, dan memungkinkan

mahasiswa fokus pada logika algoritma (Yunial, 2025). Pendekatan praktikum berbasis eksperimen membantu mahasiswa memahami konsep efisiensi algoritma secara konkret (Abdullah et al., 2023)

Dari sisi urgensi, kemampuan menganalisis dan mengukur efisiensi algoritma sangat penting bagi mahasiswa informatika di era data yang semakin besar dan kompleks. Pemahaman yang baik terhadap karakteristik algoritma pengurutan membantu mahasiswa dalam memilih metode yang tepat untuk kebutuhan pengolahan data di berbagai sistem, termasuk sistem informasi akademik. Selain itu, penguasaan konsep efisiensi algoritma juga mendukung peningkatan kualitas pembelajaran berbasis pemrograman dan komputasi.

Adapun gap penelitian terletak pada terbatasnya kajian yang menggunakan *dataset* nyata dari lingkungan akademik untuk membandingkan kinerja algoritma pengurutan dasar. Sebagian penelitian sebelumnya masih berfokus pada simulasi data acak tanpa mempertimbangkan variasi kondisi data yang umum terjadi dalam sistem nyata (Abdullah et al., 2023). Oleh karena itu, penelitian ini memiliki kebaruan (*novelty*) pada penerapan algoritma Bubble Sort, Selection Sort, dan Insertion Sort terhadap *dataset* mahasiswa yang diambil dari sistem akademik (SIKAD), serta pengujian efisiensi waktu eksekusi

menggunakan Python versi terbaru sebagai alat analisis empiris.

Penelitian ini bertujuan untuk mengukur dan menganalisis kinerja tiga algoritma pengurutan dasar, yaitu Bubble Sort, Selection Sort, dan Insertion Sort, dalam mengolah data mahasiswa menggunakan bahasa pemrograman Python. Dataset yang digunakan berupa data NIM dan nama mahasiswa sebanyak 250 entri dengan variasi ukuran data 50, 100, 150, 200, dan 250. Hasil analisis ini diharapkan dapat menjadi bahan ajar dalam kegiatan praktikum mahasiswa Program Studi Teknologi Informasi, serta memberikan gambaran empiris mengenai perbandingan efisiensi tiga algoritma tersebut dalam berbagai kondisi data.

METODE PENELITIAN

A. Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimen kuantitatif komparatif, yang bertujuan untuk membandingkan kinerja tiga algoritma pengurutan dasar: Bubble Sort, Selection Sort, dan Insertion Sort. Desain eksperimen ini dipilih untuk memperoleh data empiris terkait efisiensi waktu eksekusi dari masing-masing algoritma dalam berbagai kondisi dataset (Abdullah et al., 2023).

B. Tempat dan Waktu Penelitian

Penelitian dilaksanakan di Laboratorium Komputer Program Studi Teknologi Informasi sebagai bagian dari kegiatan praktikum mata kuliah Algoritma dan Struktur Data. Lingkungan praktikum

menggunakan perangkat komputer dengan sistem operasi Windows 10 dan bahasa pemrograman Python versi 3.12.10.

C. Populasi dan Sampel

Populasi dalam penelitian ini adalah data mahasiswa Program Studi Teknologi Informasi, Teknik Informatika, dan Sistem Informasi. Sampel data diambil sebanyak 250 entri, yang terdiri dari NIM dan Nama Mahasiswa, dan disimpan dalam format CSV agar mudah diproses oleh Python. Dataset kemudian dibagi menjadi lima kelompok ukuran data, yaitu 50, 100, 150, 200, dan 250 entri, guna melihat pengaruh ukuran data terhadap waktu eksekusi algoritma.

D. Teknik Pengumpulan Data

Data dikumpulkan melalui proses simulasi pemrograman. Setiap algoritma dijalankan pada tiga kondisi dataset berbeda, yaitu:

1. Data acak, untuk merepresentasikan kondisi umum tanpa urutan tertentu.
2. Data hampir terurut, untuk melihat bagaimana algoritma bekerja pada data yang sebagian sudah tersusun.
3. Data terbalik, untuk menguji efisiensi algoritma pada kondisi terburuk (worst case).

Waktu eksekusi diukur menggunakan modul time pada Python untuk mendapatkan durasi proses secara presisi dalam satuan *millisecond* (ms).

E. Teknik Analisis Data

Analisis dilakukan dengan membandingkan hasil waktu eksekusi dari ketiga algoritma pada setiap variasi ukuran data dan kondisi urutan. Hasil pengujian waktu eksekusi dibandingkan dalam tabel dan grafik, dianalisis secara deskriptif (Yunial, 2025 ; Iskandar et al., 2023)

Data hasil pengukuran selanjutnya dianalisis secara deskriptif untuk menjelaskan perbedaan kinerja algoritma berdasarkan efisiensi waktu.

Metode ini memberikan gambaran komprehensif mengenai bagaimana setiap algoritma merespons perubahan ukuran dan pola data, serta dapat digunakan sebagai dasar pengambilan keputusan dalam pembelajaran konsep efisiensi algoritma di kelas praktikum.

HASIL DAN PEMBAHASAN

A. Hasil

Penelitian ini menghasilkan data empiris mengenai waktu eksekusi dari tiga algoritma pengurutan, yaitu Bubble Sort, Selection Sort, dan Insertion Sort, yang diimplementasikan menggunakan bahasa pemrograman Python 3.12.10. Pengujian dilakukan terhadap lima ukuran dataset yang terdiri dari 50, 100, 150, 200, dan 250 data mahasiswa dengan tiga kondisi urutan data: acak, hampir terurut, dan terbalik.

Setiap algoritma dijalankan sebanyak tiga kali untuk setiap kombinasi ukuran dan kondisi data. Nilai waktu yang ditampilkan merupakan rata-rata dari hasil eksekusi. Data hasil pengujian direkap dalam tabel

dan grafik agar dapat digunakan pada tahap analisis lebih lanjut.

Tabel 1. Hasil pengujian kondisi data acak.

Ukuran Data	Bubble Sort (ms)	Selection Sort (ms)	Insertion Sort (ms)
50	0.637082	0.540899	0.429027
100	4.221602	3.136636	4.724552
150	9.921975	4.920433	4.137335
200	12.550713	6.787832	5.089134
250	13.866952	6.220249	4.796962

Seiring bertambahnya ukuran data, waktu eksekusi ketiga algoritma meningkat. Insertion Sort menunjukkan performa paling cepat hampir di semua ukuran data, Bubble Sort cenderung paling lambat, sedangkan Selection Sort tampil lebih konsisten. Hasil ini menunjukkan ukuran data memengaruhi efisiensi waktu eksekusi tiap algoritma.

Tabel 2. kondisi data hampir terurut

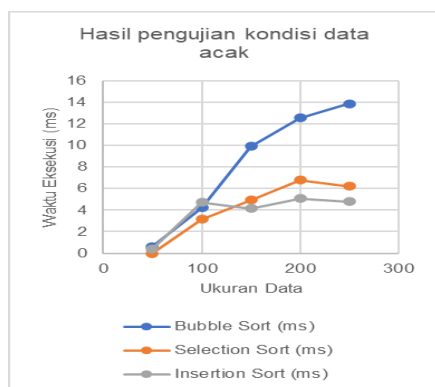
Ukuran Data	Bubble Sort (ms)	Selection Sort (ms)	Insertion Sort (ms)
50	0.614964	0.943995	0.227476
100	2.732347	2.552092	0.740774
150	7.90783	5.930444	1.851298
200	11.008869	9.039211	0.971595
250	5.656498	5.298479	0.836481

Waktu eksekusi ketiga algoritma cenderung meningkat seiring bertambahnya ukuran data, meskipun dataset hampir terurut. Insertion Sort tetap menunjukkan kinerja paling cepat di hampir semua ukuran data. Bubble Sort relatif lebih lambat, sementara Selection Sort tampil stabil tetapi lebih lambat dibanding Insertion Sort pada data yang hampir terurut.

Tabel 3. Hasil pengujian kondisi data terbaik

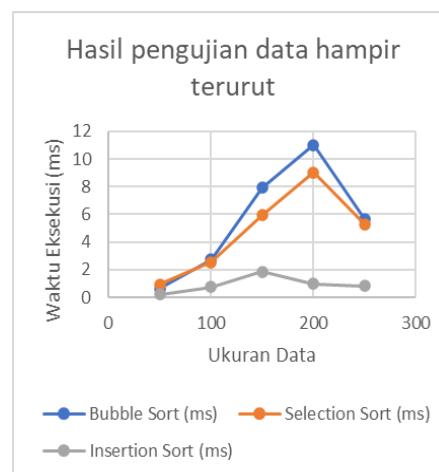
Ukuran Data	Bubble Sort (ms)	Selection Sort (ms)	Insertion Sort (ms)
50	1.407552	1.254195	1.027694
100	4.918719	5.18928	3.83919
150	31.30381	4.270884	5.11206
200	9.860104	4.982178	6.354714
250	13.14096	6.750111	7.885218

Pada kondisi data terbaik, Insertion Sort umumnya lebih cepat dibanding Bubble Sort dan Selection Sort, meskipun perbedaan waktu eksekusi tidak terlalu besar pada beberapa ukuran data. Bubble Sort cenderung memiliki waktu lebih tinggi, sedangkan Selection Sort relatif stabil.



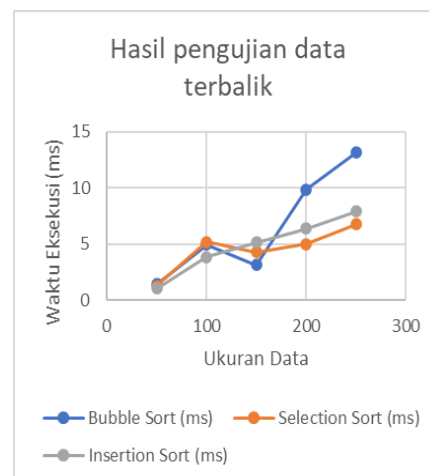
Grafik 1. Perbandingan Waktu Eksekusi (ms) pada data acak

Waktu eksekusi ketiga algoritma meningkat seiring bertambahnya ukuran data. Insertion Sort paling cepat, Bubble Sort paling lambat, dan Selection Sort menunjukkan kinerja yang relatif stabil (Putra et al., 2023).



Grafik 2. Perbandingan Waktu Eksekusi (ms) pada data hampir terurut

Seiring bertambahnya ukuran data hampir terurut, Insertion Sort tetap paling cepat, Bubble Sort paling lambat, dan Selection Sort menunjukkan kinerja yang stabil.



Grafik 3. Perbandingan Waktu Eksekusi (ms) pada data terbalik

Pada data terbalik, Insertion Sort umumnya tercepat, Bubble Sort cenderung lambat, dan Selection Sort menunjukkan kinerja yang relatif stabil di semua ukuran data.

Setelah mencantumkan tabel/grafik, berikan penjelasan dari tabel dan grafik tersebut dan kaitkan data yang ada dengan penelitian yang relevan dengan penelitian anda.

B. Pembahasan

Hasil pengujian menunjukkan bahwa Insertion Sort paling efisien pada dataset hampir terurut, rata-rata <1 ms, karena hanya memindahkan elemen yang diperlukan ke posisi tepat, sehingga jumlah operasi minimal (Yunial, 2025; Retnoningsih, 2018). Bubble Sort paling lambat terutama pada dataset terbalik (>30 ms) karena membutuhkan banyak pertukaran elemen (Abdullah et al., 2023; Mahrozi, N, Faisal, 2023) Selection Sort stabil pada semua kondisi tetapi tetap lebih lambat dibanding Insertion Sort karena tetap melakukan banyak perbandingan elemen (Bawamenewi et al., 2022; Iskandar et al., 2023)

Analisis tabel dan grafik juga memperlihatkan pengaruh ukuran dataset: dataset besar memperbesar perbedaan kinerja algoritma, di mana Insertion Sort tetap efisien pada data hampir terurut, dan Bubble Sort meningkat drastis pada dataset besar. Hasil ini konsisten dengan literatur bahwa karakteristik dataset dan urutan data memengaruhi efektivitas algoritma (Saputri Saputri & Yahfizham Yahfizham, 2023).

Penggunaan Python memudahkan mahasiswa mengamati langsung pengaruh pola dan ukuran dataset terhadap kinerja algoritma, menegaskan teori kompleksitas $O(n^2)$ (Wirth, 1976; Sedgewick, R. &

Wayne, 2011) sekaligus memberikan pengalaman praktikum yang aplikatif (Sunandar, 2019; Bawamenewi et al., 2022) Dengan demikian, hasil pengujian empiris mendukung literatur sekaligus menyimpulkan bukti nyata tentang efisiensi algoritma pada berbagai kondisi data.

Secara keseluruhan, eksperimen ini menegaskan bahwa pemilihan algoritma pengurutan harus disesuaikan dengan karakteristik data. Data yang hampir terurut memberikan keuntungan besar bagi Insertion Sort, sedangkan Bubble Sort dan Selection Sort lebih stabil namun kurang optimal pada dataset yang sebagian besar sudah tersusun. Penelitian ini memperkuat pemahaman empiris mahasiswa mengenai efektivitas algoritma, serta menjadi dasar pembelajaran berbasis eksperimen yang aplikatif dalam mata kuliah Algoritma dan Struktur Data.

KESIMPULAN

Berdasarkan hasil eksperimen dan literatur, karakteristik dataset berpengaruh signifikan terhadap efisiensi algoritma. Insertion Sort paling efisien pada data hampir terurut, Bubble Sort paling lambat terutama pada data terbalik, dan Selection Sort stabil tetapi kurang optimal dibanding Insertion Sort (Yunial, 2025; Abdullah et al., 2023; (Mahrozi, N, Faisal, 2023)

Ukuran dataset memperbesar perbedaan performa: semakin besar data, semakin terlihat keunggulan Insertion Sort dan kelemahan Bubble Sort. Python mempermudah implementasi dan pengukuran waktu presisi, sehingga

mahasiswa dapat menghubungkan konsep kompleksitas waktu dengan hasil empiris (Sunandar, 2019; (Wirth, 1976; Sedgewick, R. & Wayne, 2011)

Kesimpulan akhir menunjukkan bahwa pemilihan algoritma harus disesuaikan dengan karakteristik dataset. Dataset hampir terurut menguntungkan Insertion Sort, sementara Bubble Sort dan Selection Sort lebih stabil tetapi kurang optimal. Hasil ini memberikan dasar empiris yang kuat untuk pembelajaran praktikum Algoritma dan Struktur Data, meningkatkan kemampuan analitis mahasiswa dalam mengevaluasi performa algoritma dan memperkuat pemahaman konsep efisiensi algoritma dalam konteks nyata (Bawamenewi et al., 2022; Iskandar et al., 2023; Saputri Saputri & Yahfizham Yahfizham, 2023; Retnoningsih, 2018; Sunandar, 2019; Sedgewick, R. & Wayne, 2011)

DAFTAR PUSTAKA

Abdullah, M. F., Hafiza, I., Wahyuni, R., & Syahputra, A. (2023). Penggunaan Algoritma Bubble Sort dalam Pengurutan Nomor Induk Mahasiswa. *Hello World Jurnal Ilmu Komputer*, 2(1), 14–19. <https://doi.org/10.56211/helloworld.v2i1.206>

Bawamenewi, D., Tanoto, C., Banjarnahor, J., Nababan, N. K., Purba, W., & Aisyah, S. (2022). Implementasi Metode Selection Sort Dalam Sistem Repository Skripsi. *Jurnal Sistem Informasi Dan Ilmu Komputer Prima*, 5(2).

Hassan, M., Muhammad, G., Malik, J. A., Akhtar, M., Baloch, M. A., & Rajwana,

M. A. (2025). *Sort Data Faster : Comparing Algorithms*. 01(02), 28–37.

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. <https://doi.org/10.33379/gtech.v8i1.3556>

Mahrozi, N, Faisal, M. (2023). Analisis Perbandingan Kecepatan Algoritma Selection Sort dan Bubble Sort. *Scientica (Jurnal Ilmiah Sain Dan Teknologi)*, 1(2), 89–98.

Putra, Y. I., Kusmana, A., & Fitrah, Y. (2023). Falsifikasi sebagai pedoman Memahami Informasi di Media Sosial secara Objektif. *Jurnal Inovasi Pendidikan Dan Teknologi Informasi (JIPTI)*, 4(2), 289–295. <https://doi.org/10.52060/pti.v4i2.1515>

Retnoningsih, E. (2018). *Copyright@2018. P2M STMIK BINA INSANI INFORMATION MANAGEMENT FOR EDUCATORS AND PROFESSIONALS Algoritma Pengurutan Data (Sorting) Dengan Metode Insertion Sort dan Selection Sort*. 3(1), 95–106.

Saputri Saputri, & Yahfizham Yahfizham. (2023). Analisis Study Komperatif Bubble Sort Dan Selection Sort Pada Algoritma Dan Pemograman Berdasarkan Seleksi Pengurutan. *Jurnal Arjuna : Publikasi Ilmu Pendidikan, Bahasa Dan Matematika*, 1(6), 151–161. <https://doi.org/10.61132/arjuna.v1i6.305>

Sedgewick, R. & Wayne, K. (2011). *Algorithms (4th Edition)*. Addison-Wesley.

Sunandar, E. (2019). *Perbandingan Metode Selection Sort dan Insertion Sort dalam Pengurutan Data Menggunakan*

Bahasa Program Java. 12(2).

Wirth, N. (1976). *Algorithms + data structures = programs*. Prentice Hall.

Yunial, A. H. (2025). *Analisa Perbandingan Algoritma Bubble Sort, Insert Sort, dan Selection Sort. 10(1), 7–14.*
<https://doi.org/10.32493/informatika.v10i1.44216>