

IMPLEMENTASI ALGORITMA INSERTION SORT DALAM SISTEM PENILAIAN AKADEMIK BERBASIS WEB

Alfarizi Wijaya¹, Nazwa Aidilia Octa Mevia², Raffi Anggi Reswen³, Ammar Kamil
Al-Abror⁴, Fanny Ramadhani⁵

alfarizisitimorang98@gmail.com¹, nazwaaidila9@gmail.com², raffi.anggi@gmail.com³,
ammarkamil1310@gmail.com⁴, fannyr@unimed.ac.id⁵

Universitas Negeri Medan

ABSTRAK

Pengelolaan nilai akademik secara manual seringkali tidak efisien dan rentan terhadap kesalahan manusia, terutama dalam proses penentuan peringkat siswa. Untuk mengatasi masalah ini, diperlukan sebuah sistem terkomputerisasi yang dapat mengotomatisasi proses tersebut secara akurat dan cepat. Penelitian ini bertujuan untuk merancang dan mengimplementasikan algoritma Insertion Sort dalam sebuah aplikasi berbasis web bernama S-Grade untuk pengelolaan peringkat siswa. Metode penelitian yang digunakan adalah pengembangan perangkat lunak dengan analisis kinerja algoritma. Aplikasi S-Grade dibangun menggunakan teknologi web standar yaitu HTML, CSS, dan JavaScript, di mana logika pengurutan diimplementasikan menggunakan fungsi Insertion Sort. Hasil dari penelitian ini adalah sebuah aplikasi fungsional yang mampu menerima input data siswa, mengurutkannya berdasarkan nilai secara ascending atau descending, menampilkan tabel peringkat, serta menyajikan statistik nilai dasar seperti rata-rata, nilai tertinggi, dan terendah. Analisis menunjukkan bahwa algoritma Insertion Sort terbukti sangat efektif dan responsif untuk skala data yang umum di lingkungan akademik (hingga 100 siswa), sejalan dengan teori yang menyatakan keunggulannya pada set data berukuran kecil hingga menengah.

Kata Kunci: Algoritma Pengurutan, Insertion Sort, Sistem Penilaian, Kinerja Algoritma, Aplikasi Web.

ABSTRACT

Manual management of academic grades is often inefficient and prone to human error, especially in the process of determining student rankings. To address this issue, a computerized system that can accurately and quickly automate the process is necessary. This research aims to design and implement the Insertion Sort algorithm within a web-based application named S-Grade for managing student rankings. The research method used is software development combined with an analysis of the algorithm's performance. The S-Grade application is built using standard web technologies—HTML, CSS, and JavaScript—where the sorting logic is implemented using the Insertion Sort function. The result of this research is a functional application capable of receiving student data input, sorting it by grades in ascending or descending order, displaying a ranking table, and presenting basic statistics such as the average, highest, and lowest scores. The analysis shows that the Insertion Sort algorithm proves to be highly effective and responsive for data scales common in academic environments (up to 100 students), which is consistent with the theory stating its advantages for small to medium-sized data sets.

Keywords: Sorting Algorithm, Insertion Sort, Grading System, Algorithm Performance, Web. Application.

PENDAHULUAN

Perkembangan teknologi yang terus berlanjut telah menghasilkan peningkatan volume data secara eksponensial, sehingga penyelesaian secara manual menjadi tidak lagi efisien [1]. Adanya algoritma pemrograman menjadi solusi fundamental dalam menyelesaikan proses pengelolaan data agar informasi yang dihasilkan lebih akurat dan terpercaya [2]. Spektrum algoritma sangatlah luas, mencakup tidak hanya pengurutan data tetapi juga teknik lain seperti kompresi data dengan Huffman Coding dan penyembunyian data melalui steganografi [3]. Di antara berbagai jenis algoritma, algoritma pengurutan

(sorting) memegang peran krusial dalam menyusun data secara teratur baik menaik (ascending) maupun menurun (descending), yang merupakan langkah penting untuk mempermudah pemrosesan data lebih lanjut [4].

Terdapat berbagai macam algoritma pengurutan yang telah dikembangkan, masing-masing dengan karakteristiknya sendiri. Algoritma sederhana seperti Bubble Sort dan Selection Sort, misalnya, sering dianalisis karena kemudahan implementasinya meskipun memiliki kelemahan dalam hal efisiensi waktu [5], [6]. Di antara algoritma-algoritma sederhana ini, Insertion Sort menonjol karena konsepnya yang intuitif [7]. Cara kerjanya sering dianalogikan seperti seseorang yang mengurutkan kartu di tangan, di mana selebar kartu diambil dari tumpukan acak lalu disisipkan (insert) ke posisi yang tepat pada kumpulan kartu yang sudah teratur [7], [8]. Proses ini membagi data menjadi dua bagian, yaitu bagian teratur dan belum teratur, dan secara iteratif memindahkan elemen dari bagian belum teratur ke posisi yang benar di bagian teratur.

Kinerja Insertion Sort sangat dipengaruhi oleh kondisi awal data. Berbagai literatur dan penelitian secara konsisten menyimpulkan bahwa kompleksitas waktu untuk kasus rata-rata (average case) dan kasus terburuk (worst case) adalah $O(n^2)$ [9], [10]. Namun, keunggulan utamanya terletak pada kasus terbaik (best case), di mana jika data sudah dalam keadaan teratur atau hampir teratur, kompleksitas waktunya menjadi sangat efisien, yaitu $O(n)$ [10], [11]. Dalam perbandingan, Insertion Sort menunjukkan performa yang lebih baik dari Bubble Sort, namun untuk set data yang besar, efisiensinya masih di bawah algoritma canggih seperti Merge Sort atau Quick Sort [10]. Meskipun begitu, untuk set data berukuran kecil (misalnya di bawah 100 elemen), Insertion Sort justru dapat lebih cepat karena implementasinya yang sederhana [11]. Kelebihan lain dari algoritma ini adalah sifatnya yang stabil dan hanya memerlukan sedikit memori tambahan (in-place) [4], [9].

Penerapan algoritma dalam konteks penilaian akademik menjadi sangat relevan. Pentingnya sebuah sistem yang objektif untuk evaluasi siswa, seperti penentuan siswa berprestasi atau pengolahan nilai ujian, telah menjadi fokus dalam banyak penelitian, baik menggunakan metode Weighted Product [12] maupun pengembangan sistem terintegrasi lainnya untuk mengelola peringkat [13]. Proses manual untuk tugas-tugas ini sangat memakan waktu dan berisiko menimbulkan kesalahan. Oleh karena itu, penelitian ini bertujuan untuk mengimplementasikan dan menganalisis kinerja algoritma Insertion Sort dalam sebuah "Sistem Peringkat Siswa Modern" bernama S-Grade. Aplikasi ini dirancang sebagai studi kasus praktis yang menunjukkan bagaimana fleksibilitas Insertion Sort dapat dimanfaatkan untuk menangani pemerincian multi-kriteria yang kompleks (nilai, sikap, dan prestasi), namun tetap efisien dalam skala yang sesuai dengan keunggulannya.

LANDASAN TEORI

Algoritma Pengurutan (Sorting)

Algoritma pengurutan (sorting) memegang peran krusial dalam menyusun data secara teratur baik menaik (ascending) maupun menurun (descending), yang merupakan langkah penting untuk mempermudah pemrosesan data lebih lanjut [4]. Adanya algoritma pemrograman menjadi solusi fundamental dalam menyelesaikan proses pengelolaan data agar informasi yang dihasilkan lebih akurat dan terpercaya [2].

Algoritma Insertion Sort

1. Konsep Dasar

Insertion Sort menonjol karena konsepnya yang intuitif [7]. Cara kerjanya sering dianalogikan seperti seseorang yang mengurutkan kartu di tangan, di mana selebar kartu diambil dari tumpukan acak lalu disisipkan (insert) ke posisi yang tepat pada kumpulan

kartu yang sudah terurut [7], [8]. Proses ini membagi data menjadi dua bagian, yaitu bagian terurut dan belum terurut, dan secara iteratif memindahkan elemen dari bagian belum terurut ke posisi yang benar di bagian terurut.

2. Kompleksitas Waktu

Kinerja Insertion Sort sangat dipengaruhi oleh kondisi awal data. Kompleksitas waktu untuk kasus rata-rata (average case) dan kasus terburuk (worst case) adalah $O(n^2)$ [9], [10]. Namun, keunggulannya terletak pada kasus terbaik (best case), di mana jika data sudah dalam keadaan terurut atau hampir terurut, kompleksitas waktunya menjadi sangat efisien, yaitu $O(n)$ [10], [11].

3. Karakteristik dan Keunggulan

Insertion Sort menunjukkan performa yang lebih baik dari Bubble Sort [5], [8]. Untuk set data berukuran kecil (misalnya di bawah 100 elemen), Insertion Sort justru dapat lebih cepat karena implementasinya yang sederhana [11]. Kelebihan lain dari algoritma ini adalah sifatnya yang stabil dan hanya memerlukan sedikit memori tambahan (in-place) [4], [9]. Algoritma ini juga fleksibel dan mudah dimodifikasi untuk memenuhi kebutuhan spesifik tanpa harus mengubah struktur dasar [14].

Sistem Penilaian Akademik

Pentingnya sebuah sistem yang objektif untuk evaluasi siswa, seperti penentuan siswa berprestasi atau pengolahan nilai ujian, telah menjadi fokus dalam banyak penelitian [12], [13]. Proses manual untuk tugas-tugas ini sangat memakan waktu dan berisiko menimbulkan kesalahan. Penggunaan algoritma pengurutan dalam aplikasi manajemen data akademik terbukti efektif untuk skala data kecil hingga menengah yang umum ditemukan di lingkungan pendidikan [15].

METODE PENELITIAN

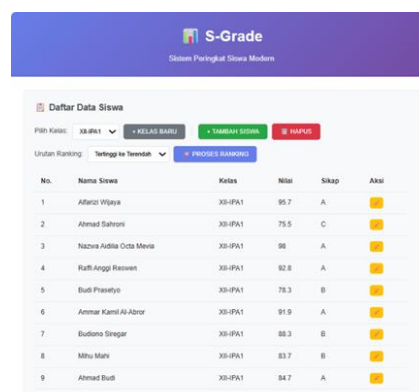
Metodologi yang digunakan dalam penelitian ini adalah pengembangan perangkat lunak (software development) untuk membangun aplikasi S-Grade, diikuti dengan analisis deskriptif terhadap fungsionalitas dan implementasi algoritma di dalamnya.

HASIL DAN PEMBAHASAN

Hasil dari penelitian ini adalah sebuah aplikasi berbasis web bernama S-Grade yang berfungsi penuh untuk melakukan pemeringkatan nilai siswa

1. Antarmuka Pengguna (User Interface)

- Tampilan baru S-Grade jauh lebih profesional dan kaya fitur. Halaman utama menyajikan daftar siswa dalam sebuah tabel yang informatif, lengkap dengan fungsionalitas manajemen kelas dan data siswa.



The screenshot displays the 'Daftar Data Siswa' (Student Data List) page of the S-Grade application. At the top, there's a header with the 'S-Grade' logo and the subtitle 'Sistem Penilaian Siswa Modern'. Below the header, there are navigation buttons: 'KELAS BARU', 'TAMBAH SISWA', and 'HAPUS'. A dropdown menu for 'Urutan Ranking' is set to 'Tertinggi ke Terendah'. The main content is a table with columns: No., Nama Siswa, Kelas, Nilai, Sikap, and Aksi. The table lists 9 students with their respective scores and attitudes.

No.	Nama Siswa	Kelas	Nilai	Sikap	Aksi
1	Adhico Wijaya	XI-IPA1	95.7	A	[icon]
2	Almad Sahroni	XI-IPA1	75.5	C	[icon]
3	Nazwa Adila Octa Meva	XI-IPA1	96	A	[icon]
4	Rafi Anggi Reswan	XI-IPA1	92.8	A	[icon]
5	Budi Prasetyo	XI-IPA1	78.3	B	[icon]
6	Ammar Kamil Al Abor	XI-IPA1	91.9	A	[icon]
7	Budono Singar	XI-IPA1	88.3	B	[icon]
8	Miru Muri	XI-IPA1	83.7	B	[icon]
9	Almad Rudi	XI-IPA1	84.7	A	[icon]

Gambar 1. Tampilan Antarmuka Utama S-Grade

- b. Setelah proses ranking dijalankan, hasil disajikan dalam format kartu yang terorganisir, memisahkan antara tabel peringkat, statistik nilai, dan fitur pencarian. Penggunaan lencana (badge) untuk peringkat 1, 2, dan 3 memberikan penekanan visual yang efektif.

The screenshot displays three main components of the application:

- Tabel Ranking Siswa (Student Ranking Table):** A table with columns: PERINGKAT (Rank), NAMA (Name), KELAS (Class), NILAI (Score), SIPAP (Status), PRESTASI (Achievement), and TINGKAT (Level). It lists 10 students with their respective scores and ranks. The first three students (Nuzwa Adila Ota Mewa, Afriani Wijaya, and Ruli Anggi Roswani) are highlighted with gold, silver, and bronze medals respectively.
- Statistik Nilai (Value Statistics):** A section showing four key statistics: Rata-rata (Average) at 87.66, Tertinggi (Highest) at 98.00, Terendah (Lowest) at 75.50, and Total Siswa (Total Students) at 9.
- Cari Peringkat Siswa (Find Student Rank):** A search interface with a text input field containing 'Budi', a 'Cari' (Search) button, and a 'CLEAR' button. Below the input, it shows search results for 'Budi Prasetyo' and 'Budi Prasetyo' with their respective details.

Gambar 2. Tampilan Hasil Peringkat, Statistik, dan Pencarian

2. Analisis Implementasi Algoritma

Logika inti dari aplikasi ini adalah fungsi `insertionSort()` yang diperkaya dengan fungsi `compareStudent()` untuk menangani tie-breaking, seperti yang terlihat pada cuplikan kode dari `Script.js` berikut:

```

1  ===== ALGORITMA INSERTION SORT =====
2  function insertionSort(data, ascending = false) {
3  // Fungsi untuk mengurutkan data siswa menggunakan algoritma insertion sort
4  let sortedData = [...data]; // Menyalin data agar data asli tidak berubah
5
6  for (let i = 1; i < sortedData.length; i++) {
7    let key = sortedData[i];
8    let j = i - 1;
9
10   // Membandingkan berdasarkan nilai
11   if (ascending) {
12     // Urutkan nilai (terendah ke tertinggi)
13     while (j >= 0 && sortedData[j].score > key.score) {
14       sortedData[j + 1] = sortedData[j];
15       j = j - 1;
16     }
17   } else {
18     // Urutkan nilai (tertinggi ke terendah)
19     while (j >= 0 && sortedData[j].score < key.score) {
20       sortedData[j + 1] = sortedData[j];
21       j = j - 1;
22     }
23   }
24   sortedData[j + 1] = key;
25 }
26 }
27 return sortedData;
28 }
29

```

Gambar 3. Kode Fungsi `insertionSort`

Implementasi ini menunjukkan fleksibilitas dari algoritma Insertion Sort. Aplikasi ini secara cerdas menangani kasus di mana beberapa siswa memiliki nilai akademik yang sama. Dalam situasi ini, sistem akan menggunakan kriteria tambahan seperti nilai sikap, tingkat prestasi, dan tingkat kompetisi sebagai pemecah nilai yang sama (tie-breaker) secara berurutan, sebelum akhirnya mengurutkan berdasarkan nama. Pendekatan multi-kriteria ini memastikan pemeringkatan yang lebih adil dan komprehensif, sekaligus menegaskan sifat stabil dari algoritma yang tidak mengubah urutan relatif elemen dengan kriteria identik [4], [9]. Analisis ini memperkuat argumen dari literatur:

- a. Kinerja Tetap Optimal: Meskipun fungsi perbandingan menjadi lebih kompleks, jumlah data yang diurutkan (ukuran satu kelas) tetap berada dalam rentang "kecil", sehingga waktu eksekusi hampir tidak terasa oleh pengguna. Hal ini membuktikan bahwa Insertion Sort adalah pilihan yang tepat untuk aplikasi interaktif semacam ini [1], [11].

- b. Fleksibilitas: Algoritma ini mudah dimodifikasi untuk memenuhi kebutuhan spesifik, seperti penambahan kriteria peringkat baru di masa depan, tanpa harus mengubah struktur dasar dari algoritma pengurutannya [14].

Aplikasi ini secara cerdas menangani kasus di mana beberapa siswa memiliki nilai akademik yang sama. Dalam situasi ini, sistem akan menggunakan kriteria tambahan seperti nilai sikap, tingkat prestasi, dan tingkat kompetisi sebagai pemecah nilai yang sama (tie-breaker) secara berurutan, sebelum akhirnya mengurutkan berdasarkan nama. Pendekatan multi-kriteria ini memastikan pemeringkatan yang lebih adil dan komprehensif, sekaligus menegaskan sifat stabil dari algoritma yang tidak mengubah urutan relatif elemen dengan kriteria identik [1], [2].

KESIMPULAN

Penelitian ini berhasil mengimplementasikan algoritma Insertion Sort dengan logika perbandingan multi-kriteria ke dalam aplikasi web S-Grade. Aplikasi yang dihasilkan tidak hanya berfungsi sebagai alat pemeringkatan nilai, tetapi juga sebagai sistem manajemen data siswa tingkat kelas yang sederhana namun kuat, dilengkapi dengan fitur CRUD dan manajemen multi-kelas.

Berdasarkan hasil implementasi dan analisis, dapat disimpulkan bahwa Insertion Sort adalah algoritma yang fleksibel dan efisien untuk aplikasi dengan skala data kecil hingga menengah. Kemampuannya untuk diadaptasi dengan logika pembandingan yang kompleks menjadikannya solusi praktis untuk skenario dunia nyata seperti penilaian akademik yang memerlukan tie-breaking yang adil. Proyek ini berhasil menunjukkan bahwa algoritma klasik seperti Insertion Sort tetap memiliki relevansi tinggi dalam pengembangan perangkat lunak modern ketika diterapkan pada konteks dan skala yang tepat [15].

DAFTAR PUSTAKA

- E. Retnoningsih, "Algoritma Pengurutan Data (Sorting) Dengan Metode Insertion Sort dan Selection Sort," *Information Management For Educators And Professionals*, vol. 3, no. 1, pp. 95-106, Dec. 2018.
- Saputri and Yahfizham, "Analisis Study Komperatif Bubble Sort Dan Selection Sort Pada Algoritma Dan Pemograman Berdasarkan Seleksi Pengurutan," *Jurnal Arjuna: Publikasi Ilmu Pendidikan, Bahasa dan Matematika*, vol. 1, no. 6, pp. 151-161, Dec. 2023.
- H. Saputra, M. Z. S. Hadi, and N. Syahroni, "Implementasi Algoritma Steganografi Embedding dengan Metode Least Significant Bit (LSB) Insertion dan Huffman Coding pada Pengiriman Pesan Menggunakan Media MMS Berbasis J2ME," *Jurnal Teknik Telekomunikasi*, pp. 1-6.
- M. Dhamma, "Analisis Kompleksitas Diantara Algoritma Insertion Sort dan Selection Sort dan Diimplemntasikan dengan Bahasa Pemograman Java," *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan*, vol. 8, no. 1, pp. 7-9, Sep. 2023.
- N. Mahrozi and M. Faisal, "Analisis Perbandingan Kecepatan Algoritma Selection Sort dan Bubble Sort," *Scientica: Jurnal Ilmiah Sain dan Teknologi*, vol. 1, no. 2, pp. 89-98, 2023.
- Safwandi and Nurdin, *Struktur Data*. Lhokseumawe: Fakultas Teknik Universitas Malikussaleh, 2014.
- Painem, R. Ragam, and M. Sjukani, *Algoritma dan Struktur Data 1*. Yogyakarta: Deepublish, 2019.
- M. R. S. Barokah, T. Saputra, and T. Sutabri, "Perbandingan Kinerja Algoritma Bubble Sort dan Insertion Sort dalam Pengurutan Data Penjualan UMKM," *MIFORTEKH (Jurnal Manajemen Informatika & Teknologi)*, vol. 5, no. 1, pp. 184-195, May 2025.
- R. R. Basir, "Analisis Kompleksitas Ruang dan Waktu Terhadap Laju Pertumbuhan Algoritma Heap Sort, Insertion Sort dan Merge dengan Pemrograman Java," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 5, no. 2, pp. 109-118, Dec. 2020.

- Y. Heryanto, Fauziah, and T. W. Harjanti, "Analisis Perbandingan Ruang dan Waktu pada Algoritma Sorting Menggunakan Bahasa Pemrograman Python," *KESATRIA Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)*, vol. 4, no. 2, pp. 342-347, Apr. 2023.
- V. F. Tambunan, S. Akva, S. Davina, and Z. Indra, "Analisis Kinerja Insertion Sort dan Selection Sort dalam Pemrograman Python," *Jurnal Kajian Riset Multidisiplin*, vol. 8, no. 12, pp. 33-38, Dec. 2024.
- R. Nursyanti and M. S. Hartawan, "Aplikasi Berbasis Web untuk Pemilihan Siswa Berprestasi Menggunakan Metode Weighted Product," *Sisinfo: Jurnal Sistem Informasi dan Informatika*, vol. 4, no. 2, pp. 66-70, Aug. 2022.
- N. C. Hilary, E. Y. Abadi, A. Elizabeth, B. A. Z. Tristiani, and A. B. Aulia, "Pengembangan Sistem Pengolahan Nilai Ujian dan Peringkat Siswa," *JISEM Jurnal Program Studi Informatika Universitas Katolik Widya Mandala Surabaya*, vol. 1, no. 1, pp. 32-37, Apr. 2025.
- F. R. Hakim, Ryan, H. S. Ismallah, M. L. Firdaus, and Suharsono, "Penerapan Algoritma Insertion Sorting Terhadap Data Transaksi Saham Per Kota di Indonesia," *Prosiding Seminar Nasional Sains dan Teknologi Seri 02*, vol. 1, no. 2, pp. 332-341, 2024.
- M. Reza, K. Nasution, and M. Z. Siambaton, "Penerapan Algoritma Insertion Sort Pada Aplikasi Manajemen Aset E-Asset Fakultas Universitas Islam Sumatera Utara," *Jurnal Deli Sains Informatika*, vol. 1, no. 1, pp. 68-79, Jan. 2022.