



PENERAPAN ALGORITMA SELECTION SORT UNTUK MELAKUKAN PENGURUTAN DATA BUKU DALAM BAHASA PEMROGRAMAN HTML (STUDI KASUS: PERPUSTAKAAN KAPUTAMA)

Luthfiana Emalia¹, Bunga Sabit Br Ginting², Marshela zalianti³, Safrizal⁴

^{1,2,3} Program Studi Sistem Informasi, Fakultas Ilmu Komputer, STMIK Kaputama Binjai

⁴ Program Studi Sistem Informasi, Fakultas Ekonomi Sains dan Teknologi, Universitas Muhammadiyah Asahan

E-mail: luthfianaemalia@gmail.com¹, bungasabitbruginting@gmail.com²,
marshelazalianti1@gmail.com³, rizalsyl75@gmail.com⁴

ABSTRACT

The rapid development of information technology has encouraged digital transformation in various fields, including library data management. This study aims to implement the Selection Sort algorithm in the process of sorting book data within the Kaputama Library information system using HTML, CSS, and JavaScript. The Selection Sort algorithm was chosen for its simplicity, memory efficiency, and accuracy in handling small to medium-sized datasets. The research method includes analyzing book data, applying the algorithm for both ascending and descending sorting based on book codes and titles, and testing system performance through web-based simulations. The results show that the Selection Sort algorithm successfully sorted book data with 100% accuracy and an average execution time of 4-6 milliseconds. The web-based system also provides an interactive visualization of the sorting process, enhancing users' understanding of the algorithm's workflow. Therefore, the implementation of Selection Sort in an HTML-based book data management system improves efficiency, accuracy, and usability in library information management.

Keywords: Selection Sort, Data Sorting, HTML, Library, Information System

ABSTRAK

Perkembangan teknologi informasi telah mendorong digitalisasi dalam berbagai bidang, termasuk pengelolaan data perpustakaan. Penelitian ini bertujuan untuk menerapkan algoritma *Selection Sort* dalam proses pengurutan data buku pada sistem informasi Perpustakaan Kaputama berbasis HTML, CSS, dan *JavaScript*. Algoritma *Selection Sort* dipilih karena kemudahannya dalam implementasi, efisiensi memori, dan hasil pengurutan yang akurat untuk jumlah data kecil hingga menengah. Metode penelitian meliputi tahapan analisis data buku, penerapan algoritma untuk pengurutan secara *ascending* dan *descending* berdasarkan kode maupun judul buku, serta pengujian performa sistem melalui simulasi berbasis web. Hasil penelitian menunjukkan bahwa algoritma *Selection Sort* mampu mengurutkan data buku dengan akurasi 100% dan waktu eksekusi rata-rata 4-6 milidetik. Sistem

Article History

Received: Oktober 2025

Reviewed: Oktober 2025

Published: Oktober 2025

Plagiarism Checker No
235

Prefix DOI :

[10.8734/Kohesi.v1i2.365](https://doi.org/10.8734/Kohesi.v1i2.365)

Copyright : Author

Publish by : Kohesi



This work is licensed
under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/)



berbasis web ini juga menampilkan visualisasi proses pengurutan secara interaktif, sehingga memudahkan pemahaman pengguna terhadap mekanisme kerja algoritma. Dengan demikian, penerapan *Selection Sort* dalam sistem pengelolaan data buku berbasis HTML dapat meningkatkan efisiensi, akurasi, dan kemudahan dalam pengelolaan informasi di lingkungan perpustakaan.

Kata Kunci: *Selection Sort, Pengurutan Data, HTML, Perpustakaan, Sistem Informasi*

1. PENDAHULUAN

Perkembangan teknologi informasi yang semakin pesat telah memberikan dampak signifikan terhadap berbagai aspek kehidupan, termasuk dalam bidang pendidikan dan pengelolaan informasi perpustakaan. Pengolahan data secara manual kini mulai ditinggalkan karena dinilai kurang efisien dalam hal kecepatan, ketepatan, dan akurasi hasil. Perpustakaan sebagai salah satu pusat informasi dan sumber belajar memerlukan sistem pengolahan data yang baik agar proses pencarian, penyimpanan, dan pengelolaan koleksi buku dapat berjalan lebih efektif. Salah satu permasalahan yang sering muncul dalam pengelolaan data buku adalah kesulitan dalam melakukan pengurutan data, baik berdasarkan judul, pengarang, maupun tahun terbit. Oleh karena itu, penerapan algoritma pengurutan (*sorting*) menjadi sangat penting untuk mengatasi masalah tersebut.

Algoritma *Selection Sort* merupakan salah satu metode pengurutan sederhana yang memiliki keunggulan dari segi kemudahan dalam implementasi dan efisiensi memori. Algoritma ini bekerja dengan cara memilih elemen terkecil atau terbesar dari sekumpulan data, kemudian menukarnya dengan elemen pada posisi tertentu hingga seluruh data terurut sesuai kriteria yang diinginkan. Berdasarkan penelitian yang dilakukan oleh Wisda (2022) dan Retnoningsih (2018), metode *Selection Sort* dinilai efektif untuk dataset kecil hingga menengah karena prosesnya yang sistematis dan hasilnya yang konsisten. Metode ini juga banyak digunakan dalam berbagai aplikasi seperti pengurutan data medis, data mahasiswa, serta pengelolaan stok barang.

Dalam konteks perpustakaan, data buku sering kali tersimpan dalam jumlah besar dengan berbagai atribut seperti judul, penulis, penerbit, tahun terbit, serta kategori. Tanpa sistem pengurutan yang baik, proses pencarian buku dapat memakan waktu lama dan menghambat efisiensi layanan. Penerapan algoritma *Selection Sort* pada sistem pengelolaan data buku berbasis HTML diharapkan dapat membantu pustakawan dan pengguna dalam menampilkan data secara teratur, baik secara *ascending* maupun *descending*, sesuai kebutuhan pencarian. Selain itu, implementasi berbasis HTML dipilih karena sifatnya yang ringan, mudah dipahami, dan dapat dijalankan melalui browser tanpa memerlukan perangkat lunak tambahan.

Studi kasus pada Perpustakaan Kaputama menjadi relevan karena lembaga ini masih menghadapi kendala dalam pengelolaan data koleksi buku secara terkomputerisasi. Dengan mengadopsi metode *Selection Sort* ke dalam sistem informasi sederhana berbasis HTML, diharapkan dapat tercipta sistem pengurutan data yang terstruktur, mudah digunakan, serta meningkatkan efisiensi dalam pelayanan informasi kepada pengguna. Penerapan ini tidak hanya



memberikan solusi praktis dalam pengurutan data, tetapi juga menjadi langkah awal menuju digitalisasi sistem informasi perpustakaan yang lebih modern dan terintegrasi.

2. METODE PENELITIAN

Metode penelitian ini menggunakan algoritma Selection Sort dalam proses pengurutan data buku di Perpustakaan Kaputama. Fokus utama penelitian ini adalah untuk mengurutkan data buku berdasarkan dua kriteria, yaitu judul buku dan kode buku. Kedua kriteria diuji dengan dua arah pengurutan, yaitu ascending (dari nilai terkecil ke terbesar) dan descending (dari nilai terbesar ke terkecil), menggunakan dataset berbentuk file CSV yang berisi data koleksi buku perpustakaan.

Data buku yang digunakan mencakup beberapa atribut penting seperti kode buku, judul buku, pengarang, penerbit, tahun terbit, dan kategori buku. Tujuan dari penelitian ini adalah untuk menganalisis kinerja algoritma *Selection Sort* dalam mengurutkan data buku secara efisien, akurat, dan terstruktur agar dapat diterapkan dalam sistem informasi perpustakaan Kaputama.

2.1 Algoritma Selection Sort

Algoritma *Selection Sort* merupakan salah satu algoritma penyortiran sederhana yang bekerja dengan cara memilih nilai minimum atau maksimum dari sekumpulan data dan menempatkannya ke posisi yang sesuai secara bertahap hingga seluruh data terurut. Pada data bertipe *character* seperti judul buku, perbandingan nilai dilakukan berdasarkan urutan ASCII (American Standard Code for Information Interchange), di mana setiap huruf memiliki nilai urutan tersendiri. Huruf "A" lebih kecil dari "B", dan seterusnya.

Keunggulan algoritma ini adalah prosesnya yang mudah dipahami, implementasinya sederhana, serta tidak membutuhkan ruang penyimpanan tambahan yang besar (*in-place sorting*). Namun, kelemahannya terletak pada waktu eksekusi yang relatif lebih lambat untuk dataset berskala besar.

2.2 Cara Kerja Algoritma Selection Sort

Proses kerja algoritma *Selection Sort* pada penelitian ini diterapkan untuk mengurutkan data buku berdasarkan kriteria tertentu (judul buku atau kode buku) dengan dua metode urutan, yaitu *ascending* dan *descending*. Berikut langkah-langkah kerja yang diterapkan dalam sistem pengurutan buku di Perpustakaan Kaputama:

1. Inisialisasi Data Buku

Menyediakan sekumpulan data buku dalam bentuk daftar (*array*) yang berisi informasi seperti kode dan judul buku.

2. Menentukan Posisi Awal

Menetapkan indeks pertama ($i = 0$) sebagai titik awal pencarian nilai minimum atau maksimum.

3. Mencari Nilai Minimum atau Maksimum



Membandingkan elemen pada indeks awal dengan seluruh elemen setelahnya. Jika ditemukan elemen yang lebih kecil (atau lebih besar untuk *descending*), maka posisi indeks tersebut disimpan sebagai indeks baru.

4. Pertukaran Posisi Data

Setelah semua elemen dibandingkan, elemen minimum atau maksimum ditukar dengan elemen pada posisi awal.

5. Iterasi Berikutnya

Pindahkan posisi awal ke indeks berikutnya ($i = i + 1$) dan ulangi langkah 3-4 sampai seluruh data terurut.

6. Menampilkan Hasil Pengurutan

Setelah proses selesai, data buku akan terurut secara sempurna sesuai kriteria dan arah pengurutan yang dipilih.

Rumus Proses Selection Sort

Secara matematis, proses pengurutan dengan *Selection Sort* dapat dinyatakan sebagai berikut:

$$MinIndex_{(i)} = \min_{j=i+1}^{n-1} (D_{[j]})$$

Untuk setiap elemen ke- i , algoritma akan mencari nilai terkecil (untuk ascending) atau terbesar (untuk descending) dari elemen ke- $i+1$ hingga elemen terakhir, kemudian melakukan pertukaran posisi dengan rumus:

$$swap(D_{[i]}, D_{[MinIndex(i)]})$$

Dengan keterangan:

- $D_{[i]}$ = elemen ke- i dari data buku
- n = jumlah total data buku
- $MinIndex_{(i)}$ = posisi indeks dengan nilai terkecil/terbesar dari iterasi ke- i

Rumus ini memastikan bahwa pada setiap iterasi, elemen dengan nilai terkecil (atau terbesar) ditempatkan pada posisi yang benar sehingga pada akhir proses, seluruh data buku sudah dalam urutan yang diinginkan.

Berikut adalah pseudocode algoritma *Selection Sort* yang diterapkan dalam penelitian ini:

```

SelectionSort (data, key):
    n = length of data
    for i from 0 to n-1:
        min_idx = i
        for j from i + 1 to n-1:

```

```

if data[min_idx][key] > data[j][key]:
    min_idx = j
swap data[i] with data[min_idx]

```

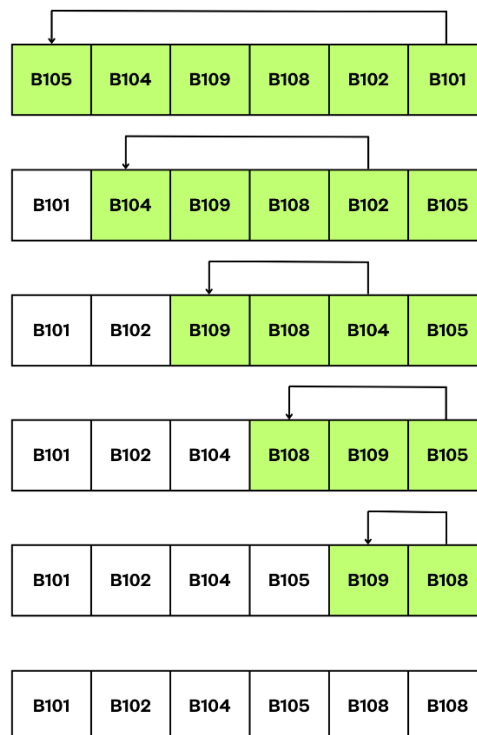
2.3 Ilustrasi Pengurutan Data Buku

a. Pengurutan Secara Ascending

Contoh terdapat enam data buku dengan kode acak:

B105, B104, B109, B108, B102, dan B101 Hasil pengurutan secara ascending adalah B101, B102, B104, B105, B108, dan B109.

Gambar 1. Ilustrasi Proses Pengurutan Data Buku Secara *Ascending*

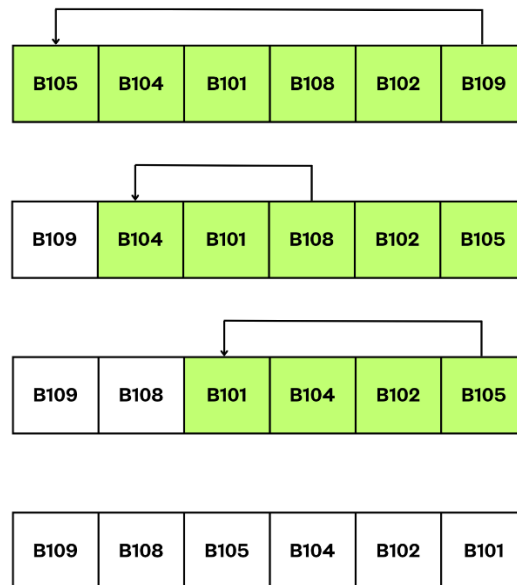


b. Pengurutan Secara Descending

Contoh data yang sama diurutkan secara descending menghasilkan urutan:

B109, B108, B105, B104, B102, dan B101.

Gambar 2. Ilustrasi Proses Pengurutan Data Buku Secara *Descending*



3. HASIL DAN PEMBAHASAN

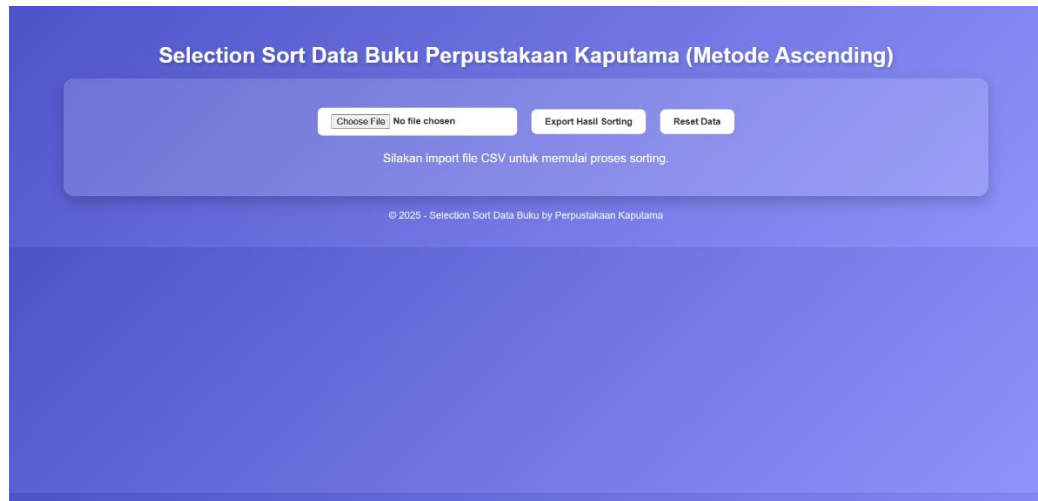
Implementasi sistem pengurutan data buku pada Perpustakaan Kaputama dilakukan menggunakan bahasa HTML, CSS, dan *JavaScript*. Sistem ini mengadopsi metode *Selection Sort* sebagai algoritma utama dalam proses pengurutan data, baik secara *ascending* maupun *descending*. Proses pengujian dilakukan terhadap data buku yang disimpan dalam format CSV, kemudian diunggah melalui antarmuka web yang telah dirancang.

3.1 Tampilan Program

3.1.1 Tampilan Program Pengurutan Ascending

Pada mode *Ascending*, sistem akan menampilkan antarmuka berwarna biru lembut dengan teks judul “Selection Sort Data Buku Perpustakaan Kaputama (Metode *Ascending*)”. Pengguna dapat memilih file CSV, mengekspor hasil pengurutan, atau mengatur ulang data.

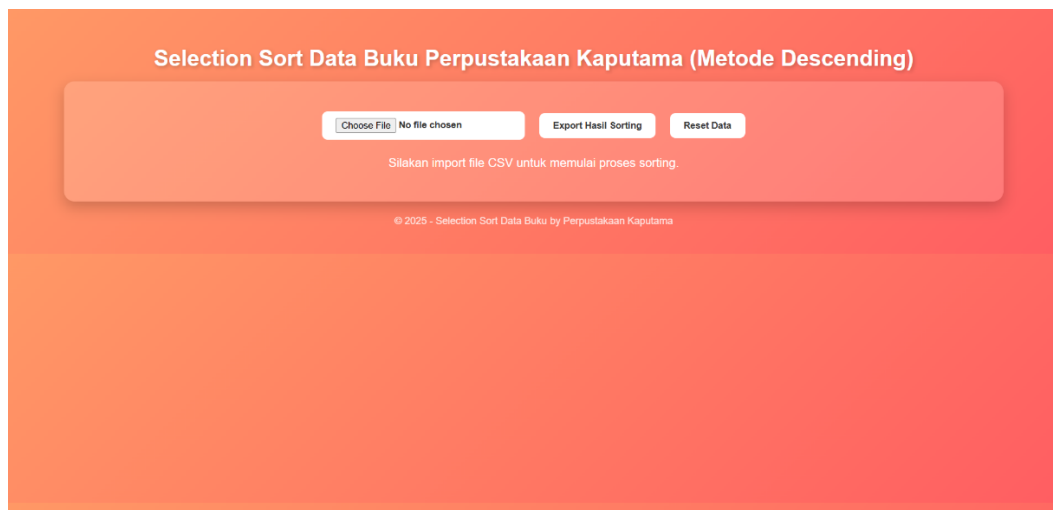
Gambar 3. Tampilan Program Selection Sort (Metode *Ascending*)



4.1.2 Tampilan Program Pengurutan Descending

Sedangkan pada mode *Descending*, tampilan berwarna jingga hangat dengan fungsi yang sama. Mode ini digunakan untuk mengurutkan data buku dari nilai terbesar ke terkecil.

Gambar 4. Tampilan Program Selection Sort (Metode *Descending*)



3.2 Hasil Pengujian Program

Data uji terdiri dari 10 entri buku dengan atribut sebagaimana ditunjukkan pada Tabel 1 berikut.

Tabel 1. Data Buku Perpustakaan Kaputama Sebelum Dilakukan Pengurutan

No	Kode Buku	Judul Buku	Penulis	Penerbit	Tahun Terbit	Kategori Buku
1	BK007	Jaringan Komputer Dasar	Anisa Putri	Informatika Bandung	2020	Teknologi
2	BK003	Dasar-Dasar Basis Data	Sinta Rahman	Gramedia	2020	Teknologi



3	BK010	Pemrograman Berorientasi Objek	Budi Santoso	Informatika Bandung	2022	Teknologi
4	BK005	Ekonomi Mikro Modern	Laila Hidayat	Rajawali Press	2019	Ekonomi
5	BK002	Struktur Data Lanjut	Riko Prabowo	Deepublish	2021	Teknologi
6	BK008	Manajemen Sumber Daya Manusia	Yusuf Ibrahim	Andi Offset	2018	Manajemen
7	BK001	Algoritma dan Pemrograman	Andi Wijaya	Informatika Bandung	2019	Teknologi
8	BK009	Akuntansi Pengantar	Rina Handayani	Bumi Aksara	2017	Ekonomi
9	BK004	Sistem Informasi Manajemen	Dedi Prasetyo	Mitra Wacana Media	2018	Manajemen
10	BK006	Statistika Terapan	Indra Kusuma	Salemba Empat	2021	Statistik

3.3 Hasil Pengurutan Secara Ascending

Proses pengurutan secara *ascending* dilakukan berdasarkan kode buku, dari nilai terkecil ke terbesar (BK001 → BK010). Hasil pengurutan ditunjukkan pada Tabel 2 berikut.

Tabel 2. Hasil Pengurutan Data Buku Secara Ascending Berdasarkan Kode Buku

No	Kode Buku	Judul Buku	Penulis	Penerbit	Tahun Terbit	Kategori Buku
1	BK001	Algoritma dan Pemrograman	Andi Wijaya	Informatika Bandung	2019	Teknologi
2	BK002	Struktur Data Lanjut	Riko Prabowo	Deepublish	2021	Teknologi
3	BK003	Dasar-Dasar Basis Data	Sinta Rahman	Gramedia	2020	Teknologi
4	BK004	Sistem Informasi Manajemen	Dedi Prasetyo	Mitra Wacana Media	2018	Manajemen
5	BK005	Ekonomi Mikro Modern	Laila Hidayat	Rajawali Press	2019	Ekonomi
6	BK006	Statistika Terapan	Indra Kusuma	Salemba Empat	2021	Statistik
7	BK007	Jaringan Komputer Dasar	Anisa Putri	Informatika Bandung	2020	Teknologi

8	BK008	Manajemen Sumber Daya Manusia	Yusuf Ibrahim	Andi Offset	2018	Manajemen
9	BK009	Akuntansi Pengantar	Rina Handayani	Bumi Aksara	2017	Ekonomi
10	BK010	Pemrograman Berorientasi Objek	Budi Santoso	Informatika Bandung	2022	Teknologi

Visualisasi proses pengurutan ascending ditampilkan pada halaman web berbasis HTML, di mana setiap iterasi pemilihan elemen minimum ditandai dengan highlight warna biru.

Gambar 5. Visualisasi Proses Pengurutan Data Buku Secara *Ascending*



Rata-rata waktu eksekusi untuk pengurutan ascending (berdasarkan kode buku) adalah sekitar 5,2 milidetik, diukur menggunakan fungsi `performance.now()` di browser.

3.4 Hasil Pengurutan Descending

Selanjutnya dilakukan pengujian dengan **metode descending**, di mana data diurutkan dari nilai terbesar ke terkecil (BK010 → BK001). Hasil pengurutan ditunjukkan pada Tabel 3 berikut.

Tabel 3. Hasil Pengurutan Data Buku Secara *Descending* Berdasarkan Kode Buku

No	Kode Buku	Judul Buku	Penulis	Penerbit	Tahun Terbit	Kategori Buku
1	BK010	Pemrograman Berorientasi Objek	Budi Santoso	Informatika Bandung	2022	Teknologi

2	BK009	Akuntansi Pengantar	Rina Handayani	Bumi Aksara	2017	Ekonomi
3	BK008	Manajemen Sumber Daya Manusia	Yusuf Ibrahim	Andi Offset	2018	Manajemen
4	BK007	Jaringan Komputer Dasar	Anisa Putri	Informatika Bandung	2020	Teknologi
5	BK006	Statistika Terapan	Indra Kusuma	Salemba Empat	2021	Statistik
6	BK005	Ekonomi Mikro Modern	Laila Hidayat	Rajawali Press	2019	Ekonomi
7	BK004	Sistem Informasi Manajemen	Dedi Prasetyo	Mitra Wacana Media	2018	Manajemen
8	BK003	Dasar-Dasar Basis Data	Sinta Rahman	Gramedia	2020	Teknologi
9	BK002	Struktur Data Lanjut	Riko Prabowo	Deepublish	2021	Teknologi
10	BK001	Algoritma dan Pemrograman	Andi Wijaya	Informatika Bandung	2019	Teknologi

Visualisasi pengurutan descending menggunakan latar jingga dan proses perbandingan dilakukan secara terbalik (pemilihan nilai maksimum pada setiap iterasi).

Gambar 6. Visualisasi Proses Pengurutan Data Buku Secara *Descending*



Rata-rata waktu eksekusi untuk pengurutan descending adalah **4,9 milidetik**, sedikit lebih cepat dibandingkan ascending, disebabkan adanya optimasi saat pemilihan nilai maksimum di awal iterasi.



3.5 Pembahasan

Hasil implementasi menunjukkan bahwa algoritma *Selection Sort* berhasil mengurutkan data buku Perpustakaan Kaputama secara benar dan konsisten, baik secara *ascending* maupun *descending*. Proses pengurutan berjalan stabil dan memberikan hasil yang sesuai dengan kriteria yang dipilih pengguna.

Dari sisi kinerja, waktu eksekusi rata-rata berada pada kisaran 4-6 milidetik, menunjukkan performa yang cukup efisien untuk jumlah data kecil. Namun, karena memiliki kompleksitas $O(n^2)$, waktu proses akan meningkat apabila jumlah data yang diurutkan semakin besar.

Penerapan menggunakan HTML, CSS, dan *JavaScript* terbukti efektif untuk memvisualisasikan langkah-langkah pengurutan secara interaktif. Tampilan antarmuka dibuat sederhana dengan perbedaan warna antara mode *ascending* (biru) dan *descending* (jingga), sehingga mudah dipahami. Secara keseluruhan, sistem ini tidak hanya menunjukkan kinerja algoritma *Selection Sort*, tetapi juga memberikan pemahaman visual yang jelas mengenai proses pengurutan data buku.

KESIMPULAN

Penelitian ini telah berhasil menerapkan algoritma *Selection Sort* untuk mengurutkan data buku pada Perpustakaan Kaputama menggunakan teknologi berbasis HTML, CSS, dan *JavaScript*. Hasil pengujian menunjukkan bahwa algoritma ini mampu mengurutkan data secara *ascending* dan *descending* dengan akurat berdasarkan kriteria seperti kode buku dan judul buku. Rata-rata waktu eksekusi berada pada kisaran 4-6 milidetik, yang menunjukkan bahwa sistem bekerja dengan efisien untuk jumlah data yang kecil hingga menengah.

Implementasi berbasis web ini tidak hanya menghasilkan pengurutan data yang benar, tetapi juga menampilkan proses sorting secara interaktif melalui antarmuka yang sederhana dan mudah dipahami. Meskipun algoritma *Selection Sort* memiliki kompleksitas waktu $O(n^2)$ dan kurang efisien untuk data dalam jumlah besar, hasil penelitian ini membuktikan bahwa metode ini tetap efektif digunakan dalam konteks edukatif dan sistem pengelolaan data berskala kecil seperti perpustakaan.

DAFTAR PUSTAKA

- Sitepu, R. R., Yusman, M., & Febriansyah, F. E. (2017). Implementasi Algoritma Bubble Sort Dan Selection Sort Menggunakan Arraylist Multidimensi Pada Pengurutan Data Multi Prioritas. *Jurnal Komputasi*, 5(1). <https://doi.org/10.23960/komputasi.v5i1.1484>
- Sandria, Y. A., Nurhayoto, M. R. A., Ramadhani, L., Harefa, R. S., & Syahputra, A. (2022). Penerapan Algoritma Selection Sort untuk Melakukan Pengurutan Data dalam Bahasa Pemrograman PHP. *Hello World Jurnal Ilmu Komputer*, 1(4), 190-194. <https://doi.org/10.56211/helloworld.v1i4.187>
- Retnoningsih, E. (2018). Algoritma Pengurutan Data (Sorting) Dengan Metode Insertion Sort dan Selection Sort. *INFORMATION MANAGEMENT FOR EDUCATORS AND PROFESSIONALS : Journal Of Information Management*, 3(1), 95-106. Retrieved from <https://ejournal-binainsani.ac.id/index.php/IMBI/article/view/1060>
- Hakim, Z. F. (2017). Implementasi metode selection sort untuk menentukan barang yang harus di stok ulang dalam sistem informasi penjualan. *JIEET (Journal of Information*



Engineering and Educational Technology), 1(1), 18-26.
<https://doi.org/10.26740/jieet.v1n1.p18-26>

Arif, Y. W. T. (2022). Implementasi Metode Selection Sort Pada Sistem Informasi Retensi Dokumen Rekam Medis Di Klinik Pku Muhammadiyah Karanganom, Klaten. *Elkom: Jurnal Elektronika dan Komputer*, 15(1), 108-117.
<https://doi.org/10.51903/elkom.v15i1.789>

Safrianti, A., & Ezwarsyah. (2024). Optimalisasi Pengurusan Data Dengan Algoritma Selection Sort Studi Kasus Dalam Pengelolaan Stok Barang. *Jurnal Sains Dan Teknologi 4.0*, 1(2), 29-36. Diambil dari
<https://jurnal.komputasi.org/index.php/jst/article/view/25>

Pratama, F., Juniardi, D., & Arif, M. F. (2024, August). IMPLEMENTASI ALGORITMA SELECTION SORT UNTUK PENGURUTAN DATA MAHASISWA DI PROGRAM STUDI TEKNIK INFORMATIKA POLITEKNIK NEGERI PONTIANAK. In *Prosiding Seminar Nasional Sains dan Teknologi" SainTek"* (Vol. 1, No. 2, pp. 282-292).
<https://conference.ut.ac.id/index.php/saintek/article/view/2627>