Jurnal SAINTEK Patompo
Publisher by Fakultas Sains dan Teknologi Universitas Patompo
p-ISSN : 3025-082X
e-ISSN : 3021-7113
Vol. 4 No. 2, Agustus 2026

Pengaruh Karakteristik Data Terhadap Efektivitas *Clustering* Pada Optimalisasi Kompleksitas Waktu Algoritma *Super Sort*

Raoda
Universitas Patompo, Makassar, Indonesia, missoda01@gmail.com

Abstract

Super Sort is a hybrid sorting algorithm that combines the concepts of Merge Sort and Quick Sort. Previous research identified a weakness in Super Sort, namely the large number of sublists generated when handling duplicate elements, which led to the development of Super Sort Clustering, a modification that groups identical values prior to sorting. That study showed Super Sort Clustering outperformed the original Super Sort in execution time on datasets with a high proportion of identical values. However, the effectiveness of this approach on data without identical values had not been empirically tested. This study evaluates the performance of Super Sort Clustering on large datasets without identical values (unique/random) with data sizes ranging from 100 to 500,000 elements. Testing measured the execution time of the original Super Sort and Super Sort Clustering under identical hardware and conditions. The results show that across all datasets without identical values produced higher execution times than both the original Super Sort. This occurs because the clustering step produces a number of clusters close to or equal to the number of original data points, so it does not reduce the number of elements to be sorted and merely adds computational overhead. It is therefore concluded that the Super Sort Clustering method cannot be used, or is not recommended, for data without identical values or evenly distributed data, and is only effective on datasets with a high proportion of duplicate values.

Keywords: Algorithm, Super Sort, Clustering, Time Complexity, Data Characteristics

Abstrak

Super Sort merupakan algoritma pengurutan hibrid yang menggabungkan konsep *Merge Sort* dan *Quick Sort*. Penelitian sebelumnya mengidentifikasi kelemahan *Super Sort* berupa banyaknya *sublist* yang terbentuk ketika menangani elemen duplikat, sehingga dikembangkan modifikasi *Super Sort Clustering* yang mengelompokkan nilai identik terlebih dahulu sebelum proses pengurutan dilakukan. Hasil penelitian tersebut menunjukkan bahwa *Super Sort Clustering* unggul secara waktu eksekusi dibandingkan *Super Sort* asli pada *dataset* dengan proporsi nilai identik yang tinggi. Namun, efektivitas pendekatan ini pada data yang tidak memiliki nilai identik belum diuji secara empiris. Penelitian ini bertujuan mengevaluasi kinerja *Super Sort Clustering* pada karakteristik *dataset* yang berukuran besar tanpa nilai identik (unik/acak) dengan variasi jumlah data 100 hingga 500.000 elemen. Pengujian dilakukan dengan mengukur waktu eksekusi *Super Sort* asli dan *Super Sort Clustering* pada perangkat dan kondisi yang sama. Hasil penelitian menunjukkan bahwa pada seluruh *dataset* tanpa nilai identik menghasilkan waktu eksekusi yang lebih tinggi dibandingkan *Super Sort* asli. Hal ini terjadi karena proses pengelompokan menghasilkan jumlah *cluster* yang mendekati atau sama dengan jumlah data asli, sehingga tidak mengurangi jumlah elemen yang perlu diurutkan dan hanya menambah *overhead* komputasi. Dengan demikian, dapat disimpulkan bahwa metode *Super Sort Clustering* tidak dapat digunakan atau tidak direkomendasikan untuk data yang tidak memiliki nilai identik atau data yang terdistribusi merata, dan hanya efektif diterapkan pada *dataset* dengan proporsi duplikasi nilai yang tinggi.

Kata Kunci: *Algoritma*, *Super Sort*, *Clustering*, Kompleksitas Waktu, Karakteristik Data

PENDAHULUAN

Algoritma pengurutan (*sorting*) merupakan salah satu komponen fundamental dalam ilmu komputer yang digunakan secara luas dalam berbagai proses komputasi, mulai dari pengolahan data hingga algoritma pencarian. Berbagai algoritma pengurutan telah dikembangkan dengan karakteristik kompleksitas waktu dan ruang yang berbeda-beda, seperti *Bubble Sort, Insertion Sort, Selection Sort, Quick Sort,* dan *Merge Sort*. Pemilihan algoritma yang tepat sangat bergantung pada karakteristik data yang akan diurutkan, termasuk ukuran data, tingkat keteraturan data sebelumnya, dan proporsi nilai duplikat di dalamnya.

Publisher by Fakultas Sains dan Teknologi Universitas Patompo
p-ISSN : 3025-082X
e-ISSN : 3021-7113
Vol. 4 No. 2, Agustus 2026

Super Sort merupakan algoritma pengurutan hibrid yang menggabungkan proses *Marge Sort* dan *Quick Sort* melalui tahapan *forward selection*, *backward selection*, *partition*, dan *merging*, dengan kompleksitas waktu kasus rata-rata O(n log n). Penelitian sebelumnya mengungkapkan bahwa *Super Sort* memiliki kelemahan pada *dataset* dengan banyak elemen duplikat, karena algoritma ini menghasilkan sejumlah besar *sublist* yang harus digabungkan kembali, sehingga meningkatkan biaya komputasi dan penggunaan memori. Untuk mengatasi kelemahan tersebut, dikembangkan modifikasi berupa *Super Sort Clustering*, yaitu penambahan tahap pengelompokan (*clustering*) nilai-nilai identik pada awal proses, sebelum data diurutkan menggunakan algoritma *Super Sort* asli. Pendekatan ini terbukti secara empiris mampu menurunkan waktu eksekusi secara signifikan pada *dataset* dengan proporsi nilai identik yang tinggi (Raoda & Ilham, 2024).

Meskipun demikian, penelitian tersebut turut mencatat sejumlah batasan secara teoretis, yaitu bahwa pendekatan *clustering* berpotensi tidak efektif atau bahkan menjadi beban tambahan (overhead) apabila diterapkan pada *dataset* yang seluruh elemennya bersifat unik, *dataset* berukuran kecil tanpa duplikat, maupun *dataset* dengan nilai yang tersebar merata. Namun, dugaan ini belum diuji secara empiris melalui eksperimen langsung. Padahal, pengetahuan mengenai batas efektivitas suatu metode sama pentingnya dengan pengetahuan mengenai keunggulannya, agar algoritma yang dikembangkan dapat diterapkan secara tepat sesuai karakteristik data yang dihadapi.

Selain pengembangan pada sisi algoritmik, penelitian mengenai optimalisasi pengurutan data juga berkembang pada arah lain, seperti implementasi paralel *Super Sort* menggunakan MPI dan CUDA yang terbukti unggul pada *dataset* berskala besar (Anaghashree et al., 2021), pengembangan *Enhanced Merge Sort* dengan deteksi dini bagian data yang sudah terurut untuk meningkatkan efisiensi (Abu Sara et al., 2019), serta algoritma *Right Bonding Sort* yang diklaim lebih cepat dibandingkan *Selection Sort, Bubble Sort*, dan *Insertion Sort* pada pengurutan array (Hasan et al., 2020). Pendekatan berbasis pembelajaran mesin turut mulai digunakan untuk menemukan strategi pengurutan baru, sebagaimana ditunjukkan oleh Mankowitz et al. (2023) yang berhasil menemukan algoritma pengurutan lebih cepat menggunakan deep reinforcement learning. Berbagai penelitian ini menegaskan bahwa efisiensi algoritma pengurutan senantiasa bergantung pada karakteristik data dan skenario penggunaannya, bukan bersifat mutlak berlaku pada semua kondisi.

Berdasarkan uraian tersebut, penelitian ini bertujuan untuk menguji secara empiris kinerja algoritma *Super Sort Clustering* pada data yang tidak memiliki nilai identik (unik/acak) pada skala data 100 hingga 500.000 elemen. Hasil pengujian dibandingkan dengan algoritma *Super Sort* asli untuk mengetahui apakah keunggulan *Super Sort Clustering* yang diperoleh pada penelitian sebelumnya masih berlaku, atau justru sebaliknya, tidak dapat digunakan pada karakteristik data tersebut.

METODE PENELITIAN

Penelitian ini merupakan penelitian eksperimen lanjutan (*extended experiment*) dari penelitian sebelumnya mengenai optimalisasi kompleksitas waktu algoritma *Super Sort* menggunakan pendekatan *clustering*. Algoritma *Super Sort Clustering* yang diuji tetap menggunakan proses kerja yang sama seperti pada penelitian sebelumnya, yaitu: (1) tahap *clustering*, yang mengelompokkan elemen data ke dalam struktur *key-value* berdasarkan kesamaan nilai; (2) ekstraksi *key* unik dari hasil pengelompokan; (3) pengurutan *key* menggunakan algoritma *Super Sort* asli (*forward selection*, *backward selection*, *partition*, dan *merging*); dan (4) rekonstruksi (*grouping/expand*) daftar terurut akhir berdasarkan *key* yang telah terurut.

Perbedaan utama penelitian ini dengan penelitian sebelumnya terletak pada karakteristik *dataset* uji. Jika penelitian sebelumnya berfokus pada *dataset* dengan proporsi nilai identik yang tinggi (data *Stack Overflow*), penelitian ini menguji karakteristik *dataset* besar tanpa nilai identik dengan variasi ukuran 100 hingga 500.000 elemen, di mana hampir tidak ada nilai yang berulang.

Publisher by Fakultas Sains dan Teknologi Universitas Patompo
p-ISSN : 3025-082X
e-ISSN : 3021-7113
Vol. 4 No. 2, Agustus 2026

Pengujian dilakukan dengan mengukur waktu eksekusi (dalam detik) dari algoritma *Super Sort* asli dan *Super Sort Clustering*, pada *dataset* dan kondisi yang tidak identik. Kompleksitas waktu masing-masing algoritma dianalisis menggunakan notasi *asimtotik Big-O*, yang secara umum digunakan untuk menggambarkan batas atas pertumbuhan waktu eksekusi suatu algoritma terhadap ukuran masukan (Badami, 2021; GeeksforGeeks, 2024). Seluruh pengujian dijalankan pada perangkat keras, lingkungan sistem, dan kerangka waktu yang sama untuk meminimalkan pengaruh variabel eksternal terhadap hasil pengukuran, sebagaimana prosedur yang digunakan pada penelitian sebelumnya. Hasil pengukuran waktu eksekusi kemudian dianalisis secara deskriptif-komparatif untuk melihat pola perubahan waktu terhadap pertambahan ukuran data, serta dibandingkan dengan hasil penelitian sebelumnya pada *dataset* dengan nilai identik untuk menarik kesimpulan mengenai batas efektivitas metode *Super Sort Clustering*.

HASIL PENELITIAN DAN PEMBAHASAN

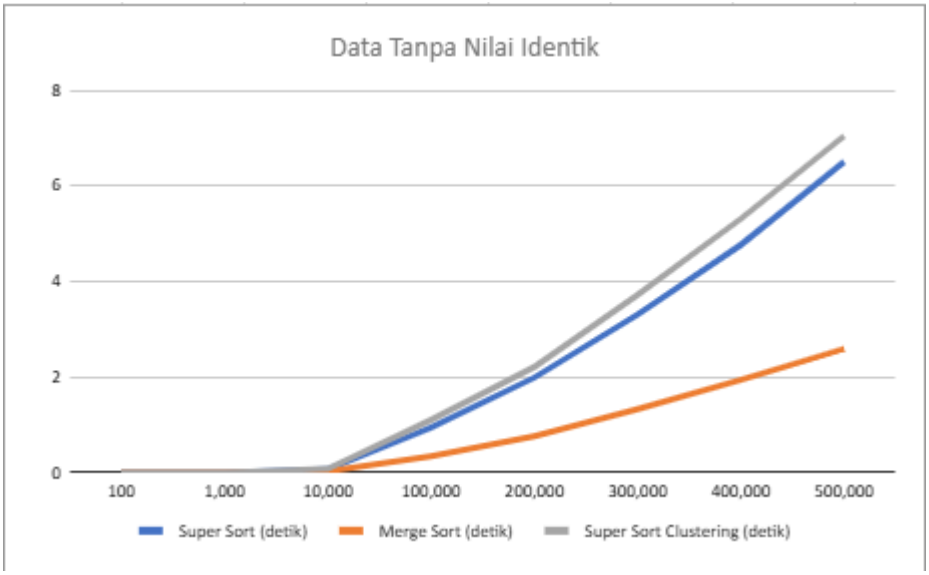
Clustering pada dasarnya merupakan teknik *unsupervised learning* yang bertujuan mengelompokkan data ke dalam beberapa kelompok berdasarkan kesamaan karakteristik tertentu, tanpa memerlukan label yang telah ditentukan sebelumnya (Kristian et al., 2023; Oyewole & Thopil, 2023). Berbagai taksonomi algoritma *clustering*, mulai dari pendekatan berbasis partisi hingga pendekatan berbasis deep learning, telah banyak dikembangkan untuk berbagai keperluan analisis data (Pitafi et al., 2023; Zhou et al., 2022). Namun demikian, manfaat *clustering* sangat bergantung pada ada tidaknya struktur kesamaan yang nyata di dalam data yang dianalisis; ketika data tidak memiliki struktur kesamaan yang berarti seperti pada *dataset* unik atau data yang terdistribusi merata, proses pengelompokan tidak akan menghasilkan kelompok yang bermakna.

Tabel 1. Waktu Eksekusi Pengurutan pada Data Besar tanpa Nilai Identik

Jumlah Data	*Super Sort* (ms)	*Super Sort Clustering* (s)
100	0.310	0.350
1000	4.120	4.890
10,000	72.140	81.670
100,000	943.260	1,104.580
200,000	1,987.610	2,213.540
300,000	3,304.740	3,717.830
400,000	4,765.410	5,312.890
500,000	6,502.840	7,043.210

Berdasarkan Tabel 1, terlihat bahwa *Super Sort Clustering* secara konsisten memiliki waktu eksekusi yang lebih tinggi dibandingkan *Super Sort* asli pada seluruh ukuran data yang diuji. Pada 500.000 elemen misalnya, *Super Sort Clustering* membutuhkan 7,043.210 detik, lebih lambat dibandingkan *Super Sort* asli (6,502.840 detik). Selisih waktu antara *Super Sort Clustering* dan *Super Sort* asli juga semakin membesar seiring bertambahnya jumlah data, menunjukkan bahwa tahap *clustering* menjadi beban tambahan yang terus meningkat, bukan penghemat komputasi seperti pada data dengan nilai identik.

Jurnal SAINTEK Patompo
Publisher by Fakultas Sains dan Teknologi Universitas Patompo
p-ISSN : 3025-082X
e-ISSN : 3021-7113
Vol. 4 No. 2, Agustus 2026



Gambar 1. Grafik Waktu Eksekusi Pengurutan pada Data Besar tanpa Nilai Identik

Hasil pengujian pada karakteristik *dataset* di atas menunjukkan pola yang konsisten dan berbanding terbalik dengan temuan pada penelitian sebelumnya terhadap *dataset* dengan proporsi nilai identik yang tinggi. Pada *dataset* dengan banyak nilai identik, tahap *clustering* mampu mengelompokkan sejumlah besar elemen ke dalam sedikit key unik, sehingga proses pengurutan *Super Sort* selanjutnya hanya perlu dilakukan terhadap jumlah key yang jauh lebih sedikit dibandingkan jumlah data asli. Sebaliknya, pada *dataset* tanpa nilai identik setiap elemen cenderung membentuk cluster tersendiri, sehingga jumlah key yang dihasilkan hampir sama dengan jumlah data asli. Akibatnya, proses *Super Sort* pada tahap berikutnya tetap harus memproses jumlah elemen yang hampir sama besar seperti tanpa *clustering*, sementara waktu yang telah digunakan untuk membangun struktur pengelompokan (kompleksitas O(n)) tidak memberikan pengurangan beban komputasi yang berarti, melainkan menjadi tambahan biaya eksekusi murni.

Temuan ini memperkuat sekaligus membuktikan secara empiris batasan yang telah diidentifikasi secara teoretis pada penelitian sebelumnya, yaitu bahwa algoritma *Super Sort Clustering* tidak cocok digunakan pada *dataset* tanpa elemen duplikat, *dataset* berukuran kecil dengan nilai unik, maupun *dataset* dengan nilai yang tersebar merata. Dengan demikian, pemilihan penggunaan algoritma *Super Sort Clustering* perlu mempertimbangkan karakteristik data yang akan diurutkan, khususnya proporsi nilai duplikat di dalamnya, dan bukan semata-mata berdasarkan ukuran *dataset*.

PENUTUP

Berdasarkan hasil pengujian dan pembahasan, dapat disimpulkan sebagai berikut:

1. Efektivitas algoritma *Super Sort Clustering* sangat bergantung pada karakteristik data, khususnya proporsi nilai duplikat (identik) di dalam *dataset*, dan bukan pada ukuran *dataset* semata.
2. Pada *dataset* yang tidak memiliki nilai identik (unik/acak), algoritma *Super Sort Clustering* justru menghasilkan waktu eksekusi yang lebih tinggi dibandingkan *Super Sort* asli, sehingga metode ini tidak dapat digunakan atau tidak direkomendasikan untuk karakteristik data tersebut.

Penelitian selanjutnya disarankan untuk mengembangkan mekanisme adaptif yang mampu mendeteksi proporsi nilai duplikat pada suatu *dataset* secara otomatis sebelum memutuskan apakah tahap *clustering* perlu diterapkan atau tidak, sehingga algoritma *Super Sort* dapat menyesuaikan strategi pengurutannya sesuai karakteristik data yang dihadapi. Selain itu, perlu dieksplorasi teknik pengelompokan alternatif dengan overhead komputasi yang lebih rendah agar tetap dapat memberikan manfaat meskipun diterapkan pada *dataset* dengan proporsi duplikasi yang rendah.

Publisher by Fakultas Sains dan Teknologi Universitas Patompo
p-ISSN : 3025-082X
e-ISSN : 3021-7113
Vol. 4 No. 2, Agustus 2026

DAFTAR PUSTAKA

Abu Sara, M. R., Klaib, M. F. J., & Hasan, M. (2019). EMS: An enhanced Merge Sort algorithm by early checking of already sorted parts. International Journal of Software Engineering and Computer Systems, 5(2), 15–25. https://doi.org/10.15282/ijsecs.5.2.2019.2.0058

Anaghashree, Pereira, S. D., Ashwath, R. B., Rai, S., & Kini, N. G. (2021). *Super Sort* algorithm using MPI and CUDA. Advances in Intelligent Systems and Computing, 1177, 165–170. https://doi.org/10.1007/978-981-15-5679-1_16

Badami, V. A. (2021). Time complexity analysis. Medium. https://medium.com/@vittal.a.badami0107/time-complexity-analysis-2fe312bfcfa3

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms (3rd ed.). MIT Press.

GeeksforGeeks. (2024). Worst, average and best case analysis of algorithms. https://www.geeksforgeeks.org/worst-average-and-best-case-analysis-of-algorithms/

Gugale, Y. (2018). *Super Sort* sorting algorithm. Proceedings of the 2018 3rd International Conference for Convergence in Technology (I2CT). https://doi.org/10.1109/I2CT.2018.8529601

Hasan, M., Bijoy, M. H. I., Hasan, M. R., & Rabbani, M. (2020). RBS: A new comparative and better solution of sorting algorithm for array. Proceedings of IEEE.

Jimmy, Utama, F., Felix, & Laia, A. P. (2021). Aplikasi pembelajaran penyortiran menggunakan algoritma *Super Sort* berbasis mobile. Jurnal Ilmiah Mikroskil.

Kristian, R., Defit, S., & Sumijan. (2023). Metode k-means *clustering* untuk mengukur tingkat kedisiplinan pegawai (studi kasus di pemerintah kabupaten Padang Pariaman). Jurnal CoSciTech (Computer Science and Information Technology), 4(1), 116–125. https://doi.org/10.37859/coscitech.v4i1.4728

Mankowitz, D. J., Michi, A., Zhernov, A., Gelmi, M., Selvi, M., Paduraru, C., Leurent, E., Iqbal, S., Lespiau, J.-B., Ahern, A., Köppe, T., Millikin, K., Gaffney, S., Elster, S., Broshear, J., Gamble, C., Milan, K., Tung, R., Hwang, M., … Hassabis, D. (2023). Faster sorting algorithms discovered using deep reinforcement learning. Nature, 618(7964), 257–263. https://doi.org/10.1038/s41586-023-06004-9

Marco Caraig, J., Anne Marie Cruz, N., Agustin, V., Mata, K., & Cortez, D. M. (2022). Enhancement of *Super Sort* sorting algorithm using identical value search concept. International Journal of Research Publications, 101(1). https://doi.org/10.47119/ijrp1001011520223223

Olukanmi, P., Popoola, P., & Olusanya, M. (2020). Centroid sort: A *clustering*-based technique for accelerating sorting algorithms. Proceedings of the 2020 2nd International Multidisciplinary Information Technology and Engineering Conference (IMITEC). https://doi.org/10.1109/IMITEC50163.2020.9334102

Oyewole, G. J., & Thopil, G. A. (2023). Data *clustering*: Application and trends. Artificial Intelligence Review, 56(7), 6439–6475. https://doi.org/10.1007/s10462-022-10325-y

Pitafi, S., Anwar, T., & Sharif, Z. (2023). A taxonomy of machine learning *clustering* algorithms, challenges, and future realms. Applied Sciences, 13(6), 3529. https://doi.org/10.3390/app13063529

Raoda, & Ilham, A. A. (2025). Optimalisasi kompleksitas waktu pengurutan data menggunakan metode algoritma *Super Sort* [Tesis, Universitas Hasanuddin]. Dipublikasikan sebagian

Publisher by Fakultas Sains dan Teknologi Universitas Patompo
p-ISSN : 3025-082X
e-ISSN : 3021-7113
Vol. 4 No. 2, Agustus 2026

pada International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA) 2024.

Rizkyatul Basir, R. (2020). Analisis kompleksitas ruang dan waktu terhadap laju pertumbuhan algoritma Heap Sort, Insertion Sort, dan Merge Sort dengan pemrograman Java. STRING (Satuan Tulisan Riset dan Inovasi Teknologi).

Yadav, R., Yadav, R., & Gupta, S. B. (2020). Artificial intelligence and speech technology.

Yerram, B., & Bhonagiri, J. K. (2020). An efficient sorting algorithm for binary data. Proceedings of IEEE.

Chauhan, Y., & Duggal, A. (2020). Different sorting algorithms comparison based upon the time complexity. International Journal of Research and Analytical Reviews, 7(3).

Zhou, S., Xu, H., Zheng, Z., Chen, J., Li, Z., Bu, J., Wu, J., Wang, X., Zhu, W., & Ester, M. (2022). A comprehensive survey on deep *clustering*: Taxonomy, challenges, and future directions. arXiv. http://arxiv.org/abs/2206.07579