

**ANALISIS PERBANDINGAN KECEPATAN ALGORITMA SELECTION SORT
DAN BUBBLE SORT****Nanang Mahrozi, Muhammad Faisal**Magister Informatika, Universitas Islam Negeri Maulana Malik Ibrahim, Malang, Indonesia
Alamat e-mail : nanang mahrozi18@gmail.com, mfaisal@ti.uin-malang.ac.id**Abstract**

Algorithms play a crucial role in software development and programming. Many programming paradigms have been developed to understand the basic concepts of algorithms, both general and specific. Data processing is essential in the sorting algorithm, especially in the context of selection sort and bubble sort. Both algorithms are designed to sort data with integer types. Each type of algorithm has different levels of effectiveness. The effectiveness of an algorithm can be measured based on the time and memory space needed to execute it. An efficient algorithm is one that optimizes the use of time and space minimally. The fewer resources needed to run an algorithm, the more effective it is considered. It is important to note that the time and space required by an algorithm are influenced by the amount of data processed and the type of algorithm applied, which is the main focus of this article, with an emphasis on the time complexity of various types of algorithms. The algorithms described in this study are implemented using the Python programming language in the Google Colab environment. Referring to the logic of the data sorting process using algorithms such as Selection Sort and Bubble Sort, it can be concluded that the selection sort algorithm has a speed advantage over bubble sort, as evidenced by testing with 100-100,000 data where selection sort is consistently faster.

Article History

Submitted: 3 November 2023
Accepted: 12 November 2023
Published: 13 November 2023

Key Words

Algorithm Complexity,
Sorting, Selection Sort,
Bubble Sort.

Abstrak

Algoritma memainkan peran yang sangat penting dalam pengembangan perangkat lunak dan pemrograman. Banyak logika pemrograman telah dikembangkan untuk memahami konsep dasar algoritma, baik yang bersifat umum maupun yang bersifat khusus. Pemrosesan data berperan penting dalam pengurutan algoritma, khususnya dalam konteks selection sort (pengurutan dengan pemilihan) dan bubble sort (pengurutan gelembung). Kedua algoritma ini dirancang untuk mengurutkan data dengan tipe integer. Setiap jenis algoritma memiliki tingkat efektivitas yang berbeda. Efektivitas suatu algoritma dapat diukur berdasarkan waktu dan ruang memori yang dibutuhkan untuk mengeksekusinya. Algoritma yang efisien adalah algoritma yang dapat mengoptimalkan penggunaan waktu dan ruang secara minimal. Semakin sedikit sumber daya yang dibutuhkan untuk menjalankan algoritma, semakin efektif algoritma tersebut dianggap. Perlu diingat bahwa waktu dan ruang yang diperlukan oleh suatu algoritma dipengaruhi oleh Jumlah data yang diolah dan jenis algoritma yang diterapkan menjadi fokus utama dalam artikel ini, dengan penekanan pada kompleksitas waktu dari berbagai jenis algoritma. Algoritma-algoritma yang dijelaskan dalam penelitian ini diimplementasikan dengan menggunakan bahasa pemrograman python dalam lingkungan google colab. Mengacu pada logika dari proses pengurutan data menggunakan algoritma seperti Selection Sort dan Bubble Sort dapat disimpulkan bahwa algoritma

Sejarah Artikel

Submitted: 3 November 2023
Accepted: 12 November 2023
Published: 13 November 2023

Kata Kunci

Kompleksitas Algoritma,
Pengurutan, Selection Sort,
Bubble Sort.

selection sort memiliki keunggulan kecepatan dibanding bubble sort, terbukti dari pengujian 100-100000 data selection sort selalu lebih cepat.

Pendahuluan

Utami[4] sorting disebut sebagai proses pengaturan data sesuai dengan aturan tertentu, sehingga data tersebut disusun dalam urutan yang sesuai dengan aturan yang ditetapkan. Secara umum, terdapat dua aturan pengurutan yang umum digunakan, yaitu urutan menaik (ascending) yang melibatkan pengurutan data dari nilai terkecil hingga terbesar, dan urutan menurun (descending) yang melibatkan pengurutan data dari nilai terbesar hingga terkecil. Proses pengurutan data melibatkan perbandingan data dan pertukaran data.

Dalam berbagai konteks pemrograman, algoritma adalah spesifikasi dari urutan langkah-langkah yang harus diikuti untuk menyelesaikan tugas tertentu. Pentingnya sebuah algoritma tidak hanya terletak pada kemampuannya untuk menghasilkan solusi yang benar, namun juga sejauh mana algoritma tersebut dapat menghasilkan output yang diinginkan berdasarkan serangkaian input yang diberikan. Dalam hal ini, keefektifan dan keefisiensi algoritma menjadi krusial. Bahkan jika algoritma tampak sangat canggih, jika hasilnya keliru, algoritma tersebut tidak dapat dianggap sebagai algoritma yang baik.

Dengan demikian, sebuah algoritma yang baik harus memenuhi beberapa kriteria penting, yaitu efektifitas, efisiensi, akurasi, dan struktur yang terorganisir. Untuk menilai kualitas suatu algoritma, kita dapat mengukurnya berdasarkan waktu eksekusi algoritma dan kebutuhan memori yang diperlukan. Dengan kata lain, sebaik apa pun algoritma tersebut, kualitasnya dapat diukur oleh seberapa cepat algoritma tersebut menyelesaikan tugasnya dan seberapa sedikit memori yang diperlukan dalam prosesnya.

Algoritma yang dianggap efisien adalah algoritma yang dapat mengurangi sebanyak mungkin penggunaan waktu dan ruang. Untuk mengukur dan menggambarkan konsep ini, digunakan istilah "kompleksitas algoritma." Namun, penting untuk dipahami bahwa waktu dan ruang yang dibutuhkan oleh suatu algoritma sangat bergantung pada dua faktor utama, Artikel ini memusatkan perhatian pada dua aspek utama, yakni jumlah data yang diolah dan jenis algoritma yang diterapkan.

Kompleksitas waktu, diwakili sebagai $T(n)$, merujuk pada jumlah operasi yang dilakukan oleh algoritma seiring dengan ukuran masukan n . Dalam mengevaluasi kompleksitas waktu, kita mencatat berapa banyak operasi yang dijalankan oleh algoritma. Untuk menggambarkan kompleksitas waktu algoritma, sering digunakan notasi Big-O (O besar). Dalam konteks notasi Big-O, algoritma akan memiliki waktu asimptotik yang berhubungan dengan fungsi $f(n)$.

Definisi notasi Big-O adalah bahwa jika suatu algoritma memiliki kompleksitas waktu asimptotik $O(f(n))$, maka ketika nilai n semakin besar, waktu yang dibutuhkan oleh algoritma tersebut tidak akan pernah melebihi suatu konstanta C dikalikan dengan $f(n)$. Dengan kata lain, $f(n)$ adalah batas atas dari waktu eksekusi algoritma untuk nilai n yang cukup besar. Ini adalah cara untuk mengukur pertumbuhan waktu algoritma secara relatif terhadap ukuran masukan n . Notasi

$O(n)$ sering digunakan untuk menghitung jumlah operasi perbandingan yang dilakukan oleh algoritma tersebut.

Dengan demikian, pengukuran kompleksitas waktu algoritma menggunakan notasi Big-O membantu kita memahami bagaimana performa algoritma berubah seiring penambahan ukuran masukan, dan ini sangat berguna dalam analisis dan perbandingan algoritma yang berbeda.

Sebuah "sorting algorithm," yang merupakan salah satu algoritma dasar yang sering digunakan untuk menyelesaikan berbagai masalah, adalah operasi yang sangat penting dalam pemrosesan data. Proses pengurutan data menjadi kunci dalam pengolahan data, pentingnya pengurutan data sangat signifikan. Hingga saat ini, berbagai metode pengurutan data telah dikembangkan, dan kemungkinan adanya metode-metode baru terus muncul seiring berjalannya waktu.

Dari penelitian sebelumnya, terungkap bahwa penggunaan algoritma Bubble Sort memiliki keunggulan dalam kesederhanaan maka lebih mudah dipahami, khususnya dalam situasi dengan jumlah data yang relatif kecil, Selection Sort dapat menjadi pilihan yang lebih baik daripada Bubble Sort (Rizki Saputra, Andryana and Sholihati, 2021). Pada penelitian berikutnya, terdapat temuan bahwa jumlah iterasi pada Selection Sort pada setiap pengujian data lebih rendah dibandingkan dengan Bubble Sort. Selain itu, waktu eksekusi Selection Sort pada setiap pengujian data juga terbukti lebih cepat dibandingkan dengan waktu eksekusi Bubble Sort. (Roma Rio, Yusman and Febi Eka, 2017).

Dalam penelitian ini, untuk menjaga fokus dan ruang lingkup pembahasan yang terbatas, hanya dua metode pengurutan akan dibahas, yaitu penelitian akan terfokus pada selection sort dan bubble sort analisis efisiensi masing-masing algoritma dengan melakukan pengujian waktu eksekusi keduanya. Pengujian dilakukan dengan menggunakan satu set data dalam bentuk bilangan bulat (integer) yang disusun dalam bentuk list atau array secara acak.

Diketahui bahwa kedua algoritma ini memiliki kecepatan waktu yang berbeda, dengan beberapa lebih cepat daripada yang lain dalam proses pengurutan. Dengan demikian, untuk memahami perbandingan waktu eksekusi kedua algoritma tersebut, penting menggunakan bahasa pemrograman sebagai alat untuk mengukur waktu eksekusi. dalam kerangka penelitian ini, bahasa pemrograman yang dipakai adalah Python. yang dijalankan dalam lingkungan Google Colab. Alasan penggunaan Google Colab adalah karena bahasa pemrograman ini sederhana dan mudah dipelajari, sehingga cocok untuk tujuan eksperimen dan analisis dalam konteks penelitian ini.

Metode Penelitian

Algoritma adalah konsep fundamental dalam dunia pemrograman dan ilmu komputer. Ini adalah langkah-langkah terinci yang merinci cara menyelesaikan suatu masalah atau tugas tertentu. Algoritma digunakan untuk menggambarkan cara-cara yang terstruktur dan sistematis untuk mencapai suatu tujuan atau solusi dalam konteks yang dapat diimplementasikan dalam bahasa pemrograman.

◆ Terdapat berbagai jenis algoritma, seperti algoritma pengurutan (seperti Selection Sort, Bubble Sort, Quick Sort), algoritma pencarian (seperti Binary Search, Linear Search), algoritma kecerdasan buatan (seperti algoritma genetika, algoritma jaringan saraf tiruan), dan banyak lagi.

Pemahaman yang kuat tentang algoritma adalah kunci untuk menjadi seorang pengembang perangkat lunak yang kompeten dan efisien. Algoritma membantu menguraikan permasalahan yang kompleks menjadi solusi yang dapat diimplementasikan melalui kode pemrograman.

2.1. Selection Sort

Selection Sort adalah salah satu algoritma pengurutan sederhana yang beroperasi dengan cara memilih elemen terkecil dari daftar dan menukarnya dengan elemen pertama. Kemudian, ia akan memilih elemen terkecil dari sisa daftar dan menukarnya dengan elemen kedua, dan seterusnya, hingga seluruh daftar terurut.

Yahya (2014:136) menjelaskan bahwa selection sort merupakan metode pengurutan yang membandingkan elemen yang sedang diproses dengan elemen-elemen berikutnya hingga mencapai elemen terakhir. Jika ditemukan elemen yang lebih kecil dari elemen yang sedang diproses, maka posisi elemen tersebut akan dicatat, dan elemen tersebut akan ditukar dengan elemen yang sedang diproses.

Metode selection sort melibatkan pemilihan elemen dengan nilai terkecil dan menukar posisinya dengan elemen pertama dalam daftar. Kemudian, proses perbandingan antara elemen yang sedang diproses dengan elemen berikutnya akan berlanjut hingga mencapai elemen terakhir. Proses perbandingan akan terus berlangsung hingga tidak ada lagi pertukaran data yang diperlukan.

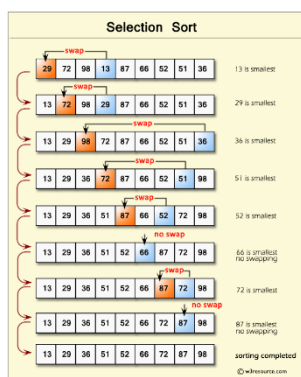


Figure 1. Contoh ilustrasi proses kerja Selection sort

Secara garis besar metode ini dijelaskan sebagai berikut :

Pertama dalam contoh ini, kita memiliki array dengan 9 elemen yang belum diurutkan. Algoritma Selection sort akan mencari elemen terkecil di dalam array, di mana dalam kasus ini elemen terkecil adalah 13. Setelah menemukan elemen terkecil, Selection sort akan menukarnya ke posisi pertama, sehingga elemen 29 dan 13 akan ditukar. Setelah berhasil melakukan pertukaran pertama, Selection sort akan memindahkan elemen 29 ke posisi kedua.

Proses ini akan terus diulang hingga semua elemen dalam array berhasil diurutkan secara berurutan. Dalam setiap iterasi, algoritma Selection sort akan mencari elemen terkecil yang tersisa

dan memindahkannya ke posisi yang sesuai, sehingga array menjadi terurut secara bertahap hingga selesai.

2.2. Bubble Sort

Bubble Sort merupakan algoritma pengurutan lainnya yang bekerja dengan membandingkan pasangan elemen berdekatan dalam daftar dan menukarkan posisinya jika elemen-elemen tersebut tidak dalam urutan yang benar. Proses ini diulang berulang-ulang hingga seluruh daftar terurut.

Yahya (2014:136) menjelaskan bahwa menurutnya, bubble sort adalah teknik pengurutan yang membandingkan elemen saat ini dengan elemen berikutnya dalam daftar. Jika elemen saat ini lebih besar dari elemen berikutnya, maka posisi keduanya akan ditukar. Jika tidak, tidak ada perubahan yang diperlukan. Misalnya, jika terdapat 7 elemen ($n = 7$), maka proses bubble sort akan mengalami 6 tahapan (dimulai dari 0 hingga $n - 2$).

Pada alur algoritma bubble sort akan membandingkan antara setiap pasangan elemen, dan jika ditemukan elemen yang acak atau tidak sesuai urutan dan tidak berada pada tempat yang benar, maka dilakukan pertukaran. Proses perbandingan ini akan terus berlanjut hingga tidak ada lagi pertukaran data yang diperlukan.

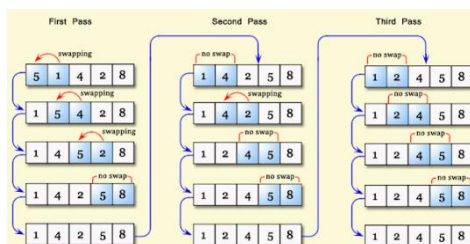


Figure 2. Contoh ilustrasi proses kerja Bubble sort

2.3. Python dan google colab

Pemrograman Python adalah suatu bentuk pemrograman komputer yang menggunakan bahasa pemrograman Python. Python merupakan bahasa pemrograman tingkat tinggi yang terkenal karena sintaksisnya yang mudah dibaca dan mudah dipahami. Ini digunakan secara luas dalam berbagai bidang seperti pengembangan perangkat lunak, analisis data, kecerdasan buatan, pengembangan web, dan banyak lagi.

1. Sintaksis yang Mudah Dipahami Python menggunakan sintaksis yang mirip dengan bahasa Inggris, yang membuatnya mudah dipahami oleh pengembang pemula dan berpengalaman.
2. Dukungan untuk Berbagai Bidang Python memiliki berbagai pustaka dan modul yang mendukung pengembangan perangkat lunak, analisis data, pemrosesan teks, pengembangan web, kecerdasan buatan, dan banyak lagi.
3. Komunitas yang Besar Python memiliki komunitas pengembang yang besar, sehingga Anda dapat dengan mudah menemukan bantuan dan dukungan dari sesama pengembang.
4. Platform-Agnostik Python dapat berjalan di berbagai platform seperti Windows, macOS, dan Linux.

5. Open Source Python adalah perangkat lunak sumber terbuka, yang berarti Anda dapat menggunakannya secara gratis dan mengubahnya sesuai kebutuhan Anda.
6. Dokumentasi yang Kaya. Python memiliki dokumentasi yang sangat baik dan tutorial online yang dapat membantu pengembang memahami dan menggunakan bahasa ini.

Python telah menjadi pilihan bahasa pemrograman yang terkenal dan digunakan di seluruh dunia, baik oleh pemula maupun oleh perusahaan teknologi besar. Itu juga menjadi bahasa yang populer dalam pengembangan solusi kecerdasan buatan dan pengolahan data, karena memiliki banyak pustaka dan alat yang mendukung analisis data yang kuat.

Sedangkan Google Colab, juga dikenal sebagai Colaboratory, adalah platform cloud yang disediakan oleh Google untuk pengembangan dan eksekusi kode Python. Ini adalah sebuah layanan gratis yang memungkinkan user untuk membuat, menjalankan, dan berbagi notebook interaktif yang berisi kode Python. Google Colab sangat populer di kalangan pengembang, peneliti, dan pembelajar karena memiliki banyak keunggulan.

Google Colab memberikan akses kepada banyak orang untuk belajar pemrograman Python, pengembangan machine learning, analisis data, dan berbagai jenis proyek teknis tanpa perlu menginstal perangkat lunak secara lokal. Dengan dukungan sumber daya komputasi yang kuat, ini juga sangat berguna untuk eksperimen machine learning dan riset data.

Hasil dan Pembahasan

Tujuan dari pengujian ini adalah untuk membandingkan performa eksekusi antara metode Selection Sort dan Bubble Sort terhadap berbagai ukuran data. Selection Sort dan Bubble Sort merupakan dua algoritma pengurutan yang berbeda dalam pendekatan dan tingkat kompleksitasnya. Penelitian ini mengukur waktu yang diperlukan setiap masing-masing algoritma untuk mengurutkan data dengan beragam ukuran, termasuk 100, 1000, 10000, dan 100000 elemen, menggunakan lingkungan Python di Google Colab.

Hasil pengukuran waktu ini akan memberikan sebuah gambaran yang lebih baik tentang seberapa efisien masing-masing algoritma dalam menangani berbagai volum data. Oleh karena itu, penelitian ini memberikan wawasan yang bermanfaat dalam pemilihan algoritma pengurutan yang paling cocok untuk keperluan tertentu, terutama ketika berurusan dengan data dalam skala besar.

3.1. Algoritma Sort Code

```
import random
import time

def selection_sort(arr):
    n = len(arr)
    for i in range(n-1):
        min_index = i
        for j in range(i+1, n):
            if arr[j] < arr[min_index]:
                min_index = j
        arr[i], arr[min_index] = arr[min_index], arr[i]
```

Figure 3. Algoritma selection sort code

```
def bubble_sort(arr):
    n = len(arr)
    for i in range(n-1):
        swapped = False
        for j in range(n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
                swapped = True
        if not swapped:
            break
```

Figure 3. Algoritma bubble sort code

Dari hasil source code di atas kemudian dimasukkan beberapa percobaan data yaitu 100,1000,10000,100000 dan mendapatkan hasil sebagai berikut :

```
Selection Sorted array: [27, 59, 62, 66, 66, 77, 77]
Selection Sort took 0.50 milliseconds

Bubble Sorted array: [27, 59, 62, 66, 66, 77, 77, 79]
Bubble Sort took 0.89 milliseconds
```

Figure 4. Hasil perhitungan 100 data

```
Selection Sorted array: [1, 1, 1, 2, 3, 3, 7]
Selection Sort took 45.41 milliseconds

Bubble Sorted array: [1, 1, 1, 2, 3, 3, 7, 9]
Bubble Sort took 90.36 milliseconds
```

Figure 5. Hasil perhitungan 1000 data

```
Selection Sorted array: [1, 1, 1, 1, 1, 1, 1, 1]
Selection Sort took 4567.21 milliseconds

Bubble Sorted array: [1, 1, 1, 1, 1, 1, 1, 1, 1]
Bubble Sort took 11185.43 milliseconds
```

Figure 6. Hasil perhitungan 10000 data

```
Selection Sorted array: [1, 1, 1, 1, 1, 1, 1, 1]
Selection Sort took 507401.96 milliseconds

Bubble Sorted array: [1, 1, 1, 1, 1, 1, 1, 1, 1]
Bubble Sort took 1161282.07 milliseconds
```

Figure 7. Hasil perhitungan 100000 data

Hasil dari perhitungan selection sort:

Table 1. Perhitungan selection sort

Data	Waktu
100	0,50 ms
1000	45,41 ms
10000	4567,21 ms
100000	507401,96 ms

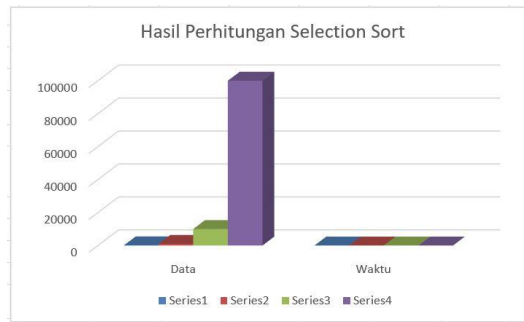


Figure 8. Grafik hasil perhitungan algoritma selection sort

Hasil dari perhitungan bubble sort:

Table 2. Perhitungan bubble sort

Data	Waktu
100	0,89 ms
1000	90,36 ms
10000	1185.43 ms
100000	1161282.07 ms

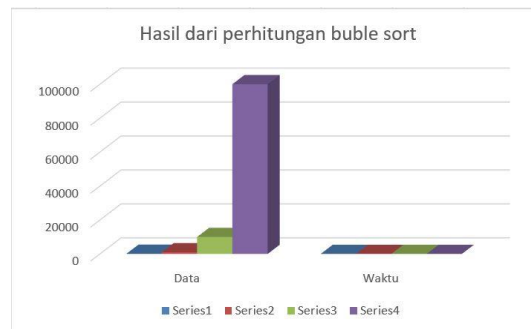


Figure 9. Grafik hasil perhitungan algoritma bubble sort

Kesimpulan

Memang benar bahwa teknik-teknik pengurutan data memiliki beragam pendekatan, dan di antara mereka, terdapat metode yang dapat dianggap sebagai yang paling cepat. Setiap metode pengurutan memiliki kelebihan dan kelemahan masing-masing. Terlebih lagi, dalam pengembangan program komputer, algoritma memiliki peran kunci. Hal ini disebabkan karena kompleksitas program yang tinggi. Meskipun teoretisnya mungkin memungkinkan untuk membuat program tanpa menggunakan algoritma, namun dalam praktiknya, program semacam itu mungkin akan memiliki akses yang terbatas atau memerlukan penggunaan penyimpanan yang besar.

Berdasarkan dasar logika tersebut, proses penyusunan data berdasarkan algoritma seperti selection sort dan bubble sort menjadi penting dalam pengembangan program komputer. Algoritma-algoritma ini membantu memastikan bahwa data diatur secara efisien, memungkinkan program untuk berjalan dengan baik dan mengoptimalkan penggunaan sumber daya, seperti waktu dan penyimpanan. Berdasarkan logika dari proses penyusunan data dengan algoritma seperti Selection Sort dan Bubble Sort dapat disimpulkan bahwa algoritma selection sort memiliki keunggulan

kecepatan dibanding bubble sort, terbukti dari pengujian 100-100000 data selection sort selalu lebih cepat. Dengan rincian dari pengujian 100 data, selection sort lebih cepat 0,39m/s sedangkan pada 100000 data selection sort lebih cepat 653880,04 m/s.

Referensi

From journal article:

Rizki Saputra, D.Y., Andryana, S. and Sholihati, I.D. (2021) ‘Analisis Perbandingan Algoritma Bubble Sort Dan Selection Sort Pada Sistem Pendukung Keputusan Pemilihan Tempat Kost Berbasis Ios (Iphone Operating System)’, *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 6(2), pp. 318–324. Available at: <https://doi.org/10.29100/jupi.v6i2.2015>.

Roma Rio, S., Yusman, M. and Febi Eka, F. (2017) ‘Algoritma Bubble Sort Dan Selection Sort’, *Jurnal Komputasi*, 5(1), pp. 81–87. Available at: <https://jurnal.fmipa.unila.ac.id/komputasi/article/viewFile/1484/1228>.

Al Rivian, M.E. (2017) ‘Perbandingan Kecepatan Gabungan Algoritma Quick Sort dan Merge Sort dengan Insertion Sort, Bubble Sort dan Selection Sort’, *Jurnal Teknik Informatika dan Sistem Informasi*, 3(2), pp. 319–331. Available at: <https://doi.org/10.28932/jutisi.v3i2.629>.

Edwin, M. et al. (no date) ‘Perbandingan Algoritma Pengurutan Merge Sort , Quick Sort dan Heap Sort Dilihat dari Kompleksitasnya’, *Program Studi Teknik Informatika*, (10), pp. 1–6.

I. Gunawan, Sumarno, and H. S. Tambunan, “PENGUNAAN ALGORITMA SORTING BUBBLE SORT UNTUK PENENTUAN NILAI PRESTASI SISWA,” *J. SISTEMASI*, vol. 8, no. 2, pp. 296-304, 2019.

R. Sanjaya, and Oktafianto, “SISTEM PENDUKUNG KEPUTUSAN PEMILIHAN TEMPAT KOST DI PRINGSEWUMENGGUNAKAN METODE SIMPLE ADDITIVE WEIGHTING (SAW),” *KMSI*, vol. 5, no. 1, pp. 458-464, 2017.

D. Yusma, N. Merlina, and Nurajijah, “SISTEM INFORMASI PENCARIAN RUMAH KOST BERBASIS WEB,” *INTI NUSA MANDIRI*, vol. 15, no. 2, pp. 9-16, 2021. Edwin, M. et al. (no date) ‘Perbandingan Algoritma Pengurutan Merge Sort , Quick Sort dan Heap Sort Dilihat dari Kompleksitasnya’, *Program Studi Teknik Informatika*, (10), pp. 1–6.

F. E. Saputro, and F. N. Khasanah, “Teknik Selection Sort dan Bubble Sort Menggunakan Borland C++,” *J. Mahasiswa Bina Insani*, vol. 2, no. 2, pp. 136-145, 2018.

Munir, Rinaldi. (2008). *Diktat Kuliah IF2091 Struktur Diskrit*, Departemen Teknik Informatika, Institut Teknologi Bandung.

Yahya, Sofyansyah Yusari. 2014. Analisa Perbandingan Algoritma Bubble Sort dan Selection Sort Dengan Metode Perbandingan Eksponensial. Jurnal Pelita Informatika Budi Darma, Vol : VI, No : 3, April 2014.

Tjaru, Setia Negara B. 2010. Kompleksitas Algoritma Pengurutan Selection Sort dan Insertion Sort.

Oladipupo Esau Taiwo, Abikoye Oluwakemi Christianah, Akande Noah Oluwatobi, Kayode Anthonia Aderonke, Adeniyi Jide kehinde, Comparative Study Of Two Divide And Conquer Sorting Algorithms: Quicksort And Mergesort, Procedia Computer Science, Volume 171, 2020, Pages 2532-2540,-ISSN-1877-0509, <https://doi.org/10.1016/j.procs.2020.04.274>.

Al Rivan, Muhammad Ezar. "Perbandingan Kecepatan Gabungan Algoritma Quick Sort Dan Merge Sort Dengan Insertion Sort, Bubble Sort Dan Selection Sort." Jurnal Teknik Informatika dan Sistem Informasi, vol. 3, no. 2, 2017, doi: 10.28932/jutisi.v3i2.629.

From providing article:

Latifah, F. and Hidayatulloh, S. (2014) 'Tryout Psikotest Dengan Algoritma Merge Sort Untuk Peningkatan Scoring Berbasis Android', Jurnal Techno Nusa Mandiri, 11(2), pp. 137–148.g.

From the website:

Selection sort – Wikipedia Waktu Akses : 21 Oktober 2023, Pukul 10.00

Bubble sort – Wikipedia Waktu Akses : 21 Oktober 2023, Pukul 11.00