Studi Komparatif Metode Sorting dalam Python

# Abstrak

Sorting adalah salah satu operasi paling dasar dalam ilmu komputer. Berbagai algoritma sorting dikembangkan untuk meningkatkan efisiensi dalam pengurutan data. Dalam paper ini dibahas 9 metode sorting: Bubble, Exchange, Selection, Insertion, Merge, Quick, Heap, Shell, dan Counting Sort. Disediakan implementasi Python beserta analisis kompleksitas waktu dan keefisienan masing-masing.

# 1. Pendahuluan

Sorting adalah proses pengurutan elemen dalam urutan tertentu. Pemilihan algoritma sorting tergantung pada ukuran data, kondisi data, serta efisiensi waktu dan memori yang dibutuhkan.

# 2. Daftar Metode Sorting

* 1. Bubble Sort
* 2. Exchange Sort
* 3. Selection Sort
* 4. Insertion Sort
* 5. Merge Sort
* 6. Quick Sort
* 7. Heap Sort
* 8. Shell Sort
* 9. Counting Sort

# 3. Implementasi dan Analisis

Berikut adalah implementasi Python dari setiap metode sorting beserta kompleksitas waktunya.

## Tabel 1. Perbandingan Kompleksitas Algoritma Sorting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Metode | Terbaik | Rata-rata | Terburuk | Stabil |
| Bubble Sort | O(n) | O(n²) | O(n²) | Ya |
| Exchange Sort | O(n²) | O(n²) | O(n²) | Tidak |
| Selection Sort | O(n²) | O(n²) | O(n²) | Tidak |
| Insertion Sort | O(n) | O(n²) | O(n²) | Ya |
| Merge Sort | O(n log n) | O(n log n) | O(n log n) | Ya |
| Quick Sort | O(n log n) | O(n log n) | O(n²) | Tidak |
| Heap Sort | O(n log n) | O(n log n) | O(n log n) | Tidak |
| Shell Sort | O(n log n) | O(n log² n) | O(n²) | Tidak |
| Counting Sort | O(n+k) | O(n+k) | O(n+k) | Ya |

# 4. Kesimpulan

Setiap algoritma memiliki karakteristik berbeda:  
- Gunakan insertion jika data hampir terurut.  
- Gunakan merge atau heap untuk stabilitas.  
- Gunakan quick sort untuk performa tinggi.  
- Gunakan counting sort jika data integer dalam rentang kecil.  
  
Pemilihan algoritma harus mempertimbangkan kondisi dan skala data.