



## **IMPLEMENTASI ALGORITMA QUICKSORT PADA SISTEM AKADEMIK**

**Agung Wijoyo, Joseph Marsiano, Muhammad Ikhwan, Risky Rianto**

<sup>2</sup>Fakultas, Program Studi, Nama Institusi, Kota, Indonesia  
Email: [risky.unpam@gmail.com](mailto:risky.unpam@gmail.com)

**Abstrak**—Abstrak Algoritma quicksort merupakan salah satu algoritma pengurutan tercepat dan efisien yang biasa digunakan pada komputer. Dalam konteks sistem informasi akademik, pemilahan data siswa berdasarkan berbagai kriteria seperti nilai, nama atau nomor siswa sangatlah penting untuk keperluan administrasi dan akademik. Jurnal ini membahas tentang implementasi algoritma quicksort pada bahasa pemrograman C++ pada sistem informasi akademik. Hasil penerapan menunjukkan bahwa algoritma Quick Sort dapat mengurutkan data siswa dengan cepat dan efisien sehingga meningkatkan kinerja dan fungsionalitas seluruh sistem informasi akademik. Evaluasi dilakukan dengan menguji kecepatan dan efisiensi algoritma dengan dataset siswa yang besar, dan hasilnya menunjukkan bahwa Quick Sort merupakan pilihan yang tepat untuk kebutuhan penyortiran data sistem informasi akademik.

**Kata Kunci:** Algoritma Quick Sort; Komputer; Sistem Informasi; Akademik; Siswa

**Abstract**— *The quicksort algorithm is one of the fastest and most efficient sorting algorithms commonly used in computers. In the context of an academic information system, sorting student data based on various criteria such as grades, names, or student numbers is crucial for administrative and academic purposes. This journal discusses the implementation of the quicksort algorithm in the C++ programming language within an academic information system. The results of the implementation show that the Quick Sort algorithm can sort student data quickly and efficiently, thereby improving the performance and functionality of the entire academic information system. Evaluation was conducted by testing the speed and efficiency of the algorithm with a large student dataset, and the results indicate that Quick Sort is the right choice for sorting data in academic information systems.*

**Keyword:** *The Quick Sprt Algorithm; Computer; Information System; Academic; Student*

### **1. PENDAHULUAN**

Penyortiran data adalah salah satu fungsi utama pemrosesan data dalam sistem informasi. Dalam sistem informasi akademik, data mining sangat penting untuk berbagai tujuan, antara lain: Misalnya membuat daftar siswa berdasarkan nilainya, mencari informasi siswa, membuat laporan akademik, dan masih banyak lagi.

Algoritma quicksort dianggap sebagai salah satu algoritma pengurutan tercepat dan paling efisien dengan kompleksitas waktu rata-rata  $O(n \log n)$ . Quicksort membagi array menjadi dua subset berdasarkan pivot, sebuah elemen yang dipilih untuk memfasilitasi pemisahan. Elemen yang lebih kecil dari poros ditempatkan di sebelah kiri poros, dan elemen yang lebih besar dari poros ditempatkan di sebelah kanan poros. Proses ini diulangi secara rekursif pada kedua subchip hingga seluruh array diurutkan.

Kecepatan dan efisiensi Penyortiran Cepat membuatnya ideal untuk digunakan dalam sistem informasi akademik, yang sering kali memerlukan penyortiran data yang cepat dan akurat. Penerapan penyortiran cepat pada sistem informasi akademik bertujuan untuk meningkatkan efisiensi sistem, mengurangi waktu pemrosesan, dan mempercepat hasil pada saat pemilahan data siswa. Dengan penerapan algoritma ini diharapkan sistem informasi akademik mampu melakukan proses pemilahan data yang kompleks dengan lebih optimal.

### **2. METODE PENELITIAN**

Penelitian ini menggunakan metode eksperimen dengan mengimplementasikan algoritma Quick Sort pada sebuah sistem informasi akademik. Langkah-langkah implementasi meliputi:



1. Pemilihan Data: Data mahasiswa yang akan diurutkan diambil dari database sistem informasi akademik.
2. Implementasi Algoritma: Algoritma Quick Sort diimplementasikan dalam bahasa pemrograman yang digunakan oleh sistem informasi akademik.
3. Pengujian dan Evaluasi: Hasil pengurutan data dengan menggunakan Quick Sort diuji dan dievaluasi berdasarkan kecepatan dan efisiensinya.

### 3. ANALISA DAN PEMBAHASAN

Pada bagian ini berisi hasil dari kegiatan penelitian yang sudah dilakukan

#### 3.1 Implementasi Algoritma Quick Sort

(Berikut adalah implementasi algoritma Quick Sort dalam bahasa C++:

```
#include <iostream>
#include <vector>
#include <string>
#include <utility>
Using namespace std;
Void swap(pair<string, int>& a, pair<string, int>& b) {
    Pair<string, int> temp = a;
    A = b;
    B = temp;
}
Int partition(vector<pair<string, int>>& arr, int low, int high) {
    Int pivot = arr[high].second; // Memilih elemen terakhir sebagai pivot
    Int i = (low - 1); // Indeks elemen yang lebih kecil
    For (int j = low; j <= high - 1; j++) {
        If (arr[j].second < pivot) {
            I++;
            Swap(arr[i], arr[j]);
        }
    }
    Swap(arr[i + 1], arr[high]);
    Return (i + 1);
}
Void quickSort(vector<pair<string, int>>& arr, int low, int high) {
    If (low < high) {
        Int pi = partition(arr, low, high);

        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}
Void printArray(vector<pair<string, int>>& arr) {
    For (int i = 0; i < arr.size(); i++) {
        Cout << arr[i].first << " " << arr[i].second << endl;
    }
}
Int main() {
    Vector<pair<string, int>> data_mahasiswa = {
        {"Josep", 85}, {"Risky", 90}, {"Ikhwan", 80}, {"Risky", 95}
    };
    Cout << "Data sebelum diurutkan:" << endl;
    printArray(data_mahasiswa);
    quickSort(data_mahasiswa, 0, data_mahasiswa.size() - 1);
    Cout << "Data setelah diurutkan:" << endl;
```



```
printArray(data_mahasiswa);  
Return 0;  
}
```

#### Penjelasan Implementasi

1. Fungsi `swap`: Fungsi ini digunakan untuk menukar dua elemen dalam array.
2. Fungsi `partition`: Fungsi ini digunakan untuk membagi array menjadi dua sub-array berdasarkan pivot. Elemen-elemen yang lebih kecil dari pivot ditempatkan di sebelah kiri, dan elemen-elemen yang lebih besar ditempatkan di sebelah kanan.
3. Fungsi `quickSort`: Fungsi utama untuk mengurutkan array menggunakan algoritma Quick Sort. Fungsi ini memanggil fungsi `partition` dan melakukan pengurutan secara rekursif pada sub-array kiri dan kanan.
4. Fungsi `printArray`: Fungsi ini digunakan untuk mencetak isi array.
5. Fungsi `main`: Fungsi utama yang menginisialisasi data mahasiswa, mencetak data sebelum dan sesudah pengurutan, serta memanggil fungsi `quickSort`.

#### 3.1.1 Hasil Pembahasan

Implementasi ini menunjukkan bagaimana algoritma Quick Sort dapat diterapkan untuk mengurutkan data mahasiswa berdasarkan nilai. Algoritma ini dapat disesuaikan untuk mengurutkan data berdasarkan kriteria lain seperti nama atau nomor induk mahasiswa dengan melakukan sedikit modifikasi pada fungsi `partition`. Setelah implementasi dan pengujian, kami menemukan bahwa algoritma quick sort dapat mengurutkan data siswa dengan cepat dan efisien. Data dalam jumlah besar pun dapat diurutkan.

## 4 KESIMPULAN

Implementasi algoritma Quick Sort pada sistem informasi akademik terbukti efektif dan efisien dalam mengurutkan data mahasiswa. Algoritma ini dapat digunakan untuk berbagai keperluan pengurutan data dalam sistem informasi akademik, sehingga dapat meningkatkan kinerja dan fungsionalitas sistem.

## REFERENCES

- Hoare, C. A. R. (1962). Quicksort. The Computer Journal, 5(1), 10-16.  
Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms. MIT Press.