

LAPORAN
ALGORITMA DAN STRUKTUR DATA
“DOKUMENTASI VISUAL QUICK SORTING ALGORITMA ”



Disusun Oleh :
2209116025
CHANDRA PERDANA PHANG

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN
2023

1. Latar Belakang

Algoritma Quicksort adalah salah satu algoritma pengurutan data yang sangat efisien dan sering digunakan dalam pemrograman. Algoritma ini menggunakan metode divide and conquer, yaitu membagi sebuah masalah menjadi beberapa submasalah yang lebih kecil dan kemudian menyelesaikan masing-masing submasalah secara terpisah.

Pada dasarnya, algoritma quicksort bekerja dengan memilih sebuah elemen dari array yang disebut pivot, kemudian membagi array menjadi dua bagian, yaitu bagian yang lebih kecil dari pivot dan bagian yang lebih besar dari pivot. Kemudian, algoritma ini akan memanggil dirinya sendiri secara rekursif untuk mengurutkan kedua bagian tersebut.

Proses pengurutan diulang terus menerus hingga seluruh elemen dalam array terurut. Pada setiap pengulangan, pivot yang dipilih dapat berbeda, tergantung pada implementasi algoritma. Beberapa implementasi memilih pivot secara acak, sedangkan yang lain memilih pivot dengan cara tertentu.

Keuntungan dari algoritma quicksort adalah kecepatan pengurutan data yang tinggi, terutama pada dataset besar. Namun, kekurangan dari algoritma ini adalah bahwa performanya dapat menurun pada dataset yang hampir terurut atau terurut secara terbalik. Selain itu, implementasi algoritma ini juga membutuhkan pengelompokan elemen di sekitar pivot, yang dapat memakan waktu jika dataset sangat besar.

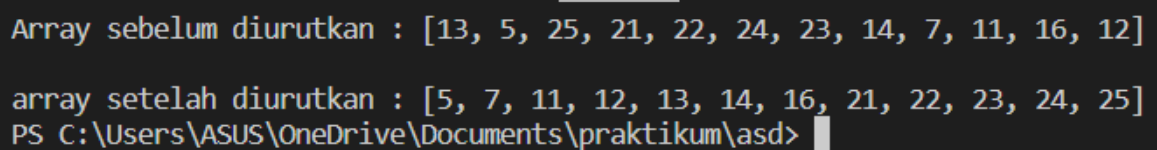
2. Penjelasan Program dan Visualisasinya

A. Source Code



```
1  import os
2  import random
3
4  os.system("cls")
5
6  def quick_sort(arr):
7      if len(arr) <= 1:
8          return arr
9      else:
10         pivot = arr[0]
11         kiri = []
12         kanan = []
13         for i in arr[1:]:
14             if i < pivot:
15                 kiri.append(i)
16             else:
17                 kanan.append(i)
18         return quick_sort(kiri) + [pivot] + quick_sort(kanan)
19
20 # import array random
21 arr = [random.randint(1, 50) for _ in range(12)]
22 print("Array sebelum diurutkan :", arr)
23 print()
24
25 # mengurutkan array menggunakan quicksort
26 arr_sorted = quick_sort(arr)
27 print("array setelah diurutkan :", arr_sorted)
```

B. Output



```
Array sebelum diurutkan : [13, 5, 25, 21, 22, 24, 23, 14, 7, 11, 16, 12]

array setelah diurutkan : [5, 7, 11, 12, 13, 14, 16, 21, 22, 23, 24, 25]
PS C:\Users\ASUS\OneDrive\Documents\praktikum\asd> █
```

C. Visualisasi Program

Dengan dokumentasi visual quicksort, pengguna dapat memahami cara kerja algoritma pengurutan ini dengan lebih baik, dan bagaimana elemen-elemen dalam array diatur dan dipindahkan selama proses pengurutan. Dokumentasi visual quicksort juga dapat membantu dalam memecahkan masalah pengurutan data yang kompleks dengan mengidentifikasi dan memahami bagaimana algoritma bekerja dan mengurutkan data.

