

Selection Sort

- Langkah pertama adalah mencari nilai terkecil (ascending) dari array, kemudian menukar nilai terkecil dengan nilai elemen pertama dari array yaitu indeks ke-0.
- Pada iterasi ke-2, dimulai dengan mencari nilai terkecil dari sisa elemen yang belum urut yaitu elemen indeks ke-1 sampai indeks ke-n, kemudian menukar nya dengan nilai indeks ke-1.
- Pada iterasi ke-3, dimulai lagi dengan mencari nilai terkecil dari elemen indeks ke-2 sampai indeks ke-n, kemudian menukarnya dengan nilai indeks ke-2.
- Begitu seterusnya sampai data terurut.

SelectionSort(arr):

```
n = panjang(arr)          # Menentukan panjang array.
untuk i dari 0 sampai n-1: # Ulangi setiap elemen array kecuali pada elemen terakhir.
    indeks_terkecil = i     # Menyimpan indeks untuk elemen terkecil yang ditemukan.
    untuk j dari i+1 sampai n: # Mencari elemen terkecil pada sisa array yang belum
        diurutkan.
        jika arr[j] < arr[indeks_terkecil]: # Jika pada elemen saat ini masih lebih kecil
            dari elemen terkecil yang di temukan sebelumnya.
            indeks_terkecil = j             # Memperbarui indeks elemen terkecil.
    tukar(arr[indeks_terkecil], arr[i])     # Menukar elemen terkecil dengan elemen
    pertama dari array yang belum diurutkan.
kembalikan arr                          # Mengembalikan array yang sudah diurutkan.
```

Quick Sort

Langkah pertama adalah memilih pivot. Bandingkan nilai pada pivot dengan data lainnya. Kemudian mempartisi yaitu membagi kumpulan data menjadi 2 bagian, satu bagian merupakan nilai yang lebih kecil dari pivot dan bagian yang lain berisi nilai yang lebih besar dari pivot. Kemudian dilakukan hal yang sama pada masing-masing bagian. Hal tersebut dilakukan dengan menerapkan konsep rekursi.

QuickSort(arr, low, high):

jika low < high maka # Jika masih ditemukan elemen yang belum di urutkan.

 pivotIndex = Partition(arr, low, high) # Memanggil fungsi Partition untuk mendapatkan pivot indeks.

 QuickSort(arr, low, pivotIndex-1) # Memanggil rekursif QuickSort untuk membagi dan mengurutkan bagian kiri pivot.

 QuickSort(arr, pivotIndex+1, high) # Memanggil rekursif QuickSort untuk membagi dan mengurutkan bagian kanan pivot.

Partition(arr, low, high):

 pivot = arr[high] # Memilih elemen pivot, yaitu elemen terakhir dalam array.

 i = low - 1 # Pemasangan indeks i yang akan digunakan untuk membagi bagian kiri dan kanan.

 untuk j = low sampai high-1 lakukan # Pengulangan dari elemen pertama sampai elemen sebelum pivot.

 jika arr[j] < pivot maka # Jika pada elemen saat ini lebih kecil dari pivot.

 i = i + 1 # Menaikkan indeks i.

 tukar(arr[i], arr[j]) # Menukarkan elemen arr[i] dengan elemen arr[j].

 tukar(arr[i+1], arr[high]) # Menukarkan elemen pivot dengan elemen arr[i+1].

 kembalikan i+1 # Mengembalikan pivot indeks.