

LAPORAN
PRAKTIKUM INFORMASI DAN STRUKTUR DATA

SEMESTER GANJIL 2022/2023

MODUL 12

Sorting



NAMA : ALEXIS DIVASONDA SIGAT NGAING

NPM : 210711407

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ATMA JAYA YOGYAKARTA
TAHUN 2022

Pertama adalah Insertion Sort. Insertion Sort adalah jenis sorting dengan cara memasukkan atau menyisipkan setiap elemen secara berulang. Metode ini mirip seperti mengurutkan kartu, dimana jika salah satu kartu dipindah posisinya maka kartu yang lain akan bergeser sesuai kondisi pemindahan kartu tersebut.

Berikut prosesnya dan contoh fungsinya.

2	3	9	6	4	5	1
2	3	9	6	4	5	1
2	3	9	6	4	5	1
2	3	6	9	4	5	1
2	3	4	6	9	5	1
2	3	4	5	6	9	1
1	2	3	4	5	6	9

```
void insertionSort(int *Data, int n) {
    int temp, j;
    for (int i = 1; i <= n-1; i++) { //n-1 karena index
        dimulai dr 0.
        temp = Data[i];
        j = i-1;

        //Data[j] > temp berfungsi untuk mengurutkan data secara
        ascending.
        //Data[j] < temp berfungsi untuk mengurutkan data secara
        descending.

        while (j>=0 && Data[j] > temp) {
            Data[j+1] = Data[j];
            j--;
        }
        Data[j+1] = temp;
    }
}
```

Pada fungsi insertSort akan melakukan perulangan dan eksekusi program insertionSort.

Kedua adalah Selection Sort, yang menggunakan perbandingan data. Data 1 dengan data lainnya akan dibandingkan dan data terkecil yang ditemukan akan menjadi acuannya. Selection Sort dapat dikatakan juga sebagai kombinasi antara sorting dan searching.

Berikut proses pengurutan dan contoh fungsinya.

7	5	1	9	2	6	4
1	5	7	9	2	6	4
1	2	7	9	5	6	4

1	2	4	9	5	6	7
1	2	4	5	9	6	7
1	2	4	5	6	9	7
1	2	4	5	6	7	9
1	2	4	5	6	7	9

```

void SelectionSort (int Data[ ], int n) {

    // variabel sementara untuk menyimpan posisi
    data minimum.

    int min;

    // mengurangi ukuran efektif array sebanyak
    satu di setiap iterasi.

    for (int i = 0; i < n-1 ; i++) {

        // anggap saja data pertama menjadi minimum
        dari array yang tidak disortir.

        min = i;

        // memberikan ukuran efektif dari array yang
        tidak disortir.

        for (int j = i+1; j < n ; j++ ) {
            if (Data[ j ] < Data[ min ]) {

                //mencari data minimum

                min = j ;
            }
        }
        // meletakkan data minimum pada posisi yang
        Tepat

        swap (&Data[ min ], &Data[ i ]) ;
    }
}

```

Ketiga adalah Counting Sort. Counting Sort mengurutkan dengan menentukan posisi elemen suatu nilai. Pada prosesnya diperlukan suatu rentang nilai yang sudah diketahui dan pada prosesnya menentukan jumlah nilai yang lebih kecil dari elemen lain agar posisi nilai tersebut dapat ditentukan. Algoritma ini dapat mengurutkan data dari besar ke kecil (Ascending) dan kecil ke besar (Descending). Algoritma ini tidak cocok untuk set data dengan jumlah besar karena kompleksitas dari algoritma ini adalah $O(n+r)$ di mana n adalah jumlah item dan r adalah jarak dari inputan.

Berikut contoh fungsinya.

```
void countingSort(int array[], int size) {
    int output[10];

    int max = array[0];
    for (int i = 1; i < size; i++) {
        if (array[i] > max)
            max = array[i];
    }

    int count[10];

    // Inisialisasi count array.
    for (int i = 0; i <= max; ++i) {
        count[i] = 0;
    }

    for (int i = 0; i < size; i++) {
        count[array[i]]++;
    }

    for (int i = 1; i <= max; i++) {
        count[i] += count[i - 1];
    }

    for (int i = size - 1; i >= 0; i--) {
        output[count[array[i]] - 1] = array[i];
        count[array[i]]--;
    }

    for (int i = 0; i < size; i++) {
        array[i] = output[i];
    }
}
```