

## MODUL VII

### SORTING

#### I. TUJUAN

1. Memahami dan menerapkan beberapa algoritma sorting dalam menyelesaikan berbagai studi kasus

#### II. DASAR TEORI

##### 1. Konsep Dasar Sorting

- Algoritma Sorting adalah algoritma untuk meletakkan kumpulan elemen data ke dalam urutan tertentu, berdasarkan satu atau beberapa kunci ke dalam tiap-tiap elemen
- Berdasarkan data terurutnya, algoritma sorting dibagi menjadi dua jenis, yaitu:
  - Ascending; pengurutan dari terkecil hingga terbesar. Contoh: a, b, c, d, e.
  - Descending; pengurutan dari nilai terbesar hingga terkecil. Contoh: e, d, c, b, a

##### 2. Insertion Sort

###### Konsep dasar Algoritma Insertion Sort

- Cara mengurutkannya adalah dicek satu persatu mulai dari yang kedua sampai dengan yang terakhir.
- Apabila ditemukan data yang lebih kecil dari data sebelumnya, maka data tersebut disisipkan pada posisi yang sesuai.

###### Pseudocode Algoritma Insertion Sort:

```
for i = 1 to n-1
    set j = i
    set t = a[j]
    repeat while j > 0 and a[j-1] > t
        move a[j-1] to right
    end repeat
    set a[j] = t
end for
```

###### Prosedur Insertion Sort dalam bahasa C++:

```
void insertion_sort(int arr[], int length) {
    int i, j, tmp;

    for (i = 1; i < length; i++) {
        j = i;

        while (j > 0 && arr[j - 1] > arr[j]) {
            tmp = arr[j];
            arr[j] = arr[j - 1];
            arr[j - 1] = tmp;
            j--;
        } //end of while loop
    } //end of for loop
}
```

**Contoh**

Diketahui array suatu integer terdiri dari 6 elemen sebagai berikut; {22, 10, 15, 3, 8, 2}. Contoh Program untuk mengurutkan ke-6 elemen tersebut adalah sebagai berikut:

```
#include <iostream>

using namespace std;

void insertion_sort(int arr[], int length) {
    int i, j, tmp;

    for (i = 1; i < length; i++) {
        j = i;

        while (j > 0 && arr[j - 1] > arr[j]) {
            tmp = arr[j];
            arr[j] = arr[j - 1];
            arr[j - 1] = tmp;
            j--;
        } //end of while loop
    } //end of for loop
}

void print_array(int a[], int length) {
    for(int i=0; i<length; i++) {
        cout << a[i] << "\t";
    }
    cout << endl;
}

int main() {
    int length = 6;
    int a[length] = {22, 10, 15, 3, 8, 2};

    insertion_sort(a, length);
    print_array(a, length);
}
```

**3. Bubble Sort**

**Konsep dasar Algoritma Bubble Sort**

- Cara mengurutkannya adalah membandingkan elemen yang sekarang dengan elemen yang berikutnya.
- Jika elemen sekarang > elemen berikutnya, maka tukar

**Contoh Prosedur Bubble Sort dalam bahasa C++**

```
void bubble_sort(int arr[], int length){
    bool not_sorted = true;
    int j=0,tmp;

    while (not_sorted){
        not_sorted = false;
        j++;
        for (int i = 0; i < length - j; i++){
            if (arr[i] > arr[i + 1]) {
```

```

        tmp = arr[i];
        arr[i] = arr[i + 1];
        arr[i + 1] = tmp;
        not_sorted = true;
    } //end of if
} //end of for loop
} //end of while loop
} //end of bubble_sort

```

#### 4. Selection Sort

##### Konsep dasar Algoritma Selection Sort:

- Cara mengurutkannya adalah dengan membandingkan elemen sekarang dengan elemen yang berikutnya sampai terakhir.
- Jika ditemukan elemen paling kecil, kemudian ditukar dengan elemen sekarang.

##### Contoh prosedur Selection Sort:

```

void selectSort(int arr[], int n) {
    int pos_min, temp;
    for (int i=0; i < n-1; i++) {
        pos_min = i;
        for (int j=i+1; j < n; j++) {
            if (arr[j] < arr[pos_min])
                pos_min=j;
        }

        if (pos_min != i) {
            temp = arr[i];
            arr[i] = arr[pos_min];
            arr[pos_min] = temp;
        }
    }
}

```

### III. GUIDED

#### I. Mengurutkan secara *ascending* untuk data numerik bertipe double menggunakan Algoritma Bubble Sort

```

#include <iostream>

using namespace std;

void bubble_sort(double arr[], int length){
    bool not_sorted = true;
    int j=0;
    double tmp;

    while (not_sorted){
        not_sorted = false;
        j++;
        for (int i = 0; i < length - j; i++){
            if (arr[i] > arr[i + 1]) {
                tmp = arr[i];

```

```
        arr[i] = arr[i + 1];
        arr[i + 1] = tmp;
        not_sorted = true;
    } //end of if
} //end of for loop
} //end of while loop
} //end of bubble_sort

void print_array(double a[], int length) {

    for(int i=0; i<length; i++) {
        cout << a[i] << "\t";
    }
    cout << endl;
}

int main() {

    int length = 5;
    double a[] = {22.1, 15.3, 8.2, 33.21, 99.99};

    cout << "Urutan bilangan sebelum sorting: " << endl;
    print_array(a, length);

    bubble_sort(a, length);

    cout << "\nUrutan bilangan setelah sorting: " << endl;
    print_array(a, length);
}
```

**2. Mengurutkan karakter secara descending (dari terbesar hingga terkecil) menggunakan Algoritma Insertion Sort**

```
#include <iostream>

using namespace std;

void insertion_sort(char arr[], int length) {
    int i, j;
    char tmp;

    for (i = 1; i < length; i++) {
        j = i;

        while (j > 0 && arr[j - 1] < arr[j]) {
            tmp = arr[j];
            arr[j] = arr[j - 1];
            arr[j - 1] = tmp;
            j--;
        } //end of while loop
    } //end of for loop
}

void print_array(char a[], int length) {

    for(int i=0; i<length; i++) {
        cout << a[i] << "\t";
    }
    cout << endl;
}
```

```

int main() {

    int length = 6;
    char a[length] = {'c', 'f', 'a', 'z', 'd', 'p'};

    cout << "Urutan karakter sebelum sorting: " << endl;
    print_array(a, length);

    insertion_sort(a, length);

    cout << "\nUrutan karakter setelah sorting: " << endl;
    print_array(a, length);

}

```

#### IV. TUGAS

1. Kelas S1 IF 2016 G memiliki 5 mahasiswa. Pada akhir semester mereka menerima lembar Indeks Prestasi Semester (IPS), masing-masing mahasiswa tersebut memiliki IPS sebagai berikut: {3.8, 2.9, 3.3, 4.0, 2.4}. Buatlah program untuk mengurutkan IPS mahasiswa tersebut dari yang terbesar hingga terkecil dengan menggunakan algoritma Selection Sort! (Score: 30)
2. Pak RT memiliki 10 warga dengan nama: siti, situ, sana, ana, ani, caca, cici, dida, dodo, dan dadi. Supaya mudah dalam melakukan pencarian, Pak RT akan mengurutkan nama-nama tersebut sesuai dengan alfabet. Buatlah program untuk membantu Pak RT dengan menggunakan algoritma Bubble Sort! (Score: 30)
3. Buatlah program yang meminta user menginputkan suatu bilangan **n** dan meminta user untuk menginputkan sejumlah **n** karakter. Kemudian program akan melakukan sorting secara menaik (ascending) dan menurun (descending)! (Score: 40)

```

input <n> = 5
Karakter ke-1: a
Karakter ke-2: c
Karakter ke-3: b
Karakter ke-4: e
Karakter ke-5: d
Urutan karakter sebelum sorting:
a      c      b      e      d

Urutan karakter setelah ascending sort:
a      b      c      d      e

Urutan karakter setelah descending sort:
e      d      c      b      a

```