JOBSHEET VI
INSERTION SORT DAN SHELL SORT SORT

6. 1 Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Mahasiswa mampu memahami algoritma *sorting* dengan menggunakan *Insertion* dan *Shell Sort*
2. Mahasiswa mampu membuat dan mendeklarasikan struktur algoritma *sorting* dengan menggunakan *Insertion* dan *Shell Sort*.
3. Mahasiswa mampu menerapkan dan mengimplementasikan algoritma *sorting* dengan menggunakan *Insertion* dan *Shell Sort*

6. 2 Insertion Sort

Waktu : 50 menit

6.2.1 Langkah-langkah Percobaan

1. Buat Project baru, dengan nama **InsertionSortTest**.
2. Tambahkan class **Insertion** pada project tersebut.
3. Tambahkan 2 atribut sebagai berikut:

```
public int [] data;
public int jumData;
```

4. Pada class **Insertion** buatlah konstruktor dengan parameter Data[] dan jumData

```
public Insertion(int Data[], int jmlData){
    jumData=jmlData;
    data=new int[jmlData];
    for (int i=0; i<jumData; i++){
        data[i]=Data[i];
    }
}
```

5. Tambahkan method **tampilData** pada class **Insertion**

```
void tampilData(){
    for (int i=0; i<jumData; i++){
        System.out.print(data[i]+"  ");
    }
    System.out.println("");
}
```

6. Implementasikan algoritma *insertion sort* pada method **insertion**

```
    for (int i=1; i<=data.length-1; i++){
    int temp=data[i];
    int j=i-1;
    while (j>=0 && data[j]>temp){
      data[j+1]=data[j];
      j--;
    }
    data[j+1]=temp;
}
}
```

7. Pada main di class **InsertionSortTest** buatlah array data bilangan bulat, seperti berikut:

```
int a[]={73, 67, 56, 32, 52, 41, 83, 37, 32, 10};
```

8. Deklarasi dan instansiasi object dari class **Insertion**

```
Insertion urut=new Insertion(a, a.length);
```

9. Cetak data awal dengan menggunakan method **tampilData**
10. Panggil proses pengurutan insertion sort dengan method **insertion**
11. Sebagai langkah terkahir, cetak data akhir untuk melihat data yang terurut dengan memanggil method **tampilData**.

6.2.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```
run:
Data sebelum urut
73  67  56  32  52  41  83  37  32  10
Data sesudah urut (ASC)
10  32  32  37  41  52  56  67  73  83
BUILD SUCCESSFUL (total time: 0 seconds)
```

6.2.3 Pertanyaan

1. Modifikasi percobaan di atas sehingga dapat menerima data secara dinamis dari inputan pengguna
2. Jelaskan maksud dari kondisi pada perulangan `while (j>=0 && data[j]>temp)`
3. Apakah tujuan dari perintah `data[j+1]= data[j];`
4. Sebutkan perintah untuk proses langkah menyisipkan (insert)

6.3 Shell Sort

Waktu : 50 menit

6.3.1 Langkah-langkah Percobaan

1. Buat Project baru, dengan nama **ShellSortTest**.
2. Tambahkan class **ShellSort** pada project tersebut.
3. Tambahkan 2 atribut sebagai berikut:

```
public int [] data;
public int jumData;
```

4. Pada class **ShellSort** buatlah konstruktor dengan parameter Data[] dan jumData

```
public ShellSort(int Data[], int jmlData){
    jumData=jmlData;
    data=new int[jmlData];
    for (int i=0; i<jumData; i++){
        data[i]=Data[i];
    }
}
```

5. Tambahkan method **tampilData** pada class **ShellSort**

```
void tampilData(){
    for (int i=0; i<jumData; i++){
        System.out.print(data[i]+"  ");
    }
    System.out.println("");
}
```

6. Implementasikan algoritma *shell sort sort* pada method **shellSort**

```
void shellSort(){
    int interval;
    for (interval= jumData/2; interval>0; interval/=2){
        for (int i=interval; i<jumData; i+=1){
            int temp=data[i];
            int j;
            for (j=i; j>=interval && data[j-interval]>temp; j-=interval){
                data[j]=data[j-interval];
            }
            data[j]=temp;
        }
    }
}
```

7. Pada main di class **ShellSortTest** buatlah array data bilangan bulat, seperti berikut:

```
int a[]={73, 67, 56, 32, 52, 41, 83, 37, 32, 10};
```

8. Deklarasi dan instansiasi object dari class **ShellSort**

```
ShellSort urut=new ShellSort(a, a.length);
```

9. Cetak data awal dengan menggunakan method **tampilData**
10. Panggil proses pengurutan insertion sort dengan method **shellSort**
11. Sebagai langkah terkahir, cetak data akhir untuk melihat data yang terurut dengan memanggil method **tampilData**.

6.3.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program anda dengan gambar berikut ini.

```
run:
Data sebelum urut
73  67  56  32  52  41  83  37  32  10
Data sesudah urut dengan Shell Sort (ASC)
10  32  32  37  41  52  56  67  73  83
BUILD SUCCESSFUL (total time: 0 seconds)
```

6.3.3 Pertanyaan

1. Modifikasi percobaan di atas sehingga dapat menerima data secara dinamis dari inputan pengguna
2. Jelaskan maksud dari kondisi pada perulangan

```
for (interval= jumData/2; interval>0; interval/=2)
```

3. Jelaskan potongan kode program berikut ini pada method shellSort

```
for (j=i; j>=interval && data[j-interval]>temp; j-=interval){
    data[j]=data[j-interval];
}
```

6.4 Tugas

Waktu : 100 menit

1. Pada Toko ATK terdapat data Barang sebagai berikut

Nama Barang	Stok	Harga
Pensil	35	1000
Buku	20	5000
Penggaris	50	1500
Bulpen	25	2000

Bantulah pemilik toko untuk mengurutkan barang tersebut berdasarkan stok yang paling banyak dengan menggunakan:

a) Insertion Sort

b) Shell Sort

2. Terdapat 3 nilai dalam proses seleksi yaitu nilai A, B, dan C seperti pada table berikut:

Nama	Nilai A	Nilai B	Nilai C	Total
Rizki	80	90	75	245
Indah	90	80	85	255
Dika	70	80	90	240
Hisam	50	90	90	230
Budi	100	100	90	290

a) Buatlah program untuk menginputkan nama dan 3 nilai mahasiswa (seperti pada table diatas), kemudian hitung total nilai A, B, dan C. Selanjutnya lakukan pengurutan data mahasiswa tersebut berdasarkan total nilai mahasiswa dari yang tertinggi ke rendah (*insertion/shell sort "pilih salah satu algoritma pengurutan"*) !

b) Tambahkan 1 method untuk menampilkan 3 mahasiswa yang mempunyai total nilai terbaik (tampilan harus terdiri dari nama, nilai A, nilai B, nilai C, dan total)!