

Laporan Praktikum ASD

Jobsheet 07

Nama: Gegas Anugrah Derajat

Kelas: SIB-1F

Percobaan 6.2.1

Membuat Class dan menambahkan method sequentialSearch()

```
public class Sorting11 {  
    public static void sequentialSearch(int[] arr, int key) {  
        for (int i = 0; i < arr.length; i++) {  
            if (arr[i] == key) {  
                System.out.println("Data ditemukan pada indeks ke-" + i);  
            }  
        }  
        System.out.println(x:"Data tidak ditemukan");  
    }  
}
```

Menambahkan fungsi main

```
public static void main(String[] args) {  
    int[] daftarNilai = {10, 5, 20, 15, 80, 45};  
    sequentialSearch(daftarNilai, key:5);  
}
```

Hasil

```
Data ditemukan pada indeks ke-1  
Data tidak ditemukan
```

Percobaan 6.2.2

Membuat class, menambahkan konstruktork dan membuat method tampil()

```
public class Mahasiswa11 {
    int nim;
    String nama;
    int umur;
    double ipk;

    Mahasiswa11(int ni, String n, int u, double i){
        nim = ni;
        nama = n;
        umur = u;
        ipk = i;
    }

    void tampil() {
        System.out.println("Nim = " + nim);
        System.out.println("Nama = " + nama);
        System.out.println("Umur = " + umur);
        System.out.println("IPK = " + ipk);
    }
}
```

Membuat class PencarianMhs dan menambahkan method tambah dan tampil

```
public class PencarianMhs11 {
    Mahasiswa11 listMhs[] = new Mahasiswa11[5];
    int idx;

    void tambah(Mahasiswa11 m) {
        if (idx < listMhs.length) {
            listMhs[idx] = m;
            idx++;
        } else {
            System.out.println(x: "Data sudah penuh!");
        }
    }

    void tampil() {
        for (Mahasiswa11 m : listMhs) {
            m.tampil();
            System.out.println(x: "-----");
        }
    }
}
```

Menambahkan method FindSeqSearch()

```
public int FindSeqSearch(int cari) {  
    int posisi = -1;  
    for (int j = 0; j < listMhs.length; j++) {  
        if (listMhs[j].nim == cari) {  
            posisi = j;  
            break;  
        }  
    }  
    return posisi;  
}
```

Membuat method Tampilposisi dan tampilData

```
public void Tampilposisi(int x, int pos) {  
    if (pos != -1) {  
        System.out.println("data : " + x + " ditemukan pada indeks " + pos);  
    } else {  
        System.out.println("data " + x + "tidak ditemukan");  
    }  
}  
  
public void tampilData(int x, int pos) {  
    if (pos != -1) {  
        System.out.println("Nim\t : " + x);  
        System.out.println("Nama\t : " + listMhs[pos].nama);  
        System.out.println("Umur\t : " + listMhs[pos].umur);  
        System.out.println("IPK\t : " + listMhs[pos].ipk);  
    } else {  
        System.out.println("data " + x + "tidak ditemukan");  
    }  
}
```

Membuat class MahasiswaMain, membuat objek PencarianMhs dan membuat 5 objek mahasiswa

```
import java.util.Scanner;

public class MahasiswaMain11 {
    Run | Debug
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        // Scanner s1 = new Scanner(System.in);

        PencarianMhs11 data = new PencarianMhs11();
        int jumMhs = 5;

        System.out.println(x:"-----");
        System.out.println(x:"Masukkan data mahasiswa secaraurut dari Nim Terkecil");
        for (int i = 0; i < jumMhs; i++) {
            System.out.println(x:"-----");
            System.out.print(s:"Nim\t: ");
            int nim = s.nextInt();
            System.out.print(s:"Nama\t: ");
            String nama = s.next();
            System.out.print(s:"Umur\t: ");
            int umur = s.nextInt();
            System.out.print(s:"IPK\t: ");
            double ipk = s.nextDouble();

            Mahasiswa11 m = new Mahasiswa11(nim, nama, umur, ipk);
            data.tambah(m);
        }
    }
}
```

Memanggil method tampil()

```
System.out.println(x:"-----");
System.out.println(x:"Data keseluruhan Mahasiswa : ");
data.tampil();
```

Membuat input pencarian. Memanggil method FindSeqSearch(), method Tampilposisi(), dan method tampilData()

```
System.out.println(x:"-----");
System.out.println(x:"-----");
System.out.println (x:"Pencarian Data : ");
System.out.println (x:"Masukkan Nim Mahasiswa yang dicari: ");
System.out.print (s:"NIM : ");
int cari = s.nextInt ();
System.out.println (x:"menggunakan sequential Search" );
int posisi = data. FindSeqSearch (cari);

data.Tampilposisi(cari, posisi);
data.tampilData(cari, posisi);
```

Hasil

```
-----  
Masukkan data mahasiswa secara urut dari Nim Terkecil  
-----  
Nim      : 2017  
Nama     : Dewilestari  
Umur     : 23  
IPK      : 3.5  
-----  
Nim      : 2018  
Nama     : SintaSanjaya  
Umur     : 22  
IPK      : 4  
-----  
Nim      : 2019  
Nama     : DanangAdi  
Umur     : 22  
IPK      : 3.7  
-----  
Nim      : 2020  
Nama     : BudiPrakasa  
Umur     : 20  
IPK      : 2.9  
-----  
Nim      : 2021  
Nama     : VaniaSiti  
Umur     : 20  
IPK      : 3.0  
-----
```

```
Data keseluruhan Mahasiswa :  
Nim = 2017  
Nama = Dewilestari  
Umur = 23  
IPK = 3.5  
-----  
Nim = 2018  
Nama = SintaSanjaya  
Umur = 22  
IPK = 4.0  
-----  
Nim = 2019  
Nama = DanangAdi  
Umur = 22  
IPK = 3.7  
-----  
Nim = 2020  
Nama = BudiPrakasa  
Umur = 20  
IPK = 2.9  
-----  
Nim = 2021  
Nama = VaniaSiti  
Umur = 20  
IPK = 3.0  
-----
```

```
-----  
Pencarian Data :  
Masukkan Nim Mahasiswa yang dicari:  
NIM : 2018  
menggunakan sequential Search  
data : 2018ditemukan pada indeks 1  
Nim      :2018  
Nama     :SintaSanjaya  
Umur     :22  
IPK      :4.0  
-----
```

6.2.3. Pertanyaan

1. Lakukan perubahan array daftarNilai pada fungsi main().

```
Run | Debug
public static void main(String[] args) {
    int[] daftarNilai = { 10, 5, 20, 15, 5, 45 };
    sequentialSearch(daftarNilai, 5);
}
```

2. Jelaskan perbedaan metod **TampilData** dan **Tampilposisi** pada class PencarianMhs
3. Jelaskan fungsi **break** pada kode program dibawah ini!

```
if (listMhs[j].nim==cari) {
    posisi = j;
    break;
}
```

4. Jika Data Nim yang dimasukkan tidak terurut dari kecil ke besar. Apakah program masih dapat berjalan? Apakah hasil yang dikeluarkan benar? Mengapa demikian!

Jawaban:

1.

```
int[] daftarNilai = {10, 5, 20, 15, 80, 45, 50, 55};
sequentialSearch(daftarNilai, key:5);
```

```
Data ditemukan pada indeks ke-1
Data tidak ditemukan
```

2. **TampilData** memiliki fungsi untuk menampilkan data yang ditemukan pada method search.
Tampilposisi memiliki fungsi untuk menampilkan posisi indeks data pada array.
3. Untuk menghentikan perulangan jika nilai telah ditemukan.
4. Program masih dapat berjalan, karena fungsi search akan mencari data dari awal array hingga akhir.

Percobaan 6.3.1

Menambahkan method `BinarySearchAsc()`

```
public static int binarySearchAsc(int[] arr, int key) {  
    int start = 0, end = arr.length - 1;  
  
    while (start <= end) {  
        int mid = start + (end - start) / 2;  
  
        if (arr[mid] == key) {  
            return mid;  
        }  
  
        if (arr[mid] < key) {  
            start = mid + 1;  
        } else {  
            end = mid - 1;  
        }  
    }  
  
    return -1;  
}
```

Menambahkan baris program untuk menguji method pada fungsi main

```
int[] sortedNilai = { 5, 5, 10, 20, 30, 40, 50 };  
int index = binarySearchAsc(sortedNilai, key:5);  
  
if (index != -1) {  
    System.out.println("Data ditemukan pada indeks ke-" + index);  
} else {  
    System.out.println("Data tidak ditemukan");  
}
```

Hasil

```
g:\code\user\workspace\storage\0932  
Data ditemukan pada indeks ke-5  
Data tidak ditemukan  
Data ditemukan pada indeks ke-1
```

Percobaan 6.3.2

Menambahkan method `FindBinarySearch()` pada class `PencarianMhs`

```
public int FindBinarySearch(int cari, int left, int right) {  
    int mid;  
    if (right >= left) {  
        mid = (left + right) / 2;  
        if (cari == listMhs[mid].nim) {  
            return mid;  
        } else if (listMhs[mid].nim > cari) {  
            return FindBinarySearch(cari, left, mid - 1);  
        } else {  
            return FindBinarySearch(cari, mid + 1, right);  
        }  
    }  
    return -1;  
}
```


Memanggil method di class MahasiswaMain

```
System.out.println(x:"=====");
System.out.println(x:"Menggunakan binary search");
posisi = data.FindBinarySearch(cari, left:0, jumMhs - 1);
data.Tampilposisi(cari, posisi);
data.tampilData(cari, posisi);
}
```

Hasil

```
-----
Masukkan data mahasiswa secara urut dari Nim Terkecil
-----
Nim      : 2017
Nama     : DewiLestari
Umur     : 23
IPK      : 3.5
-----
Nim      : 2018
Nama     : SintaSanjaya
Umur     : 22
IPK      : 4
-----
Nim      : 2019
Nama     : DanangAdi
Umur     : 22
IPK      : 3.7
-----
Nim      : 2020
Nama     : BudiPrakarsa
Umur     : 20
IPK      : 2.9
-----
Nim      : 2021
Nama     : VaniaSiti
Umur     : 20
IPK      : 3
-----
```

```
Data keseluruhan Mahasiswa :
Nim = 2017
Nama = DewiLestari
Umur = 23
IPK = 3.5
-----
Nim = 2018
Nama = SintaSanjaya
Umur = 22
IPK = 4.0
-----
Nim = 2019
Nama = DanangAdi
Umur = 22
IPK = 3.7
-----
Nim = 2020
Nama = BudiPrakarsa
Umur = 20
IPK = 2.9
-----
Nim = 2021
Nama = VaniaSiti
Umur = 20
IPK = 3.0
-----
```

```
-----
Pencarian Data :
Masukkan Nim Mahasiswa yang dicari:
NIM : 2018
menggunakan sequential search
data : 2018 ditemukan pada indeks 1
Nim      :2018
Nama     :SintaSanjaya
Umur     :22
IPK      :4.0
=====
Menggunakan binary search
data : 2018 ditemukan pada indeks 1
Nim      :2018
Nama     :SintaSanjaya
Umur     :22
IPK      :4.0
=====
```


6.3.3. Pertanyaan

1. Tunjukkan pada kode program yang mana proses divide dijalankan!
2. Tunjukkan pada kode program yang mana proses conquer dijalankan!
3. Jika data Nim yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian!
4. Jika Nim yang dimasukkan dari NIM terbesar ke terkecil (misal : 20215, 20214, 20212, 20211, 20210) dan elemen yang dicari adalah 20210. Bagaimana hasil dari binary search? Apakah sesuai? Jika tidak sesuai maka ubahlah kode program binary search agar hasilnya sesuai
5. Modifikasilah program diatas yang mana jumlah mahasiswa yang di inputkan sesuai dengan masukan dari keyboard.

Jawaban:

1.

```
if (right >= left) {  
    mid = (left + right) / 2;
```

2.

```
    mid = (left + right) / 2;  
    if (cari == listMhs[mid].nim) {  
        return (mid);  
    } else if (listMhs[mid].nim > cari) {  
        return FindBinarySearch(cari, left, mid - 1);  
    } else {  
        return FindBinarySearch(cari, mid + 1, right);  
    }  
}
```

3. Program masih dapat berjalan, karena fungsi search akan mencari data dari awal array hingga akhir.

4. Hasilnya tidak sesuai, maka kode harus diubah seperti berikut

```
    mid = (left + right) / 2;  
    if (listMhs[mid].nim < cari) {  
        return FindBinarySearch(cari, left, mid - 1);  
    } else if (listMhs[mid].nim > cari) {  
        return FindBinarySearch(cari, mid + 1, right);  
    } else {  
        return mid;  
    }  
}
```

5.

```
public class PencarianMhs11 {  
    Mahasiswa11 listMhs[];  
    int idx;  
    public PencarianMhs11(int jumMhs) {  
        listMhs = new Mahasiswa11[jumMhs];  
    }  
}
```

```
public static void main(String[] args) {  
    Scanner s = new Scanner(System.in);  
    int jumMhs;  
    // Scanner s1 = new Scanner(System.in);  
    System.out.println(x:"-----");  
    System.out.print(s:"Masukkan data yang ingin ditambahkan: ");  
    jumMhs = s.nextInt();  
  
    PencarianMhs11 data = new PencarianMhs11(jumMhs);  
}
```

Percobaan 6.4

Membuat class MergeSorting dan membuat method merge

```
public class MergeSorting11 {  
    public void mergeSort(int[] data) {  
        sort(data, left:0, data.length -1);  
    }  
}
```

Membuat method merge()

```
private void merge(int data[], int left, int middle, int right) {  
    int[] temp = new int[data.length];  
    for (int i = left; i <= right; i++) {  
        temp[i] = data[i];  
    }  
    int a = left;  
    int b = middle + 1;  
    int c = left;  
  
    while (a <= middle && b <= right) {  
        if (temp[a] <= temp[b]) {  
            data[c] = temp[a];  
            a++;  
        } else {  
            data[c] = temp[b];  
            b++;  
        }  
        c++;  
    }  
    int s = middle - a;  
    for (int i = 0; i <= s; i++) {  
        data[c + i] = temp[a + i];  
    }  
}
```

Membuat method sort() dan menambahkan method printArray()

```
private void sort(int data[], int left, int right) {  
    if (left < right) {  
        int middle = (left + right) / 2;  
        sort(data, left, middle);  
        sort(data, middle + 1, right);  
        merge(data, left, middle, right);  
    }  
}  
  
public void printArray(int arr[]) {  
    int n = arr.length;  
    for (int i = 0; i < n; i++) {  
        System.out.print(arr[i] + " ");  
    }  
    System.out.println();  
}
```

Hasil

```
sorting dengan merge sort  
data awal  
10 40 30 50 70 20 100 90  
setelah diurutkan  
10 20 30 40 50 70 90 100
```

6.5. Latihan Praktikum

1. Modifikasi percobaan searching diatas yang menggunakan Searching array of object dengan ketentuan berikut ini
 - Pencarian dilakukan berdasarkan Nama Mahasiswa (gunakan Algoritma binary Search)
 - Buat aturan untuk mendeteksi hasil pencarian lebih dari 1 hasil dalam bentuk kalimat peringatan!

Jawaban

Pencarian berdasarkan nama (menggunakan binary search)

```
System.out.println(x:"_____");
System.out.println(x:"_____");
System.out.println (x:"Pencarian Data : ");
System.out.println (x:"Masukkan Nama Mahasiswa yang dicari: ");
System.out.print (s:"Nama : ");
String cari = s.next();
System.out.println(x:"=====");
System.out.println(x:"Menggunakan binary search");
// posisi = data.FindBinarySearch(cari, 0, jumMhs - 1);
data.countSearchName(cari);
int posisi = data.FindBinarySearch(cari, left:0, jumMhs - 1, count:0 );
data.TampilposisiNama(cari, posisi);
data.TampilDataNama(cari, posisi);
```

```
public int FindBinarySearch(String cari, int left, int right, int count) {
    if (right >= left) {
        int mid = (left + right) / 2;
        int comparison = listMhs[mid].nama.compareTo(cari);

        if (comparison < 0) {
            return FindBinarySearch(cari, mid + 1, right, count);
        } else if (comparison > 0) {
            return FindBinarySearch(cari, mid - 1, left, count);
        } else {
            return mid;
        }
    }
    return -1;
}
```

```
public void TampilposisiNama(String x, int pos) {
    if (pos != -1) {
        System.out.println("data : " + x + " ditemukan pada indeks " + pos);
    } else {
        System.out.println("data " + x + " tidak ditemukan");
    }
}

public void TampilDataNama(String x, int pos) {
    if (pos != -1) {
        System.out.println("Nim\t : " + x);
        System.out.println("Nama\t : " + listMhs[pos].nama);
        System.out.println("Umur\t : " + listMhs[pos].umur);
        System.out.println("IPK\t : " + listMhs[pos].ipk);
    } else {
        System.out.println("data " + x + " tidak ditemukan");
    }
}
```

Mendeteksi hasil pencarian lebih dari satu hasil dalam kalimat peringatan

```
public void countSearchName(String cari) {  
    int count = 0;  
    for(int i=0; i<listMhs.length; i++){  
        if(cari.equalsIgnoreCase(listMhs[i].nama)){  
            count += 1;  
        }  
    }  
  
    if (count > 1) {  
        System.out.println(x:"Data yang ditemukan lebih dari 1");  
        return;  
    }  
}
```

```
// posisi = data.Findbinary.  
data.countSearchName(cari);
```