

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN STRUKTUR DATA  
JOBSHEET 7**



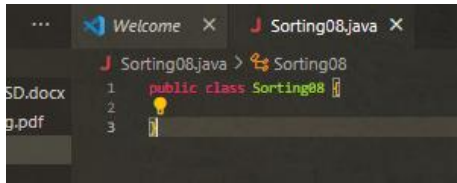
**Oleh:  
DZULFIKAR MUHAMMAD AL GHIFARI  
NIM. 2341760071  
SIB-1F / 08  
D-IV SISTEM INFORMASI BISNIS  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

## PRAKTIKUM 6

### 6.2 Searching / Pencarian Menggunakan Algoritma Sequential Search

#### 6.2.1 Sequential Search Menggunakan Array

1. Membuat class sorting



2. Menambahkan method sequentialSearch

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

3. Menambahkan method main

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

#### 6.2.2 Sequential Search Menggunakan Array of Object

1. Membuat class Mahasiswa dan deklarasi var

```
public class Mahasiswa08 {  
    int nim;  
    String nama;  
    int umur;  
    double ipk;
```

2. Membuat konstruktor

```
Mahasiswa08(int ni, String n, int u, double i)  
{  
    nim = ni;  
    nama = n;  
    umur = u;  
    ipk = i;  
}
```

3. Membuat method tampil

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

4. Membuat class pencarianMhs

```
public class PencarianMhs08 {  
    }  
}
```

5. Menambahkan method tambah

```
int idx;  
void tambah(Mahasiswa08 m) {  
    if (idx < listMhs.length) {  
        listMhs[idx] = m;  
        idx++;  
    } else {  
        System.out.println("Data sudah penuh !! ");  
    }  
}
```

6. Menambahkan method tampil

```
void tampil() {  
    for (Mahasiswa08 m : listMhs) {  
        m.tampil();  
        System.out.println("-----");  
    }  
}
```

7. 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;  
}
```

8. Menambahkan method tampil posisi

```
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");  
    }  
}
```

9. Menambahkan method tampil data

```
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");  
    }  
}
```

10. Membuat class mahasiswa main

```
public class MahasiswaMain08 {  
    Run | Debug  
    public static void main(String[] args) {  
    }  
}
```

## 11 . mengisi fungsi main

```
public static void main(String[] args) {
    Scanner s = new Scanner(System.in);
    Scanner s1 = new Scanner(System.in);

    PencarianMhs08 data = new PencarianMhs08();
    int jumMhs = 5;

    System.out.println(x: "-----");
    System.out.println(x: "Masukkan data mahasiswa secara Urut 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 = s1.nextLine();
        System.out.print(s: "Umur\t: ");
        int umur = s.nextInt();
        System.out.print(s: "IPK\t: ");
        double ipk = s.nextDouble();

        Mahasiswa08 m = new Mahasiswa08(nim, nama, umur, ipk);
        data.tambah(m);
    }
}
```

## VERIFIKASI HASIL PERCOBAAN 6.2.3

```
-----
Masukkan data mahasiswa secara Urut dari Nim Terkecil
-----
Nim      : 2017
Nama     : Dewi Lestari
Umur     : 23
IPK      : 3.5
-----
Nim      : 2018
Nama     : Sinta sanjaya
Umur     : 22
IPK      : 4
-----
Nim      : 2019
Nama     : Danang Adi
Umur     : 22
IPK      : 3.7
-----
Nim      : 2020
Nama     : Budi Prakarsa
Umur     : 20
IPK      : 2.9
-----
Nim      : 2021
Nama     : Vania Siti
Umur     : 20
IPK      : 3.0
-----
```

```

Data keseluruhan Mahasiswa :
Nim = 2017
Nama = Dewi Lestari
Umur = 23
IPK = 3.5
-----
Nim = 2018
Nama = Sinta sanjaya
Umur = 22
IPK = 4.0
-----
Nim = 2019
Nama = Danang Adi
Umur = 22
IPK = 3.7
-----
Nim = 2020
Nama = Budi Prakarsa
Umur = 20
IPK = 2.9
-----
Nim = 2021
Nama = Vania Siti
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     : Sinta sanjaya
Umur     : 22
IPK      : 4.0

```

### PERTANYAAN 3.2.3

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.

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

Data ditemukan pada indeks ke-5
Data tidak ditemukan
PS E:\1POLINEMA\2Genap 2023-2024\Pra
```

Data tidak ditemukan tampil karena pada method sequentialSearch terdapat print di bawah sendiri

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

2. TampilData memiliki fungsi untuk menampilkan data yang telah di temukan pada method search  
TampilPosisi memiliki fungsi untuk menampilkan detail data yang ditemukan yaitu posisi index data pada array
3. Untuk menghentikan perulangan jika kondisi terpenuhi
4. Program masih dapat berjalan, hasilnya juga benar, hal ini karena fungsi searching tidak akan mencari data dari awal array hingga akhir

```
-----
Pencarian Data :
Masukkan Nim Mahasiswa yang dicari:
NIM : 17
menggunakan sequential Search
data : 17 ditemukan pada indeks 2
Nim      : 17
Nama     : cek
Umur     : 9
IPK      : 2.3
PS E:\1POLINEMA\2Genap 2023-2024\PraktikumAlgoritma\pertemuan7> []
```

## 6.3 Searching / Pencarian Menggunakan Binary Search

### 6.3.1 Langkah-langkah Percobaan Binary Search menggunakan Array

1. Menambahkan method binarySearchAsc pada class Sorting

```
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;
}
```

2. Modifikasi kode untuk memanggil method selectionSort

```
System.out.println(x:"Data mahasiswa setelah sorting asc berdasarkan ipk");
list.selectionSort();
list.tampil();
```



### 3. Menambahkan baris main untuk binary search

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

    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");
    }
}
```

Data ditemukan pada indeks ke-1

## 6.3.2 Langkah-langkah Percobaan Binary Search menggunakan Array

### 1. 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;
}
```

### 2. Memanggil method FindBinarySearch

```
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);
```

## VERIFIKASI HASIL PERCOBAAN 6.3.3

```
-----
Masukkan data mahasiswa secara Urut dari Nim Terkecil
-----
Nim      : 1
Nama     : agus
Umur     : 23
IPK      : 3
-----
Nim      : 2
Nama     : budi
Umur     : 14
IPK      : 4
-----
Nim      : 3
Nama     : budi
Umur     : 51
IPK      : 2
-----
Nim      : 4
Nama     : bagong
Umur     : 20
IPK      : 4
-----
Nim      : 5
Nama     : setpi
Umur     : 20
IPK      : 3
-----
```

```

Data keseluruhan Mahasiswa :
Nim = 1
Nama = agus
Umur = 23
IPK = 3.0
-----
Nim = 2
Nama = budi
Umur = 14
IPK = 4.0
-----
Nim = 3
Nama = budi
Umur = 51
IPK = 2.0
-----
Nim = 4
Nama = bagong
Umur = 20
IPK = 4.0
-----
Nim = 5
Nama = setpi
Umur = 20
IPK = 3.0
-----

```

```

=====
Pencarian Data :
Masukkan Nim Mahasiswa yang dicari:
NIM : 5
menggunakan sequential Search
data : 5 ditemukan pada indeks 4
Nim      : 5
Nama      : setpi
Umur      : 20
IPK       : 3.0
=====
menggunakan binary Search
data : 5 ditemukan pada indeks 4
Nim      : 5
Nama      : setpi
Umur      : 20
IPK       : 3.0
=====

```

#### PERTANYAAN 6.3.4

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. 

```
(right >= left) {
    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;
    }
}
```
- 2.



3. Program bisa di run meskipun data tidak urut, karena searching akan mencari nilai dari data urutan awal hingga akhir

```
Data keseluruhan Mahasiswa :
Nim = 4
Nama = aagus
Umur = 45
IPK = 3.0
-----
Nim = 1
Nama = adib
Umur = 21
IPK = 4.0
-----
Nim = 2
Nama = gges
Umur = 32
IPK = 3.0
-----
Nim = 5
Nama = uwi
Umur = 2
IPK = 2.0
-----
Nim = 3
Nama = gerard
Umur = 14
IPK = 4.0
-----

Pencarian Data :
Masukkan Nim Mahasiswa yang dicari:
NIM : 4
menggunakan sequential Search
data : 4 ditemukan pada indeks 0
Nim      : 4
Nama     : aagus
Umur     : 45
IPK      : 3.0
```

4. Hasilnya tidak sesuai, dengan melakukan ubahan seperti dibawah ini , maka searching akan dapat dilakukan

```
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;
}

menggunakan binary Search
data : 8 ditemukan pada indeks 4
Nim      : 8
Nama     : ten
Umur     : 5
IPK      : 4.0
```

5. Dengan menambahkan inputan seperti di bawah ini

```
Scanner s1 = new Scanner(System.in);
int jumMhs;

PencarianMhs08 data = new PencarianMhs08();
System.out.println(x:"-----");
System.out.print(s:"Masukkan berapa banyak data yang ingin ditambahkan : ");
jumMhs = s.nextInt();

public class PencarianMhs08 {
    Mahasiswa08 listMhs[];
    int idx;
    public PencarianMhs08(int jumMhs) {
        listMhs = new Mahasiswa08[jumMhs];
    }
}
```

## 6.4 Percobaan Pengayaan Divide and Conquer

### 6.4.2 Langkah-langkah Percobaan Merge Sort

#### 1. Membuat class merge sort

```
public class MergeSorting08 {  
    public void MergeSorting(int[] data){  
  
    }  
}
```

#### 2. menambahkan method merge

```
public 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;  
  
    // membandingkan setiap bagian  
    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];  
    }  
}
```

#### 3. Menambahkan method sort

```
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);  
    }  
}
```

#### 4. Menambahkan method printArray

```
public void printArray (int arr []) {  
    int n = arr.length;  
    for (int i=0; i<n;i++){  
        System.out.print (arr[i]+" ");  
    }  
    System.out.println ();  
}
```

#### 5. Membuat class main

```
public class SortMain {  
    Run | Debug  
    public static void main(String[] args) {  
        int data[] = {10, 40, 30, 50, 70, 20, 100, 90};  
        System.out.println(x:"sorting dengan merge sort");  
        MergeSorting08 mSort= new MergeSorting08 ();  
        System.out.println(x:"data awal");  
        mSort.printArray (data);  
        mSort.mergeSort (data);  
        System.out.println(x:"setelah diurutkan");  
        mSort.printArray (data);  
    }  
}
```

### VERIFIKASI HASIL PERCOBAAN 6.4.3

```
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

Untuk pencarian berdasarkan nama dapat menyesuaikan kode seperti di bawah ini

```
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, jumbhs - 1);
int posisi = data.FindBinarySearchByName(cari, left:0, jumbhs - 1);
data.TampilposisiNama(cari, posisi);
data.TampilDataNama (cari, posisi);
}
```

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

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

untuk memberi peringatan bahwa data yang ditemukan lebih dari satu, maka dengan ditambahkan kode dibawah ini untuk mendeteksi

```
// posisi = data.FindBinarySearch  
data.CountSearchName(cari);  
int posisi = data.FindBinarySearch
```

```
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("Data yang ditemukan lebih dari 1");  
    }  
}
```

```
menggunakan binary Search  
Data yang ditemukan lebih dari 1
```