

LAPORAN PRAKTIKUM MINGGU KE-7

SEARCHING



Disusun oleh:

Daffa Aqila Rahmatullah

2041720098

D4 TEKNIK INFORMATIKA

TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2021

LAPORAN

A. KODE PROGRAM

6.2 Searching / Pencarian Menggunakan Algoritma Sequential Search

```
public class Mahasiswa {
    int nim;
    String nama;
    int umur;
    double ipk;
    Mahasiswa(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);
    }
}

public class PencarianMHS {
    Mahasiswa listMHs[] = new Mahasiswa[5];
    int idx;
    void tambah(Mahasiswa m){
        if(idx < listMHs.length)
        {
            listMHs[idx] = m;
            idx++;
        }
        else
        {
            System.out.println("Data sudah Penuh");
        }
    }
}
```

```

    }
    void tampil(){
        for (Mahasiswa m : listMHs) {
            m.Tampil();

System.out.println("=====");
        }
    }
    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;
    }
    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");
    }
}

public class MahasiswaMain {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        Scanner s1 = new Scanner(System.in);

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

        System.out.println("=====
=====");

        System.out.println("Masukkan data mahasiswa
secara urut dari Nim terkecil");
        for (int i = 0; i < jumMhs; i++)
        {
            System.out.println("-----");

```

```

        System.out.print("nim\t : ");
        int nim = s.nextInt();
        System.out.print("Nama\t: ");
        String nama = s1.nextLine();
        System.out.print("Umur\t: ");
        int umur = s.nextInt();
        System.out.print("IPK\t: ");
        double ipk = s.nextDouble();

        Mahasiswa m = new Mahasiswa(nim, nama, umur,
ipk);

        data.tambah(m);
    }
    System.out.println("-----
-----");
    System.out.println("Data Keseluruhan Mahasiswa :
");

    data.tampil();
    System.out.println("data mahasiswa setelah
sorting desc berdasarkan ipk");
    data.selectionSort();
    data.tampil();

System.out.println("_____");

System.out.println("_____");
    System.out.println("Pencarian data: ");
    System.out.println("Masukkan Nim Mahasiswa Yang
dicari: ");
    System.out.print("Nim: ");
    int cari = s.nextInt();
    System.out.println("Menggunakan Sequential
Search");
    int posisi = data.FindSeqSearch(cari);

```

```
data.TampilPosisi(cari, posisi);  
data.TampilData(cari, posisi);
```

6.3 Searching / Pencarian Menggunakan Binary Search

```
public class Mahasiswa {  
    int nim;  
    String nama;  
    int umur;  
    double ipk;  
    Mahasiswa(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);  
    }  
}  
  
public class PencarianMHS {  
    Mahasiswa listMHs[] = new Mahasiswa[5];  
    int idx;  
    void tambah(Mahasiswa m){  
        if(idx < listMHs.length)  
        {  
            listMHs[idx] = m;  
            idx++;  
        }  
        else  
        {  
            System.out.println("Data sudah Penuh");  
        }  
    }  
}
```

```

    }
    void tampil(){
        for (Mahasiswa m : listMHs) {
            m.Tampil();

System.out.println("=====");
        }
    }
    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;
    }
    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");
    }
}

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;
}

public class MahasiswaMain {

/**
 * @param args the command line arguments

```



```

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

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

    System.out.println("=====
=====");

    System.out.println("Masukkan data mahasiswa
secaraurut dari Nim terkecil");
    for (int i = 0; i < jumMhs; i++)
    {
        System.out.println("-----");
        System.out.print("nim\t : ");
        int nim = s.nextInt();
        System.out.print("Nama\t: ");
        String nama = s1.nextLine();
        System.out.print("Umur\t: ");
        int umur = s.nextInt();
        System.out.print("IPK\t: ");
        double ipk = s.nextDouble();

        Mahasiswa m = new Mahasiswa(nim, nama, umur,
ipk);

        data.tambah(m);
    }
    System.out.println("-----
-----");
    System.out.println("Data Keseluruhan Mahasiswa :
");
    data.tampil();
}

```

```

//          System.out.println("data mahasiswa setelah
sorting desc berdasarkan ipk");
//          data.selectionSort();
//          data.tampil();

System.out.println("_____");

System.out.println("_____");
        System.out.println("Pencarian data: ");
        System.out.println("Masukkan Nim Mahasiswa Yang
dicari: ");
        System.out.print("Nim: ");
        int cari = s.nextInt();
        System.out.println("Menggunakan Sequential
Search");
        int posisi = data.FindSeqSearch(cari);
        data.TampilPosisi(cari, posisi);
        data.TampilData(cari, posisi);

System.out.println("=====
=====");
        System.out.println("menggunakan binary search");
        posisi = data.FindBinarySearch(cari, 0, jumMhs-
1);

        data.TampilPosisi(cari, posisi);
        data.TampilData(cari, posisi);

```

6.4. Percobaan Pengayaan Divide and Conquer

```

public class MergeSorting {
    public void mergeSort(int [] data){
        Sort(data, 0, data.length-1);
    }
    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;

        //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];
        }
    }
    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();
    }
}

public class SortMain {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        int data[] = {10,40,30,50,70,20,100,90};
        System.out.println("Sorting dengan Merge Sort");
        MergeSorting mSort = new MergeSorting();
        System.out.println("data awal");
        mSort.printArray(data);
        mSort.mergeSort(data);
        System.out.println("Data urut");
        mSort.printArray(data);
    }

}

```

B. OUTPUT PROGRAM

6.2. Searching / Pencarian Menggunakan Algoritma Sequential Search

Low Help

029.5/305.000

Output - TestSearching (run-single) X

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

Output - TestSearching (run-single) X

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 : 2018 ditemukan pada indeks 1

Nim : 2018

Nama : Sinta Sanjaya

Umur : 22

IPK : 4.0

BUILD SUCCESSFUL (total time: 2 minutes 37 seconds)

6.3. Searching / Pencarian Menggunakan Binary Search

```
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
=====

Output - TestSearching (run-single) x
=====
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 : 2018 ditemukan pada indeks 1
Nim      : 2018
Nama     : Sinta Sanjaya
Umur     : 22
IPK      : 4.0
=====
menggunakan binary search
Data : 2018 ditemukan pada indeks 1
Nim      : 2018
Nama     : Sinta Sanjaya
Umur     : 22
IPK      : 4.0
BUILD SUCCESSFUL (total time: 2 minutes 27 seconds)
```

6.4. Percobaan Pengayaan Divide and Conquer

```
Output - TestSearching (run-single) x
[ ant -f D:\Algostrkdat\TestSearching -Dmb.internal.action.name=run.single -Djavac.i
[ tTest/SortMain.java -Drun.class=MergeSortTest.SortMain run-single
init:
Deleting: D:\Algostrkdat\TestSearching\build\build-jar.properties
deps-jar:
Updating property file: D:\Algostrkdat\TestSearching\build\build-jar.properties
Compiling 1 source file to D:\Algostrkdat\TestSearching\build\classes
compile-single:
run-single:
Sorting dengan Merge Sort
data awal
10 40 30 50 70 20 100 90
Data urut
10 20 30 40 50 70 90 100
BUILD SUCCESSFUL (total time: 2 seconds)
```

C. PENJELASAN

6.2.3 Pertanyaan

1. Jelaskan perbedaan metod TampilData dan Tampilposisi pada class PencarianMhs

Jawab: Pada Method TampiData hanya menampilkan data, pada method TampilPosis menampilkan data yang di posisi tersebut.

2. Jelaskan fungsi break pada kode program dibawah ini!

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

Jawab: fungsi break tersebut untuk menghentikan perulangan jika kondisi terpenuhi dan supaya bisa melanjutkan ke baris selanjutnya

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

Jawab: Program masih bisa dijalankan karena sequential search tidak mengutamakan urutan. Menggunakan data random(tidak berurutan) masih bisa ditemukan nilai yang dicari karena sequential mencari satu per satu tempat. Beda kasus dengan binary search yang harus menggunakan urutan nilai untuk menjalankan kode program

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!

Jawab :

Tidak bisa berjalan karena syarat dari binary search data harus urut. Karena juga terdapat nilai tengah juga maka nilai harus urut.

4. Jika Nim yang dimasukkan dari NIM terbesar ke terkecil (missal : 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 seach agar hasilnya sesuai

Jawab :

Binary search tidak bisa berjalan karena pada kode program menggunakan ascending yang mana dari kecil ke besar

```

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;
}

```

*Hanya mengubah operasi penbandingnya

```

=====
Masukkan data mahasiswa secara urut dari Nim terkecil
=====
nim      : 20215
Nama     : Schuzel
Umur     : 19
IPK      : 3.9
=====
nim      : 20214
Nama     : Wuhan
Umur     : 20
IPK      : 4.0
=====
nim      : 20212
Nama     : Ghuangzou
Umur     : 23
IPK      : 2.9
=====
nim      : 20211
Nama     : Berlin
Umur     : 24
IPK      : 3.2
=====
nim      : 20210
Nama     : Amsterdam
Umur     : 25
IPK      : 3.6
=====

```

```

=====
Data Keseluruhan Mahasiswa :
Nim = 20215
Nama = Schuzel
Umur = 19
IPK = 3.9
=====
Nim = 20214
Nama = Wuhan
Umur = 20
IPK = 4.0
=====
Nim = 20212
Nama = Ghuangzou
Umur = 23
IPK = 2.9
=====
Nim = 20211
Nama = Berlin
Umur = 24
IPK = 3.2
=====
Nim = 20210
Nama = Amsterdam
Umur = 25
IPK = 3.6
=====

```



```

Pencarian data:
Masukkan Nim Mahasiswa Yang dicari:
Nim: 20212
=====
menggunakan binary search
Data : 20212 ditemukan pada indeks 2
Nim      : 20212
Nama     : Ghuangzou
Umur     : 23
IPK      : 2.9
BUILD SUCCESSFUL (total time: 2 minutes 19 seconds)

```

5. Modifikasilah program diatas yang mana jumlah mahasiswa yang di inputkan sesuai dengan masukan dari keyboard.

```

public class Mahasiswa2 {
    int nim;
    String nama;
    int umur;
    double ipk;
    Mahasiswa2(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);
    }
}

public class PencaharianMHS1 {
    Mahasiswa2 [] ListMHS;
    int idx;
    PencaharianMHS1(int jumMhs){
        ListMHS = new Mahasiswa2[jumMhs];
    }
    void tambah(Mahasiswa2 m){
        if(idx < ListMHS.length)
        {

```

```

        ListMHS[idx]= m;
        idx++;
    }
    else
    {
        System.out.println("Data sudah
penuh!!");
    }
}

void tampil(){
    for (Mahasiswa2 m:ListMHS){
        m.Tampil();
        System.out.println("-----
---");
    }
}

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;
}

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");
        }
    }
    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, left, right);
            }
        }
        return -1;
    }
}

```

```

public class MahasiswaMain2 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);
        Scanner s1=new Scanner(System.in);

        System.out.print("Masukkan Jumlah Mahasiswa
: ");

        int jumMhs = s.nextInt();
        PencaharianMHS1 data = new
        PencaharianMHS1(jumMhs);
        System.out.println("-----
-----");

        System.out.println(" Masukkan data Mahasiswa
secara urut dari Nim yang terkecil ");
        System.out.println("-----
-----");

        for(int i=0; i<jumMhs; i++){
            System.out.println("-----
---");

            System.out.print("Nim\t: ");
            int nim=s.nextInt();
            System.out.print("Nama\t: ");
            String nama=s1.nextLine();
            System.out.print("Umur\t: ");
            int umur=s.nextInt();
            System.out.print("IPK\t: ");
            double ipk=s.nextDouble();
            Mahasiswa2 m= new Mahasiswa2(nim, nama,
umur, ipk);

            data.tambah(m);

```

```

        }
        System.out.println("-----
-----");
        System.out.println(" Data keseluruhan
Mahasiswa ");
        System.out.println("-----
-----");
        data.tampil();

System.out.println("_____
_____");

System.out.println("_____
_____");
        System.out.println("Pencarian Data: ");
        System.out.println("Masukkan Nim Mahasiswa
yang dicari: ");
        System.out.print("NIM: ");
        int cari=s.nextInt();
        System.out.println("Menggunakan sequential
Search");
        int posisi=data.FindSeqSearch(cari);
        data.TampilPosisi(cari, posisi);
        data.TampilData(cari, posisi);

System.out.println("=====
=====");
        System.out.println(" Menggunakan Binary
Search ");

```

```

System.out.println("=====
=====");

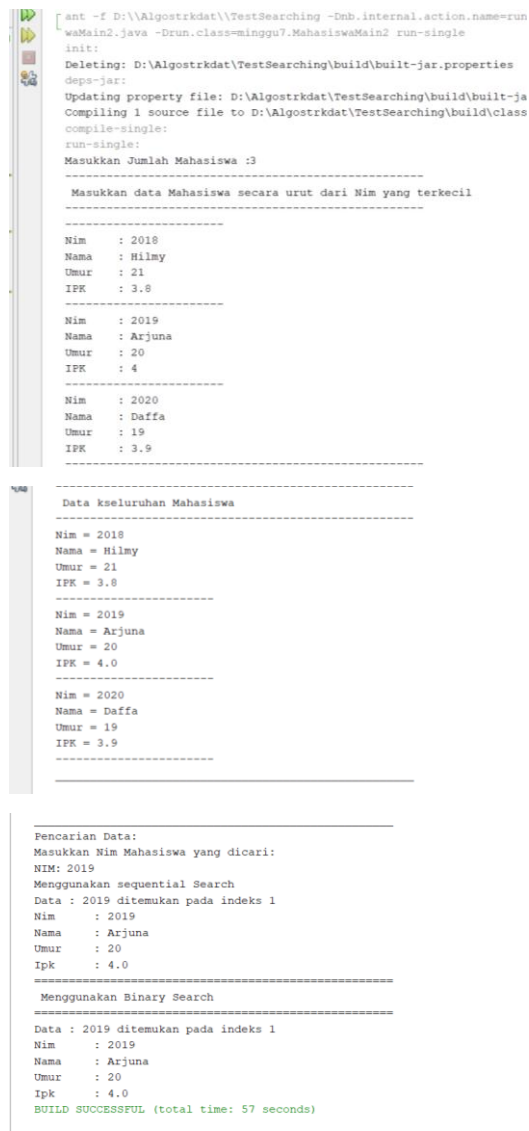
        posisi=data.FindBinarySearch(cari, 0,
jumMhs-1);

        data.TampilPosisi(cari, posisi);
        data.TampilData(cari, posisi);

    }

}

```



```

ant -f D:\Algostrkdat\TestSearching -Dmb.internal.action.name=run
waMain2.java -Drun.class=minggu7.MahasiswaMain2 run-single
init:
Deleting: D:\Algostrkdat\TestSearching\build\build-jar.properties
deps-jar:
Updating property file: D:\Algostrkdat\TestSearching\build\build-jar
Compiling 1 source file to D:\Algostrkdat\TestSearching\build\class
compile-single:
run-single:
Masukkan Jumlah Mahasiswa :3
-----
Masukkan data Mahasiswa secara urut dari Nim yang terkecil
-----
Nim      : 2018
Nama     : Hilmy
Umur     : 21
IPK      : 3.8
-----
Nim      : 2019
Nama     : Arjuna
Umur     : 20
IPK      : 4
-----
Nim      : 2020
Nama     : Daffa
Umur     : 19
IPK      : 3.9
-----

Data keseluruhan Mahasiswa
-----
Nim = 2018
Nama = Hilmy
Umur = 21
IPK = 3.8
-----
Nim = 2019
Nama = Arjuna
Umur = 20
IPK = 4.0
-----
Nim = 2020
Nama = Daffa
Umur = 19
IPK = 3.9
-----

Pencarian Data:
Masukkan Nim Mahasiswa yang dicari:
NIM: 2019
Menggunakan sequential Search
Data : 2019 ditemukan pada indeks 1
Nim      : 2019
Nama     : Arjuna
Umur     : 20
Ipk      : 4.0
=====
Menggunakan Binary Search
=====
Data : 2019 ditemukan pada indeks 1
Nim      : 2019
Nama     : Arjuna
Umur     : 20
Ipk      : 4.0
BUILD SUCCESSFUL (total time: 57 seconds)

```

D. KESIMPULAN

Di praktikum ini saya belajar sequential search, binary search dan merge sort, dan penerapan di studi kasus. Mohon maaf jika terdapat kesalahan dalam pengerjaan laporan ini.

TUGAS

1. Modifikasi percobaan searching diatas dengan ketentuan berikut ini - Sebelum dilakukan searching dengan binary search data harus dilakukan pengurutan dengan menggunakan algoritma sorting (pilih salah satu algoritma sorting dari pertemuan sebelumnya)

A. KODE PROGRAM

```
public class Mahasiswa {
    int nim;
    String nama;
    int umur;
    double ipk;
    Mahasiswa(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);
    }
}

public class PencarianMHS {
    Mahasiswa listMHs[] = new Mahasiswa[5];
    int idx;
    void tambah(Mahasiswa m){
        if(idx < listMHs.length)
        {
            listMHs[idx] = m;
            idx++;
        }
    }
}
```



```

        else
        {
            System.out.println("Data sudah
Penuh");
        }
    }
    void tampil(){
        for (Mahasiswa m : listMHs) {
            m.Tampil();

System.out.println("=====
===");
        }
    }
    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;
    }
    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");
    }
}

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;
}

void selectionSort(){
    for (int i=0; i<listMHs.length-1; i++)
    {
        int idxMin = i;
        for (int j=i+1; j<listMHs.length;
j++)
        {
            if(listMHs[j].nim <
listMHs[idxMin].nim)
            {
                idxMin = j;
            }
        }
        Mahasiswa tmp = listMHs[idxMin];
        listMHs[idxMin] = listMHs[i];
        listMHs[i] = tmp;
    }
}

public class MahasiswaMain {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        Scanner s1 = new Scanner(System.in);

        PencarianMHS data = new PencarianMHS();
        int jumMhs = 5;
    }
}

```

```

System.out.println("=====
=====");

        System.out.println("Masukkan data
mahasiswa secaraurut dari Nim terkecil");
        for (int i = 0; i < jumMhs; i++)
        {
            System.out.println("-----
-----");

            System.out.print("nim\t : ");
            int nim = s.nextInt();
            System.out.print("Nama\t: ");
            String nama = s1.nextLine();
            System.out.print("Umur\t: ");
            int umur = s.nextInt();
            System.out.print("IPK\t: ");
            double ipk = s.nextDouble();

            Mahasiswa m = new Mahasiswa(nim,
nama, umur, ipk);
            data.tambah(m);
        }
        System.out.println("-----
-----");

        System.out.println("Data Keseluruhan
Mahasiswa : ");
        data.tampil();
        System.out.println("data mahasiswa
setelah sorting desc berdasarkan ipk");
        data.selectionSort();
        data.tampil();

```

```

System.out.println("_____
_____");

System.out.println("_____
_____");
        System.out.println("Pencarian data: ");
        System.out.println("Masukkan Nim
Mahasiswa Yang dicari: ");
        System.out.print("Nim: ");
        int cari = s.nextInt();
        System.out.println("Menggunakan
Sequential Search");
        int posisi = data.FindSeqSearch(cari);
        data.TampilPosisi(cari, posisi);
        data.TampilData(cari, posisi);

System.out.println("=====
=====");
        System.out.println("menggunakan binary
search");
        posisi = data.FindBinarySearch(cari, 0,
jumMhs-1);
        data.TampilPosisi(cari, posisi);
        data.TampilData(cari, posisi);

```

B. OUTPUT PROGRAM

```
A M U T S I N G A N T
=====
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
=====

Output - TestSearching (run-single) x
data mahasiswa setelah sorting desc berdasarkan ipk
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 binary search
Data : 2018 ditemukan pada indeks 1
Nim      : 2018
Nama     : Sinta Sanjaya
Umur     : 22
IPK      : 4.0
BUILD SUCCESSFUL (total time: 2 minutes 0 seconds)
```

C. PENJELASAN

Di program ini sama dengan praktikum namun di pencarianMhs terdapat method selection sort dan nantinya di outputkan di main sebelum inputan NIM

D. KESIMPULAN

Kesimpulan yang saya dapat dari study kasus di atas bahwa kita bisa inputan tidak harus dari terkecil ke terbesar karena nantinya akan di urutkan melalui selection sort

2. Modifikasi percobaan searching diatas dengan ketentuan berikut ini

- Pencarian dilakukan berdasarkan Nama Mahasiswa (gunakan Algoritma Sequential Search)
- Jika terdapat nama yang sama? Bagaimana keluaran dari kode program ketika pencarian dilakukan pada nama yang sama!

A. KODE PROGRAM

```
public class Mahasiswa {
    int nim;
    String nama;
    int umur;
    double ipk;
    Mahasiswa(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);
    }
}

public class PencarianMHS {
    Mahasiswa listMHs[] = new Mahasiswa[5];
    int idx;
    void tambah(Mahasiswa m){
        if(idx < listMHs.length)
        {
            listMHs[idx] = m;
        }
    }
}
```

```

        idx++;
    }
    else
    {
        System.out.println("Data sudah
Penuh");
    }
}
void tampil(){
    for (Mahasiswa m : listMHs) {
        m.Tampil();

System.out.println("=====
");
    }
}
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;
}
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");
        }
    }

    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;
}

void selectionSort(){
    for (int i=0; i<listMHs.length-1; i++)
    {
        int idxMin = i;
        for (int j=i+1; j<listMHs.length; j++)
        {
            if(listMHs[j].nim <
listMHs[idxMin].nim)
            {
                idxMin = j;
            }
        }
        Mahasiswa tmp = listMHs[idxMin];
        listMHs[idxMin] = listMHs[i];
        listMHs[i] = tmp;
    }
}

public int FindSeqSearchName(String cari){
    int posisi = -1;
    for(int j = 0; j < listMHs.length; j++)
    {

        if(cari.equalsIgnoreCase(listMHs[j].nama)){
            posisi = j;
            break;
        }
    }
}

```

```

        return posisi;
    }

    public void TampilPosisi1 (String x, int
pos1){
        if(pos1 != -1)
        {
            System.out.println("Data : " + x + "
ditemukan pada indeks " + pos1);
        }
        else
        {
            System.out.println("Data " + x + "
Tidak ditemukan ");
        }
    }

    public void TampilData1(String x, int pos1){
        if(pos1 != -1)
        {
            System.out.println("Nim\t : " +
listMHs[pos1].nim);
            System.out.println("Nama\t : "+ x);
            System.out.println("Umur\t : "+
listMHs[pos1].umur);
            System.out.println("IPK\t : "+
listMHs[pos1].ipk);
        }
        else
        {
            System.out.println("data " + x +
"Tidak ditemukan");
        }
    }
}

public class MahasiswaMain {

```

```

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    Scanner s = new Scanner(System.in);
    Scanner s1 = new Scanner(System.in);

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

    System.out.println("=====
    =====");

    System.out.println("Masukkan data
    mahasiswa secara urut dari Nim terkecil");
    for (int i = 0; i < jumMhs; i++)
    {
        System.out.println("-----
    --");

        System.out.print("nim\t : ");
        int nim = s.nextInt();
        System.out.print("Nama\t: ");
        String nama = s1.nextLine();
        System.out.print("Umur\t: ");
        int umur = s.nextInt();
        System.out.print("IPK\t: ");
        double ipk = s.nextDouble();

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

```

```

        System.out.println("-----
-----");
        System.out.println("Data Keseluruhan
Mahasiswa : ");
        data.tampil();
        System.out.println("data mahasiswa setelah
sorting desc berdasarkan ipk");
        data.selectionSort();
        data.tampil();

System.out.println("_____
__");

System.out.println("_____
__");
        System.out.println("Pencarian data: ");
        System.out.println("Masukkan Nim Mahasiswa
Yang dicari: ");
        System.out.print("Nim: ");
        int cari = s.nextInt();
        //          System.out.println("Menggunakan
Sequential Search");
//          int posisi = data.FindSeqSearch(cari);
//          data.TampilPosisi(cari, posisi);
//          data.TampilData(cari, posisi);

System.out.println("=====
=====");
        System.out.println("menggunakan binary
search");
        posisi = data.FindBinarySearch(cari, 0,
jumMhs-1);
        data.TampilPosisi(cari, posisi);
        data.TampilData(cari, posisi);

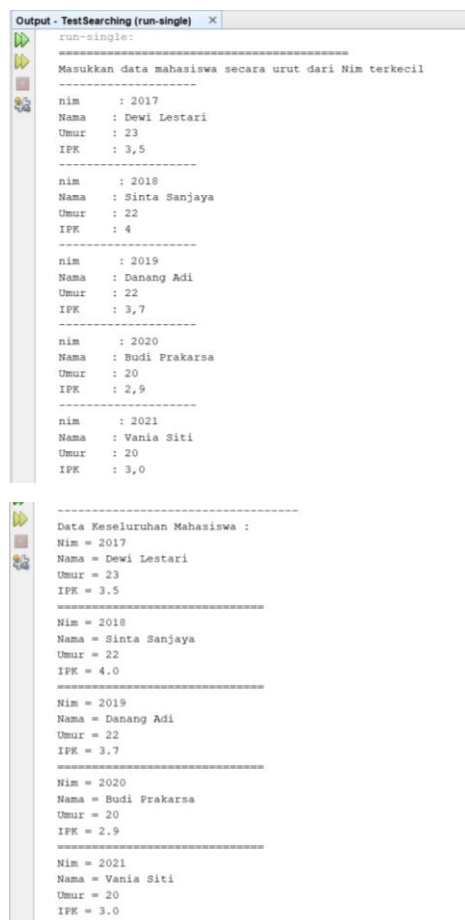
```

```

        System.out.print("Nama: ");
        String caril = s1.nextLine();
        int Posisi1 =
data.FindSeqSearchName(caril);
        data.TampilPosisi1(caril, Posisi1);
        data.TampilData1(caril, Posisi1);
    }
}

```

B. OUTPUT PROGRAM



The screenshot shows the output of a Java program. The window title is "Output - TestSearching (run-single)". The output text is as follows:

```

run-single:
=====
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
=====

```

```

Output - TestSearching (run-single) x
=====
data mahasiswa setelah sorting desc berdasarkan ipk
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 : 2018 ditemukan pada indeks 1
Nim      : 2018
Nama     : Sinta Sanjaya
Umur     : 22
IPK      : 4.0
Nama: danang adi
Data : danang adi ditemukan pada indeks 2
Nim      : 2019
Nama     : danang adi
Umur     : 22
IPK      : 3.7
BUILD SUCCESSFUL (total time: 2 minutes 8 seconds)

```

C. PENJELASAN

Di program ini sama dengan praktikum namun di pencarianMhs terdapat method FindSequentialSearchName dengan parameter String isinya sama dengan praktikum diatas namun nim diganti dengan nama dan terdapat TampilData1 dan TampilPosis1 untuk menampilkan data yang kita input melalui main dan dikerluarkan di main juga

D. KESIMPULAN

Kesimpulan yang saya dapat dari study kasus di atas bahwa kita bisa mencari berdasarkan Nama

3. Terdapat sebuah data array 2 dimensi sebagai berikut:

Indeks	0	1	2	3	4
0	45	78	7	200	80
1	90	1	17	100	50
2	21	2	40	18	65

Berdasarkan data di atas buatlah program untuk melakukan pencarian data pada array dua dimensi tersebut, dimana data yang dicari di inputkan melalui keyboard (menggunakan sequential search) !

A. KODE PROGRAM

```

public class Array2D {
    int baris, kolom;
    Array2D(int b, int k){

```

```

        baris = b;
        kolom = k;
    }
}

public class PencarianArray2D {
    Array2D[] arr[] = new Array2D[3][5];
    void pencarian(){
        Scanner sc = new Scanner(System.in);
        Array2D arr = new Array2D(3,5);
        //      int[] array[] = {
        //          {45, 78, 7, 200, 80},
        //          {90, 1, 17, 100, 50},
        //          {21, 2, 40, 18, 65}};
        int[] array[] = new
int[arr.baris][arr.kolom];
        System.out.print("Masukkan baris array
:");

        arr.baris = sc.nextInt();
        System.out.print("Masukkan kolom array
:");

        arr.kolom = sc.nextInt();
        for (int i = 0; i < arr.baris; i++) {
            for (int j = 0; j < arr.kolom; j++) {
                System.out.print("Masukkan array
ke-["+i+"]["+j+"]");
                array[i][j] = sc.nextInt();
            }
        }
        boolean ketemu = false;
        System.out.println("cari array: ");
        int cari = sc.nextInt();
        for (int i = 0; i < arr.baris; i++) {
            for (int j = 0; j < arr.baris; j++) {
                if(cari == array[i][j]){

```



```

        System.out.println("Data " +
array[i][j]+ " Data ditemukan pada index ke-
["+i+"]["+j+"]");

        ketemu = true;
        break;
    }
}
}
if(ketemu == false){
    System.out.println("Data tidak
ditemukan!");
}
}
}
public class Array2DMain {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Array2D arr = new Array2D(3,5);
        int[] array[] = new
int[arr.baris][arr.kolom];
        PencarianArray2D ary = new
PencarianArray2D();
        ary.pencarian();
    }

}

```

B. OUTPUT PROGRAM

```

Output - TestSearching (run-single) X
[gle -Djava.class.path=Tugas/Array2DMain.java -Drun.c
in run-single
init:
Deleting: D:\Algostrkdat\TestSearching\build\build-
deps-jar:
[Updating property file: D:\Algostrkdat\TestSearchin
operties
Compiling 1 source file to D:\Algostrkdat\TestSearc
compile-single:
run-single:
Masukkan baris array :3
Masukkan kolom array :5
Masukkan array ke-[0][0] :45
Masukkan array ke-[0][1] :78
Masukkan array ke-[0][2] :7
Masukkan array ke-[0][3] :200
Masukkan array ke-[0][4] :80
Masukkan array ke-[1][0] :90
Masukkan array ke-[1][1] :1
Masukkan array ke-[1][2] :17
Masukkan array ke-[1][3] :100
Masukkan array ke-[1][4] :50
Masukkan array ke-[2][0] :21
Masukkan array ke-[2][1] :2
Masukkan array ke-[2][2] :40
Masukkan array ke-[2][3] :18
Masukkan array ke-[2][4] :65
cari array:
78
Data 78 Data ditemukan pada index ke-[0][1]
BUILD SUCCESSFUL (total time: 1 minute 0 seconds)

```

C. PENJELASAN

Di class Array2D terdapat tipe data interger dengan variable baris dan kolom selanjutnya si class PencarianArray2D terdapat array dua dimensi [3][5] dan void pencarian didalamnya terdapat inputan inputan tersebut akan jalan di main class

D. KESIMPULAN

di program ini saya belajar bagaimana caranya mencari data di Array 2 dimensi

4. Terdapat sebuah data array 1 dimensi sebagai berikut

0	1	2	3	4	5	6	7	8	9
12	17	2	1	70	50	90	17	2	90

Buatlah program untuk mengurutkan array tersebut (boleh memilih metode pengurutan) selanjutnya lakukan pencarian dan mencetak isi array yang nilainya terbesar, dan mencetak ada berapa buah nilai terbesar tersebut serta berada dilokasi mana saja nilai terbesar tersebut! (menggunakan binary search)

A. KODE PROGRAM

```

public class Array1D {
    int kolom;

    Array1D(int k){
        kolom = k;
    }

    void Tampil(){
        System.out.print(kolom+" | ");
    }
}

```

```

public class Array1DService {
    Array1D listArray[] = new Array1D[10];
    int idx;
    void tambah(Array1D m){
        if(idx < listArray.length)
        {
            listArray[idx] = m;
            idx++;
        }
        else
        {
            System.out.println("Data sudah
Penuh");
        }
    }
    void tampil(){
        for (Array1D m : listArray) {
            m.Tampil();

        }
    }
    int tampil2(){
        int kol1 = 0;
        for (int i = 1; i <= 1; i++) {
            int kolom = listArray[8].kolom;
            //int kolom2 = listArray[9].kolom;
            System.out.println(kolom);
            //System.out.println(kolom2);
            kol1 = kolom;
            //kol2 = kolom2;
        }
        return kol1;
    }
    // int tampil3(){

```

```

//      int kol2 = 0;
//      for (int i = 1; i <= 1; i++) {
//          int kolom2 = listArray[9].kolom;
//          System.out.println(kolom2);
//          kol2 = kolom2;
//      }
//      return kol2;
//  }

void selectionSort(){
    for (int i=0; i<listArray.length-1; i++)
    {
        int idxMin = i;
        for (int j=i+1; j<listArray.length;
j++)
        {
            if(listArray[j].kolom <
listArray[idxMin].kolom)
            {
                idxMin = j;
            }
        }
        Array1D tmp = listArray[idxMin];
        listArray[idxMin] = listArray[i];
        listArray[i] = tmp;
    }
}

void nilaiTerbesar(){
    for (int i=0; i<listArray.length-1; i++)
    {
        int idxMin = i;
        for (int j=i+1; j<listArray.length;
j++)
        {

```

```

        if(listArray[j].kolom <
listArray[idxMin].kolom)
        {
            idxMin = j;
        }
    }
    Array1D tmp = listArray[idxMin];
    listArray[idxMin] = listArray[i];
    listArray[i] = tmp;
}

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

int FindBinarySearch2(int cari, int left, int
right){
    int mid;
    if(right>=left){
        mid = (left + right)/2;

```

```

        if(cari == listArray[mid].kolom){
            return (mid);
        }else if(listArray[mid].kolom > cari){
            return FindBinarySearch(cari,
left, mid-1);
        }else{
            return FindBinarySearch(cari,
mid+1,right);
        }
    }
    return -2;
}

public void Tampilposisi(int x, int pos){
    if(pos!=-1){
        for (int i = 0+1; i < 2; i++) {
            System.out.println("data : "+x+"
ditemukan pada indeks "+pos);
        }
        System.out.println("data : "+x+"
ditemukan pada indeks "+pos);
    }else{
        System.out.println("data "+x+" tidak
ditemukan");
    }
}

public void Tampilposisi2(int x, int pos){
    if(pos!=-2){
        for (int i = 0+1; i < 2; i++) {
            System.out.println("data : "+x+"
ditemukan pada indeks "+pos);
        }
        System.out.println("data : "+x+"
ditemukan pada indeks "+pos);
    }else{

```

```

        System.out.println("data "+x+" tidak
ditemukan");
    }
}

public class Array1DMain {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Array1DService list = new
Array1DService();

        Array1D t0 = new Array1D(12);
        Array1D t1 = new Array1D(17);
        Array1D t2 = new Array1D(2);
        Array1D t3 = new Array1D(1);
        Array1D t4 = new Array1D(70);
        Array1D t5 = new Array1D(50);
        Array1D t6 = new Array1D(90);
        Array1D t7 = new Array1D(17);
        Array1D t8 = new Array1D(2);
        Array1D t9 = new Array1D(90);

        list.tambah(t0);
        list.tambah(t1);
        list.tambah(t2);
        list.tambah(t3);
        list.tambah(t4);
        list.tambah(t5);
        list.tambah(t6);
        list.tambah(t7);
    }
}

```

```

        list.tambah(t8);
        list.tambah(t9);

System.out.println("_____
_____");
        System.out.println("Menampilkan Data
Semua");
        list.tampil();

System.out.println("\n_____
_____");
        System.out.println("Menampilkan Data yang
sudah Urut!");
        list.selectionSort();
        list.tampil();

System.out.println("\n_____
_____");
        System.out.println("Menampilkan Data Yang
terbesar");
        //System.out.println(t9);
        //list.tampil2();
        int tampil =
list.FindBinarySearch(list.tampil2(), 0, 10);
//        int tampil2 =
list.FindBinarySearch2(list.tampil3(), 0, 10);

System.out.println("_____
_____");
        list.Tampilposisi(list.tampil2(), tampil);
        list.Tampilposisi(list.tampil2(),
tampil+1);
    }

```


}

B. OUTPUT PROGRAM

```
Menampilkan Data Semua
12 | 17 | 2 | 1 | 70 | 50 | 90 | 17 | 2 | 90 |

Menampilkan Data yang sudah Urut!
1 | 2 | 2 | 12 | 17 | 17 | 50 | 70 | 90 | 90 |

Menampilkan Data Yang terbesar
90

90
data : 90 ditemukan pada indeks 8
data : 90 ditemukan pada indeks 8
90
data : 90 ditemukan pada indeks 9
data : 90 ditemukan pada indeks 9
BUILD SUCCESSFUL (total time: 0 seconds)
```

C. PENJELASAN

di Class Array1D terdapat integer kolom dan method Tampil, di Class Array1DService terdapat method Tampil1 gunanya untuk menampilkan semua data dan Tampil2 gunanya untu menampilkan bilangan terbesar dan method TampilPosisi dan Tampil data di main Class langsung saya deklarasikan bilangan jadi tinggal outuput saja tidak perlu memasukkan bilangan dan selanjutnya tinggal menampilkan data terbesar dan data pada dan data tersebut tampil dengan posisi indeksnya

D. KESIMPULAN

Kesimpulan yang saya dapat menampilkan data terbesar dengan array 1 dimensi dengan posis indeksnya