LAPORAN HASIL PRAKTIKUM

ALGORITMA DAN STRUKTUR DATA

JOBSHEET 6

OLEH :

IVAN RIZAL AHMADI

NIM. 2341760128

SIB-1F/13

D-IV SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

6.2. Searching / Pencarian Menggunakan Algoritma Sequential Search

6.2.1 Sequential Search Menggunakan Array

1. Buat folder baru dengan nama Praktikum06. Buat file dengan nama Sorting.java
2. Tambahkan method sequentialSearch() yang melakukan pencarian data bertipe integer di dalam array of integer

```
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(x:"Data tidak ditemukan");
}
```

3. Tambahkan fungsi main sebagai berikut

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

4. Compile dan run program

```
Data ditemukan pada indeks ke-5
Data tidak ditemukan
PS C:\Users\ivanr\OneDrive\Desktop\praktikum06>
```

✞ Langkah-langkah Percobaan Sequential Search

1. Buatlah Project baru pada Netbeans dengan nama **`TestSearching`**
2. Kemuadian buat packages baru dengan nama **`minggu7`**.
3. Buat class **`Mahasiswa`**, kemudian deklarasikan atribut berikut ini:

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

4. Buatlah konstruktor dengan nama **`Mahasiswa`** dengan parameter (**`int ni, String n, int u, double i`**) kemudian Isi konstruktor tersebut dengan kode berikut!

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

5 Buatlah Void tampil

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

6. Buat class baru dengan nama **`PencarianMhs`** seperti di bawah ini!

```
public class pencarianMhs {
    mahasiswa listMhs[] = new mahasiswa[5];
    int idx;
```

7. Tambahkan method **`tambah()`** di dalam class tersebut! Method **`tambah()`** digunakan untuk menambahkan objek dari class Mahasiswa ke dalam atribut listMhs.

```
void tambah(mahasiswa m) {
    if (idx < listMhs.length) {
        listMhs[idx] = m;
        idx++;
    } else {
        System.out.println(x:"Data Sudah Penuh !!");
    }
}
```

8. Tambahkan method **`tampil()`** di dalam class **`PencarianMhs`**! Method **`tampil()`** digunakan untuk menampilkan semua data mahasiswa-mahasiswa yang ada di dalam class tersebut! Perhatikan penggunaan sintaks for yang agak berbeda dengan for yang telah dipelajari sebelumnya, meskipun secara konsep sebenarnya mirip.

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

9. Tambahkan method **`FindSeqSearch`** bertipe integer dengan parameter **`cari`** bertipe integer. Kemudian Deklarasikan isi method **`FindSeqSearch`** dengan algoritma pencarian data menggunakan teknik sequential searching.

```
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;
```

10. Buatlah method **`Tampilpoisisi`** bertipe void dan Deklarasikan isi dari method

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

11. Buatlah method **`TampilData`** bertipe void dan Deklarasikan isi dari method **`TampilData.`**

```
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");
    }
}
```

12. Buatlah class baru dengan nama **`MahasiswaMain`** tambahkan method **`main`** seperti pada gambar berikut!

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

13. Di dalam method **`main()`**, buatlah sebuah objek PencarianMhs dan buatlah 5 objek mahasiswa kemudian tambahkan semua objek mahasiswa tersebut dengan memanggil fungsi tambah pada objek PencarianMhs.

```
Scanner s = new Scanner(System.in);
Scanner sl = new Scanner(System.in);

pencarianMhs data = new pencarianMhs();
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 = sl.nextLine();
    System.out.print(s:"Umur\t: ");
    int umur = s.nextInt();
    System.out.print(s:"IPK\t: ");
    double ipk = s.nextDouble();

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

14. Panggil method **`tampil()`** untuk melihat semua data yang telah dimasukan.

```
System.out.println(x:"_____________________________________________________");
System.out.println(x:"Data Keseluruhan Mahasiswa : ");
data.tampil();
```

15. Untuk melakukan pencarian berdasarkan NIM mahasiswa. Buatlah variable **`cari`** yang dapat menampung masukan dari keyboard lalu panggil method **`FindSeqSearch`** dengan isi parameternya adalah variable cari.

```
System.out.println(x:"_____________________________________________");
System.out.println(x:"Pencarian Data : ");
System.out.println(x:"Masukkan Nim Mahasiswa yang dicari : ");
System.out.print(s:"NIM\t: ");
int cari = s.nextInt();
System.out.println(x:"Menggunakan sequential Search");
int posisi = data.findSeqSearch(cari);
```

16. Lakukan pemanggilan method **`Tampilposisi`** dari class **`PencarianMhs.`**

```
data.tampilPosisi(cari, posisi);
```

17. Lakukan pemanggilan method **`TampilData`** dari class **`PencarianMhs.`**

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

18. Jalankan dan amati hasilnya.

6.2.2. Verifikasi Hasil Percobaan

Cocokkan hasil kode program anda dengan gambar berikut ini.

```
-----------------------------------------------------
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     : 2018 ditemukan pada indeks 1
Nim      : 2018
Nama     : Sinta Sanjaya
umur     : 22
IPK      : 4.0
PS C:\Users\ivanr\OneDrive\Desktop\praktikum06>
```

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

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

2. Jelaskan perbedaan metod **`TampilData`** dan **`Tampilposisi`** pada class PencarianMhs

   - **Method Tampil Data || TampilData(int x,int pos) :void**

   Merupakan sebuah method yang digunakan untuk menampilkan data yang ingin dicari oleh pengguna, dimana pengguna memasukan Nim dan kemudian akan di proses dan menampilkan data berupa NIM, NAMA, UMUR, IPK. apabila nim yang dimasukkan tidak ada pada program, maka method tampil data ini akan menampilkan output seperti berikut "Data "+ x +" Tidak ditemukan"

   - **Method TampilPosisi || Tampilpoisisi(int x,int pos): void**

   Adalah sebuah method yang digunakan untuk menampilan posisi data yang ingin dicari oleh pengguna, dimana disaat pengguna memasukkan nim kdan meudian akan di proses dan menampilkan posisi data yang dicari berada pada index ke berapa, apabila nim yang dimasukkan tidak ada pada program, maka method tampil data ini akan menampilkan output seperti berikut "data\t"+ x +" tidak ditemukan"

3. Jelaskan fungsi **`break`** pada kode program dibawah ini!

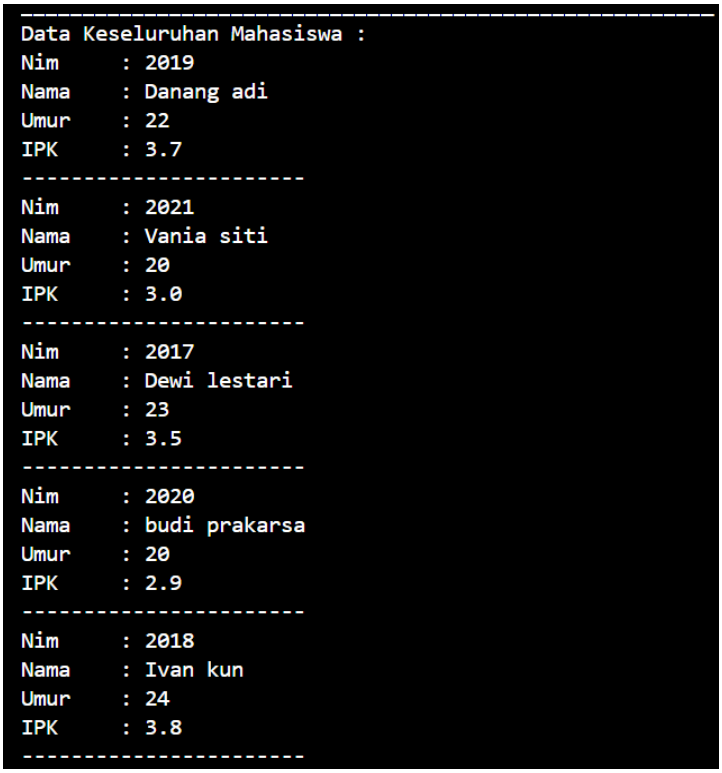
```java
if (listMhs[j].nim == cari) {
    posisi = j;
    break;
}
```

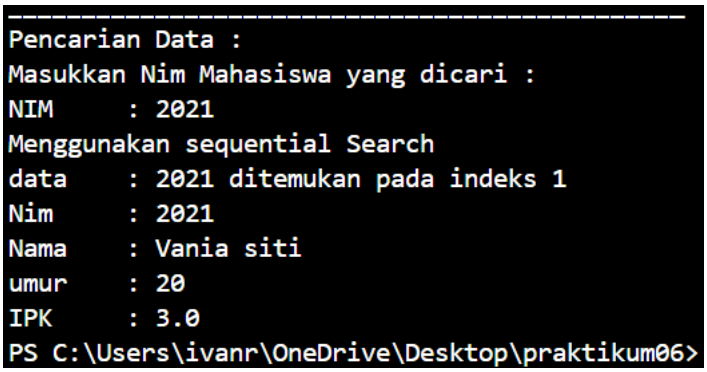
   - Break pada program diatas berfungsi apabila/jika listMhs[j].nim sama dengan variable cari atau data listMhs[j] berhasil ditemukan. maka fungsi break adaah menghentikan perulangan pencarian data. karena data yang dicari sudah ditemukan

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

Ya Hasil yang dikeluarkan Benar, Karena pada algoritma Sequential Search dilakukan Proses pencarian dilakukan dengan membandingkan elemen array satu per satu secara beruntun mulai dari elemen pertama sampai elemen yang dicari sudah ditemukan atau sampai semua elemen sudah diperiksa sehingga Kumpulan data tidak harus dalam keadaan terurut program tetap bisa dijalankan.

Untuk Bukti Compile programnya adalah sebagai berikut :





Searching / Pencarian Menggunakan Binary Search

6.3.1. Langkah-langkah Percobaan Binary Search menggunakan Array

1. Tambahkan method binarySearchAsc() pada file Sorting.java

```java
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. Tambahkan baris program untuk menguji method binarySearchAsc() pada fungsi main()

```java
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(x:"Data tidak ditemukan");
}
}
```

3. Run dan compile program

```
Data ditemukan pada indeks ke-5
Data tidak ditemukan
Data ditemukan pada indeks ke-1
PS C:\Users\ivanr\OneDrive\Desktop\praktikum06>
```

6.3.2. Langkah-langkah Percobaan Binary Search menggunakan Array of Object

1. Pada percobaan 6.2.2 (sequential search) tambahkan method **`FindBinarySearch`** bertipe integer pada class **`PencarianMhs`**. Kemudian Deklarasikan isi method **`FindBinarySearch`** dengan algoritma pencarian data menggunakan teknik binary searching.

```
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. Panggil method **`FindBinarySearch`** terdapat pada class **`PencarianMhs`** di kelas **`Mahasiswamain.`** Kemudia panggil **`method tampilposisi`** dan **`tampilData`**

```
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);
```

6.3.2. Verifikasi Hasil Percobaan

```
-------------------------
Data Keseluruhan Mahasiswa :
Nim      : 2017
Nama     : Dewi Lestari
Umur     : 23
IPK      : 3.5
-------------------------
Nim      : 2018
Nama     : Sintya 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     : Sintya sanjaya
umur     : 22
IPK      : 4.0
=============================================
Menggunakan Binary Search
data     : 2018 ditemukan pada indeks 1
Nim      : 2018
Nama     : Sintya sanjaya
umur     : 22
IPK      : 4.0
PS C:\Users\ivanr\OneDrive\Desktop\praktikum06>
```

1. Tunjukkan pada kode program yang mana proses divide dijalankan!

Divede adalah dimana terjadi Proses pembagian dilakukan, pada source code saya terdapat pada class "pencarianMHS" dan pada variable berikut =

```
mid = (left + right) / 2;
```

2. Tunjukkan pada kode program yang mana proses conquer dijalankan! =

   Conquer adalah dimana terjadi proses pengurutan dilakukan, pada source code saya terdapat pada class "pencarianMHS", di baris ini =

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

3. Jika data Nim yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian!=

   - Program masih dapat di Run, akan tetapi data yang dicari tidak dapat ditemukan karena pada algoritma BinarySearch Teknik pencarian = data dibagi menjadi dua bagian untuk setiap kali proses pencarian. Data awal harus dalam kondisi terurut. Sehingga harus dilakukan proses sorting terlebih dahulu untuk data awal. Untuk bukti hasil compile nya adalah sebagai berikut=

```
Data Keseluruhan Mahasiswa :
Nim      : 2019
Nama     : Danang Adi
Umur     : 22
IPK      : 3.5
-----------------------
Nim      : 2017
Nama     : Dewi tari
Umur     : 23
IPK      : 3.5
-----------------------
Nim      : 2012
Nama     : vania za
Umur     : 22
IPK      : 3.0
-----------------------
Nim      : 2020
Nama     : Budi
Umur     : 20
IPK      : 2.9
-----------------------
Nim      : 2018
Nama     : sinta
Umur     : 22
IPK      : 4.0
-----------------------
```

```
Pencarian Data :
Masukkan Nim Mahasiswa yang dicari :
NIM      : 2018
Menggunakan sequential Search
data     : 2018 ditemukan pada indeks 4
Nim      : 2018
Nama     : sinta
umur     : 22
IPK      : 4.0
==========================================
Menggunakan Binary Search
data     2018 tidak ditemukan
Data 2018 Tidak ditemukan
PS C:\Users\ivanr\OneDrive\Desktop\praktikum06
```

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 ! =

```
Data Keseluruhan Mahasiswa :
Nim     : 20215
Nama    : irjo
Umur    : 22
IPK     : 3.0
----------------------
Nim     : 20214
Nama    : erjo
Umur    : 33
IPK     : 2.0
----------------------
Nim     : 20213
Nama    : wrjo
Umur    : 11
IPK     : 2.0
----------------------
Nim     : 20212
Nama    : www
Umur    : 3
IPK     : 3.0
----------------------
Nim     : 20211
Nama    : ooo
Umur    : 4
IPK     : 2.0
----------------------
```

```
Pencarian Data :
Masukkan Nim Mahasiswa yang dicari :
NIM     : 20215
Menggunakan sequential Search
data    : 20215 ditemukan pada indeks 0
Nim     : 20215
Nama    : irjo
umur    : 22
IPK     : 3.0
================================================
Menggunakan Binary Search
data    20215 tidak ditemukan
Data 20215 Tidak ditemukan
PS C:\Users\ivanr\OneDrive\Desktop\praktikum06>
```

Hasil Searching menggunakan Algoritma Binary Search tersebut tidak sesuai, ss diatas tersebut sebagai pembuktian output programnya.

Jadi, agar data yang di inginkan dapat ditemukan, saya mengubah kode program pada class "pencarianMHS" pada method "findBinarySearch", karena data yang di inputkan dari yang terbesar ke terkecil (descending) maka source code nya seperti berikut ini :

```
else if (listMhs[mid].nim > cari)
```

→

```
else if (listMhs[mid].nim < cari) {
```

Dan untuk compile program nya akan menjadi seperti berikut =

```
Data Keseluruhan Mahasiswa :
Nim     : 20215
Nama    : ivan
Umur    : 22
IPK     : 2.0
----------------------
Nim     : 20214
Nama    : oi
Umur    : 22
IPK     : 2.0
----------------------
Nim     : 20213
Nama    : io
Umur    : 22
IPK     : 2.0
----------------------
Nim     : 20212
Nama    : werr
Umur    : 22
IPK     : 2.0
----------------------
Nim     : 20211
Nama    : wq
Umur    : 22
IPK     : 2.0
----------------------
```

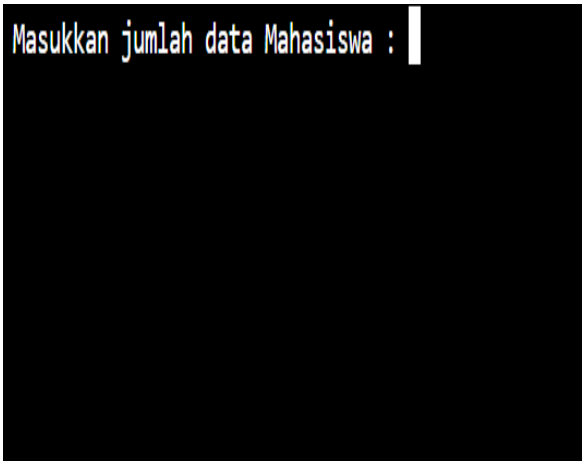
```
Pencarian Data :
Masukkan Nim Mahasiswa yang dicari :
NIM     : 20215
Menggunakan sequential Search
data    : 20215 ditemukan pada indeks 0
Nim     : 20215
Nama    : ivan
umur    : 22
IPK     : 2.0
================================================
Menggunakan Binary Search
data    : 20215 ditemukan pada indeks 0
Nim     : 20215
Nama    : ivan
umur    : 22
IPK     : 2.0
```

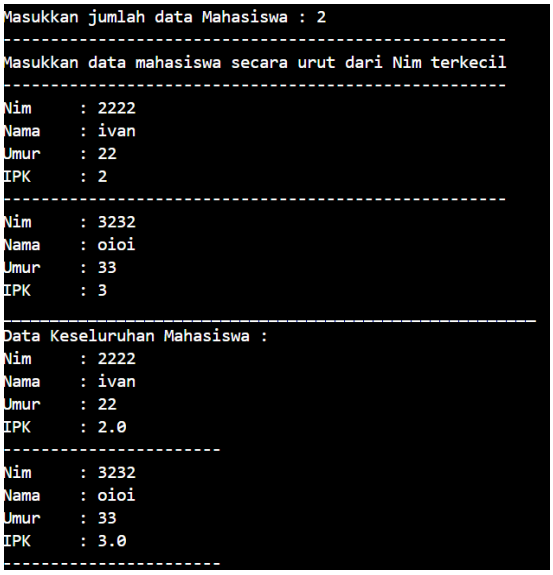
5. Modifikasilah program diatas yang mana jumlah mahasiswa yang di inputkan sesuai dengan masukan dari keyboard. = Saya memodifikasi nya dengan menambahkan variable seperti di bawah ini.

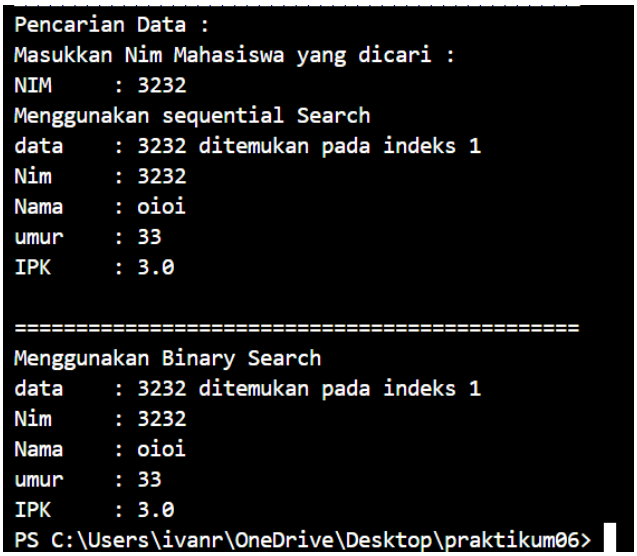
```
pencarianMhs data = new pencarianMhs();
System.out.print(s:"Masukkan jumlah data Mahasiswa : ");
int jumMhs = s.nextInt();
mahasiswa[] inputan=new mahasiswa[jumMhs];
data.listMhs=inputan;
```

disini saya memodifikasi pada class "pencarianMhs"

Variabel ini digunakan untuk mendeklarasikan variabel dan objek, memasukkan jumlah data mahasiswa, menginisialisasi array untuk menyimpan data mahasiswa, dan menghubungkan array tersebut dengan objek. Sehingga hasil Compile program saya akan menjadi seperti berikut =







Percobaan Pengayaan Divide and Conquer

6.4.1. Langkah-langkah Percobaan Merge Sort

1 Buatlah Package baru pada NetBeans dengan nama **`MergeSortTest`**

7 Tambahkan class **`MergeSorting`** pada package tersebut

8 Pada class **`MergeSorting`** buatlah method **`mergeSort`** yang menerima parameter data array yang akan diurutkan

```
public void mergeSort(int[] data){
```

9 Buatlah method **`merge`** untuk melakukan proses penggabungan data dari bagian kiri dan kanan.

```
private void merge(int data[], int left, int midle, int right){
```

10. implementasikan proses merge

```
private void merge(int data[], int left, int midle, int right){
    int[] temp = new int[data.length];
    for(int i= left; i<= right; i++){
        temp[i] = data[i];
    }
    int a = left;
    int b = midle +1;
    int c = left;

    //membandingkan setiap bagian
    while(a <= midle && b<= right){
        if(temp[a] <= temp[b]){
            data[c] = temp[a];
            a++;
        }else{
            data[c] = temp[b];
            b++;
        }
        c++;
    }
    int s= midle - a;
    for(int i=0; i<=s; i++){
        data[c+i] = temp[a+i];
    }
}
```

11. buatlah method sort

```
private void sort(int data[], int left, int right){
```

12. implementasikan pada method sort

```
private void sort(int data[], int left, int right){
    if(left < right){
        int midle = (left + right)/2;
        sort(data, left, midle);
        sort(data, midle + 1, right);
        merge(data, left, midle, right);
    }
}
```

13 Pada method **`mergeSort`**, panggil method **`sort`** dengan parameter data yang ingin diurutkan serta range data awal sampai dengan akhir.

14 Tambahkan 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();
    }
}
```

15 Sebagai langkah terakhir, deklarasikan data yang akan diurutkan kemudian panggil proses sorting pada class SortMain

```
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");
        mergeSorting mSort = new mergeSorting();
        System.out.println(x:"data awal");
        mSort.printArray(data);
        mSort.mergeSort(data);
        System.out.println(x:"Setelah diurutkan");
        mSort.printArray(data);
    }
}
```

6.4.2. Verifikasi Hasil Percobaan

```
Sorting dengan merge sort
data awal
10 40 30 50 70 20 100 90
Setelah diurutkan
10 20 30 40 50 70 90 100
PS C:\Users\ivanr\OneDrive\Desktop\praktikum06>
```

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!

```
    Arrays.sort(data.listMhs, Comparator.comparing(m -> m.nama));

    System.out.println(x:"_____________________________________________________");
    System.out.println(x:"Data Keseluruhan Mahasiswa : ");
    data.tampil();

    System.out.println(x:"_____________________________________________");
    System.out.println(x:"Pencarian Data : ");
    System.out.println(x:"Masukkan Nama Mahasiswa yang dicari : ");
    System.out.print(s:"Nama\t: ");
    String cari = s.next();
    System.out.println(x:"Menggunakan Binary Search");
    int[] posisi = data.findBinarySearch(cari, left:0, jumMhs - 1);

    data.tampilPosisi(cari, posisi);
    data.tampilData(cari, posisi);
}
```

Program yang saya buat ini mengimplementasikan pengurutan data mahasiswa berdasarkan nama menggunakan algoritma Binary search untuk mencari nama mahasiswa yang diinputkan pengguna, kemudian menampilkan hasil pencarian beserta posisinya dalam array.

Array `mahasiswa` diurutkan berdasarkan nama menggunakan metode `Arrays.sort`. - Pengurutan dilakukan dengan menggunakan `Comparator.comparing` untuk membandingkan berdasarkan atribut `nama` dari objek Mahasiswa.

```
import java.util.Scanner;
import java.util.Arrays;
import java.util.Comparator;
```

Dan untuk hasil Compile nya adalah sebagai berikut =

```
Masukkan jumlah data Mahasiswa : 2
-----------------------------------------------------
Masukkan data mahasiswa secara urut dari Nama terkecil
-----------------------------------------------------
Nim     : 2323
Nama    : ivan
Umur    : 22
IPK     : 2
-----------------------------------------------------
Nim     : 5656
Nama    : rizal
Umur    : 22
IPK     : 2
-----------------------------------------------------
Data Keseluruhan Mahasiswa :
Nim     : 2323
Nama    : ivan
Umur    : 22
IPK     : 2.0
----------------------
Nim     : 5656
Nama    : rizal
Umur    : 22
IPK     : 2.0
----------------------
```

```
-----------------------------------------------
Pencarian Data :
Masukkan Nama Mahasiswa yang dicari :
Nama    : ivan
Menggunakan Binary Search
Data 'ivan' ditemukan pada indeks :
0
0
Detail Mahasiswa dengan Nama 'ivan':
Nim     : 2323
Umur    : 22
IPK     : 2.0
----------------------
Nim     : 2323
Umur    : 22
IPK     : 2.0
----------------------
```

Program ini menggunakan Binary Search untuk mencari nama Mahasiswa dan mengurutkan.