

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA



2022

Praktikan

2141720183

RIDWAN CAESAR RIZQI KARISMA BIWARNI

TI 1C



Daftar Isi [harus diupdate setiap ada perubahan]

SEARCHING / PENCARIAN MENGGUNAKAN AGORITMA SEQUENTIAL SEARCH.....	4
LANGKAH 1.....	4
LANGKAH 2.....	4
LANGKAH 3.....	4
LANGKAH 4.....	5
LANGKAH 5.....	5
LANGKAH 6.....	5
LANGKAH 7.....	5
LANGKAH 8.....	6
LANGKAH 9.....	6
LANGKAH 10.....	6
LANGKAH 11.....	6
LANGKAH 12.....	7
LANGKAH 13.....	7
LANGKAH 14.....	7
LANGKAH 15.....	8
LANGKAH 16.....	8
LANGKAH 17.....	8
LANGKAH 18.....	8
VERIFIKASI HASIL PERCOBAAN	8
<i>Pertanyaan.....</i>	9
<i>Jawaban.....</i>	9
SEARCHING / PENCARIAN MENGGUNAKAN BINARY SEARCH	11
LANGKAH 1.....	11
LANGKAH 2.....	11
LANGKAH 3.....	11
VERIFIKASI HASIL PERCOBAAN	12
<i>Pertanyaan.....</i>	12
<i>Jawaban.....</i>	13
PERCOBAAN PENGAYAAN DIVIDE AND CONQUER.....	15
LANGKAH 1.....	15
LANGKAH 2.....	15
LANGKAH 3.....	15



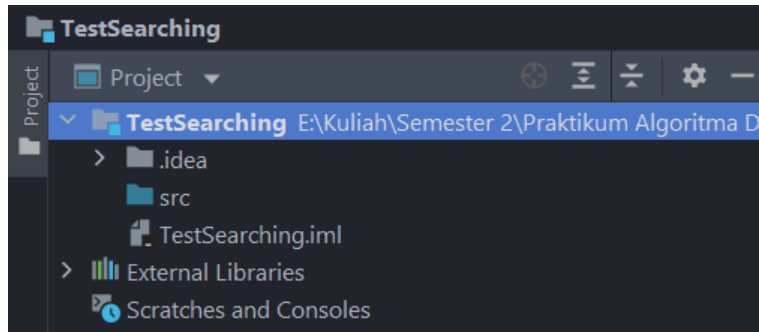
PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

LANGKAH 4.....	15
LANGKAH 5.....	15
LANGKAH 6.....	16
LANGKAH 7.....	16
LANGKAH 8.....	16
LANGKAH 9.....	17
LANGKAH 10.....	17
VERIFIKASI HASIL PERCOBAAN	17
LATIHAN PRAKTIKUM	18
SOAL NOMOR 1	18
JAWABAN NOMOR 1	18
<i>Modifikasi source code class PencarianMhs</i>	<i>18</i>
<i>Modifikasi source code class MahasiswaMain</i>	<i>18</i>
<i>Output.....</i>	<i>19</i>
SOAL NOMOR 2	19
JAWABAN NOMOR 2	20
<i>Modifikasi source code class PencarianMhs</i>	<i>20</i>
<i>Modifikasi source code class MahasiswaMain</i>	<i>20</i>
<i>Output.....</i>	<i>21</i>
SOAL NOMOR 3	21
JAWABAN NOMOR 3	21
<i>Source code.....</i>	<i>21</i>
<i>Output.....</i>	<i>23</i>

Searching / Pencarian Menggunakan Algoritma Sequential Search

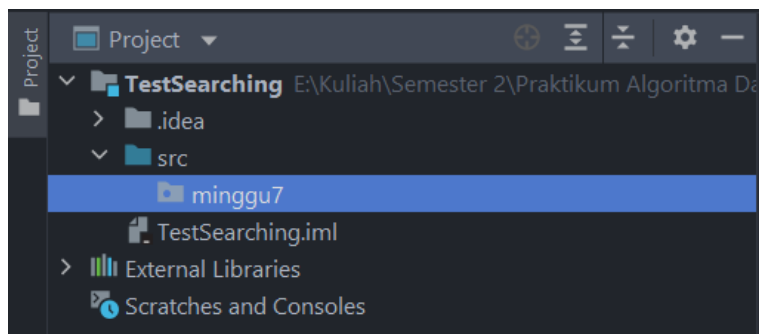
Langkah 1

Buatlah Project baru pada Netbeans dengan nama TestSearching



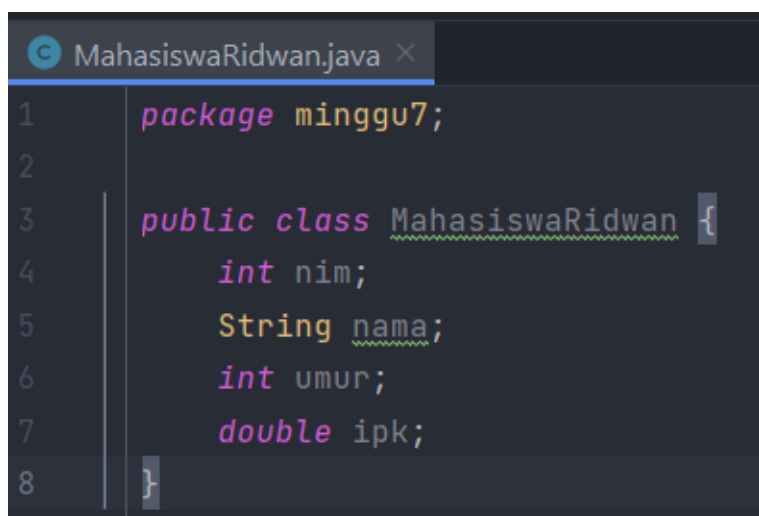
Langkah 2

Kemudian buat packages baru dengan nama minggu7.



Langkah 3

Buat class Mahasiswa, kemudian deklarasikan atribut berikut ini:



Langkah 4

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

```
9      MahasiswaRidwan(int ni, String n, int u, double i) {
10          nim = ni;
11          nama = n;
12          umur = u;
13          ipk = i;
14      }
```

Langkah 5

Buatlah method tampil bertipe void.

```
16      void tampil() {
17          System.out.println("Nim = " + nim);
18          System.out.println("Nama = " + nama);
19          System.out.println("Umur = " + umur);
20          System.out.println("IPK = " + ipk);
21      }
```

Langkah 6

Buat class baru dengan nama PencarianMhs seperti di bawah ini!

```
3      public class PencarianMhsRidwan {
4          MahasiswaRidwan[] listMhs = new MahasiswaRidwan[5];
5          int idx;
6      }
```

Langkah 7

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

```
7      void tambah(MahasiswaRidwan m) {
8          if (idx < listMhs.length) {
9              listMhs[idx] = m;
10             idx++;
11         } else {
12             System.out.println("Data sudah penuh !! ");
13         }
14     }
```

Langkah 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.

```
16 void tampil() {
17     for(MahasiswaRidwan m : listMhs) {
18         m.tampil();
19         System.out.println("-----");
20     }
21 }
```

Langkah 9

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

```
23 public int FindSeqSearch(int cari) {
24     int posisi = -1;
25     for (int j = 0; j < listMhs.length; j++) {
26         if (listMhs[j].nim == cari) {
27             posisi = j;
28             break;
29         }
30     }
31     return posisi;
32 }
```

Langkah 10

Buatlah method Tampilposisi bertipe void dan Deklarasikan isi dari method Tampilposisi

```
34 void Tampilposisi(int x, int pos) {
35     if (pos != -1) {
36         System.out.println("data : " + x + " ditemukan pada indeks " + pos);
37     } else {
38         System.out.println("data : " + x + " tidak ditemukan");
39     }
40 }
```

Langkah 11

Buatlah method TampilData bertipe void dan Deklarasikan isi dari method TampilData.

```

42 public void TampilData(int x, int pos) {
43     if (pos != -1) {
44         System.out.println("Nim\t : " + x);
45         System.out.println("Nama\t : " + listMhs[pos].nama);
46         System.out.println("Umur\t : " + listMhs[pos].umur);
47         System.out.println("IPK\t : " + listMhs[pos].ipk);
48     } else {
49         System.out.println("data " + x + " tidak ditemukan");
50     }
51 }

```

Langkah 12

Buatlah class baru dengan nama MahasiswaMain tambahkan method main seperti pada gambar berikut!

```

PencarianMhsRidwan.java x MahasiswaMain.java x MahasiswaRidwan.java
1 package minggu7;
2
3 public class MahasiswaMain {
4     public static void main(String[] args) {
5
6     }
7 }

```

Langkah 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.

```

6 public static void main(String[] args) {
7     Scanner s = new Scanner(System.in);
8     Scanner sl = new Scanner(System.in);
9
10    PencarianMhsRidwan data = new PencarianMhsRidwan();
11    int jumMhs = 5;
12
13    System.out.println("-----");
14    System.out.println("Masukkan data mahasiswa secara Urut dari Nim Terkecil");
15    for (int i = 0; i < jumMhs; i++) {
16        System.out.println("-----");
17        System.out.print("Nim\t : ");
18        int nim = s.nextInt();
19        System.out.print("Nama\t : ");
20        String nama = sl.nextLine();
21        System.out.print("Umur\t : ");
22        int umur = s.nextInt();
23        System.out.print("IPK\t : ");
24        double ipk = s.nextDouble();
25
26        MahasiswaRidwan m = new MahasiswaRidwan(nim, nama, umur, ipk);
27        data.tambah(m);
28    }
29 }

```

Langkah 14

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

```
30      System.out.println("-----");
31      System.out.println("Data keseluruhan mahasiswa : ");
32      data.tampil();
```

Langkah 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.

```
34      System.out.println("-----");
35      System.out.println("-----");
36      System.out.println("Pencarian Data : ");
37      System.out.println("Masukkan Nim mahasiswa yang dicari : ");
38      System.out.print("NIM : ");
39      int cari = s.nextInt();
40      System.out.println("Menggunakan sequential Search");
41      int posisi = data.FindSeqSearch(cari);
```

Langkah 16

Lakukan pemanggilan method Tampilposisi dari class PencarianMhs.

```
42
43      data.TampilPosisi(cari, posisi);|
```

Langkah 17

Lakukan pemanggilan method TampilData dari class PencarianMhs.

```
44      data.TampilData(cari, posisi);|
```

Langkah 18

Jalankan dan amati hasilnya.

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

Process finished with exit code 0

```

Pertanyaan

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

```

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

```

3. 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. Untuk method tampilData jika keyword ditemukan akan menampilkan isi dari data yang dicari sedangkan untuk method tampilPosisi akan menampilkan posisi indeks dari keyword yang dicari.

2. Fungsi break pada program tersebut adalah jika nilai listMhs[j].nim sama dengan nilai keyword yang dicari maka akan keluar dari perulangan.
3. Masih bisa berjalan karena ini menggunakan algoritme pencarian sequential yang tidak mempedulikan urutan data.

Searching / Pencarian Menggunakan Binary Search

Langkah 1

Pada percobaan 6.2.1 (sequential search) tambahkan method FindBinarySearch bertipe integer pada class PencarianMhs. Kemudian Deklarasikan isi method FindBinarySearch dengan algoritma pencarian data menggunakan teknik binary searching.

```
34 public int FindBinarySearch(int cari, int left, int right) {  
35     int mid;  
36     if (right >= left) {  
37         mid = (left + right)/2;  
38         if (cari == listMhs[mid].nim) {  
39             return mid;  
40         } else if (listMhs[mid].nim > cari) {  
41             return FindBinarySearch(cari, left, right: mid-1);  
42         } else {  
43             return FindBinarySearch(cari, left: mid+1, right);  
44         }  
45     }  
46     return -1;  
47 }
```

Langkah 2

Panggil method FindBinarySearch terdapat pada class PencarianMhs di kelas Mahasiswamain. Kemudian panggil method tampilposisi dan tampilData

```
46 System.out.println("=====");  
47 System.out.println("menggunakan binary search");  
48 posisi = data.FindBinarySearch(cari, left: 0, right: jumMhs-1);  
49 data.TampilPosisi(cari, posisi);  
50 data.TampilData(cari, posisi);
```

Langkah 3

Jalankan dan amati hasilnya.

Verifikasi Hasil Percobaan

```
-----
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
=====
menggunakan binary search
data      : 2018 ditemukan pada indeks 1
Nim       : 2018
Nama      : Sinta Sanjaya
Umur      : 22
IPK       : 4.0

Process finished with exit code 0
```

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. Berikut adalah kode program yang melakukan proses divide

```

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

```

2. Berikut adalah kode program yang melakukan proses conquer

```

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

```

3. Tidak bisa karena jika menggunakan algoritma binary search data harus di urutkan terlebih dahulu
4. Hasilnya tidak sesuai maka kita harus mengubah sedikit algoritma nya dengan mengubah tanda lebih dari menjadi kurang dari maka hasilnya akan sesuai.

```

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

```

5. Pada class PencarianMhs

```

3 public class PencarianMhsRidwan {
4     MahasiswaRidwan[] listMhs;
5     int idx;
6
7     PencarianMhsRidwan(int jumlahMhs) {
8         this.listMhs = new MahasiswaRidwan[jumlahMhs];
9     }

```

Pada class MahasiswaMain

```
5 public class MahasiswaMain {  
6     public static void main(String[] args) {  
7         Scanner s = new Scanner(System.in);  
8         Scanner sl = new Scanner(System.in);  
9  
10        System.out.print("Masukkan jumlah mahasiswa : ");  
11        int jumMhs = s.nextInt();  
12        PencarianMhsRidwan data = new PencarianMhsRidwan(jumMhs);  
13    }
```

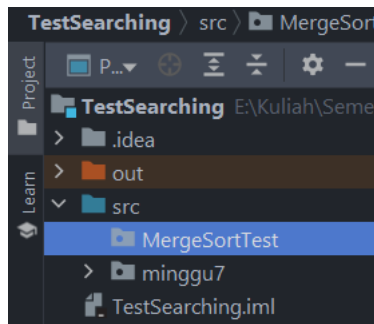
Output

```
Masukkan jumlah mahasiswa : 2  
-----  
Masukkan data mahasiswa secara Urut dari Nim Terkecil  
-----  
Nim      : 12  
Nama     : ridwan  
Umur     : 19  
IPK      : 4  
-----  
Nim      : 13  
Nama     : rizqi  
Umur     : 19  
IPK      : 4  
-----  
Data keseluruhan mahasiswa :  
Nim = 12  
Nama = ridwan  
Umur = 19  
IPK = 4.0  
-----  
Nim = 13  
Nama = rizqi  
Umur = 19  
IPK = 4.0  
-----  
-----  
Pencarian Data :  
Masukkan Nim mahasiswa yang dicari :  
NIM : 13  
Menggunakan sequential Search
```

Percobaan Pengayaan Divide and Conquer

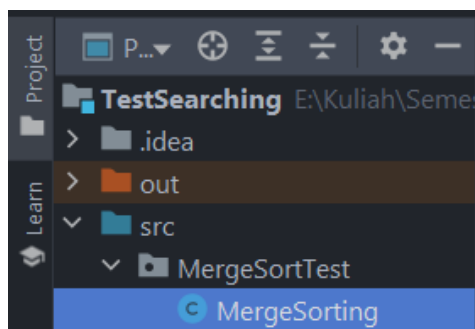
Langkah 1

Buatlah Package baru pada NetBeans dengan nama MergeSortTest



Langkah 2

Tambahkan class MergeSorting pada package tersebut



Langkah 3

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

```
3 public class MergeSorting {  
4     public void mergeSort(int[] data) {  
5  
6     }  
7 }
```

Langkah 4

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

```
8 private void merge(int[] data, int left, int middle, int right) {  
9  
10 }
```

Langkah 5

Implementasikan proses merge sebagai berikut.

```
8 @ private void merge(int[] data, int left, int middle, int right) {
9     int[] temp = new int[data.length];
10    for (int i = left; i <= right; i++) {
11        temp[i] = data[i];
12    }
13    int a = left;
14    int b = middle + 1;
15    int c = left;
16
17    // membandingkan setiap bagian
18    while (a <= middle && b <= right) {
19        if (temp[a] <= temp[b]) {
20            data[c] = temp[a];
21            a++;
22        } else {
23            data[c] = temp[b];
24            b++;
25        }
26        c++;
27    }
28    int s = middle - a;
29    for (int i = 0; i <= s; i++) {
30        data[c + i] = temp[a + i];
31    }
32 }
```

Langkah 6

Buatlah method sort

```
34 private void sort(int[] data, int left, int right) {
35
36 }
```

Langkah 7

Implementasikan kode berikut pada method sort

```
34 private void sort(int[] data, int left, int right) {
35     //membagi menjadi 2 bagian dan dibagi kembali hingga tidak dapat dibagi
36     if (left < right) {
37         int middle = (left + right) / 2;
38         sort(data, left, middle);
39         sort(data, left: middle + 1, right);
40         merge(data, left, middle, right);
41     }
42 }
```

Langkah 8

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

```
4 public void mergeSort(int[] data) {
5     sort(data, left: 0, right: data.length - 1);
6 }
```


Langkah 9

Tambahkan method printArray

```
44 @ public void printArray(int[] arr) {  
45     int n = arr.length;  
46     for (int i = 0; i < n; i++) {  
47         System.out.print(arr[i] + " ");  
48     }  
49     System.out.println();  
50 }
```

Langkah 10

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

```
3 public class SortMain {  
4     public static void main(String[] args) {  
5         int[] data = {10, 40, 30, 50, 70, 20, 100, 90};  
6         System.out.println("sorting dengan merge sort");  
7         MergeSorting mSort = new MergeSorting();  
8         System.out.println("Data awal");  
9         mSort.printArray(data);  
10        mSort.mergeSort(data);  
11        System.out.println("Setelah diurutkan");  
12        mSort.printArray(data);  
13    }  
14 }
```

Verifikasi Hasil Percobaan

```
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2  
sorting dengan merge sort  
Data awal  
10 40 30 50 70 20 100 90  
Setelah diurutkan  
10 20 30 40 50 70 90 100  
  
Process finished with exit code 0
```

Latihan Praktikum

Soal nomor 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)

Jawaban nomor 1

Modifikasi source code class PencarianMhs

```
37 public void urutkanData() {
38     MahasiswaRidwan temp;
39     for (int i = 0; i < listMhs.length - 1; i++) {
40         for (int j = 0; j < listMhs.length - 1 - i; j++) {
41             if (listMhs[j].nim > listMhs[j + 1].nim) {
42                 temp = listMhs[j];
43                 listMhs[j] = listMhs[j + 1];
44                 listMhs[j + 1] = temp;
45             }
46         }
47     }
48 }
49
50 public int FindBinarySearch(int cari, int left, int right) {
51     int mid;
52     if (right >= left) {
53         mid = (left + right)/2;
54         if (cari == listMhs[mid].nim) {
55             return mid;
56         } else if (listMhs[mid].nim > cari) {
57             return FindBinarySearch(cari, left, right: mid-1);
58         } else {
59             return FindBinarySearch(cari, left: mid+1, right);
60         }
61     }
62     return -1;
63 }
```

Modifikasi source code class MahasiswaMain

```
47 System.out.println("=====");
48 System.out.println("menggunakan binary search");
49 data.urutkanData();
50 posisi = data.FindBinarySearch(cari, left: 0, right: jumMhs-1);
51 data.TampilPosisi(cari, posisi);
52 data.TampilData(cari, posisi);
```

Output

```
Masukkan jumlah mahasiswa : 4
-----
Masukkan data mahasiswa secara Urut dari Nim Terkecil
-----
Nim      : 12
Nama     : ridwan
Umur     : 19
IPK      : 4
-----
Nim      : 15
Nama     : caesar
Umur     : 19
IPK      : 4
-----
Nim      : 13
Nama     : rizqi
Umur     : 19
IPK      : 3.9
-----
Nim      : 19
Nama     : karisma
Umur     : 19
IPK      : 3.9
-----

Data keseluruhan mahasiswa :
Nim = 12
Nama = ridwan
Umur = 19
IPK = 4.0
-----
Nim = 15
Nama = caesar
Umur = 19
IPK = 4.0
-----
Nim = 13
Nama = rizqi
Umur = 19
IPK = 3.9
-----
Nim = 19
Nama = karisma
Umur = 19
IPK = 3.9
-----

-----
Pencarian Data :
Masukkan Nim mahasiswa yang dicari :
NIM : 12
Menggunakan sequential Search
data      : 12 ditemukan pada indeks 0
Nim       : 12
Nama      : ridwan
Umur      : 19
IPK       : 4.0
=====
menggunakan binary search
data      : 12 ditemukan pada indeks 0
Nim       : 12
Nama      : ridwan
Umur      : 19
IPK       : 4.0

Process finished with exit code 0
```

Soal nomor 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!

Jawaban nomor 2

Modifikasi source code class PencarianMhs

```

26     public int FindSeqSearch(String cari) {
27         int posisi = -1;
28         for (int j = 0; j < listMhs.length; j++) {
29             if (listMhs[j].nama.equalsIgnoreCase(cari)) {
30                 posisi = j;
31                 break;
32             }
33         }
34         return posisi;
35     }

```

```

64     // modifikasi bagian parameter
65     public void TampilPosisi(String x, int pos) {
66         if (pos != -1) {
67             System.out.println("data\t : " + x + " ditemukan pada indeks " + pos);
68         } else {
69             System.out.println("data\t : " + x + " tidak ditemukan");
70         }
71     }
72     // modifikasi bagian parameter dan variable
73     public void TampilData(String x, int pos) {
74         if (pos != -1) {
75             System.out.println("Nim\t\t : " + listMhs[pos].nim);
76             System.out.println("Nama\t : " + x);
77             System.out.println("Umur\t : " + listMhs[pos].umur);
78             System.out.println("IPK\t\t : " + listMhs[pos].ipk);
79         } else {
80             System.out.println("data " + x + " tidak ditemukan");
81         }
82     }

```

Modifikasi source code class MahasiswaMain

```

35     System.out.println("-----");
36     System.out.println("-----");
37     System.out.println("Pencarian Data : ");
38     System.out.println("Masukkan Nama mahasiswa yang dicari : ");
39     System.out.print("NAMA : ");
40     String cari = sl.nextLine();
41     System.out.println("Menggunakan sequential Search");
42     int posisi = data.FindSeqSearch(cari);
43
44     data.TampilPosisi(cari, posisi);
45     data.TampilData(cari, posisi);
46
47     //
48     //     System.out.println("=====");
49     //     System.out.println("menggunakan binary search");
50     //     data.urutkanData();
51     //     posisi = data.FindBinarySearch(cari, 0, jumMhs-1);
52     //     data.TampilPosisi(cari, posisi);
53     //     data.TampilData(cari, posisi);

```

Output

```
Masukkan jumlah mahasiswa : 3
-----
Masukkan data mahasiswa secara Urut dari Nim Terkecil
-----
Nim      : 12
Nama     : ridwan
Umur     : 19
IPK      : 4
-----
Nim      : 13
Nama     : rizqi
Umur     : 20
IPK      : 3.9
-----
Nim      : 14
Nama     : ridwan
Umur     : 20
IPK      : 3.9
-----

Data keseluruhan mahasiswa :
Nim = 12
Nama = ridwan
Umur = 19
IPK = 4.0
-----
Nim = 13
Nama = rizqi
Umur = 20
IPK = 3.9
-----
Nim = 14
Nama = ridwan
Umur = 20
IPK = 3.9
-----

Pencarian Data :
Masukkan Nama mahasiswa yang dicari :
NAMA : ridwan
Menggunakan sequential Search
data      : ridwan ditemukan pada indeks 0
Nim       : 12
Nama      : ridwan
Umur      : 19
IPK       : 4.0

Process finished with exit code 0
```

Jika terdapat nama yang sama maka pada program diatas akan ditampilkan nama yang pertama kali ditemukan.

Soal nomor 3

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)

Jawaban nomor 3

Source code

```
SoalNomor3Ridwan.java
1 package soal;
2 import java.util.Arrays;
3 public class SoalNomor3Ridwan {
4     public static void main(String[] args) {
5         int[] arr = {12, 17, 2, 1, 70, 50, 90, 17, 2, 90};
6         System.out.println("Data sebelum diurutkan");
7         System.out.println(Arrays.toString(arr));
8         int temp;
9         int max = arr[0];
10        for (int i = 0; i < arr.length - 1; i++) {
11            if (max < arr[i]) {
12                max = arr[i];
13            }
14            for (int j = 0; j < arr.length - 1 - i; j++) {
15                if (arr[j] > arr[j + 1]) {
16                    temp = arr[j];
17                    arr[j] = arr[j + 1];
18                    arr[j + 1] = temp;
19                }
20            }
21        }
22        System.out.println("Data setelah diurutkan");
23        System.out.println(Arrays.toString(arr));
24        System.out.print("Nilai terbesar : " + max);
25        System.out.println();
26
27        int awal = 0;
28        int startindex = -1;
29        int count = 0;
30        int akhir = arr.length - 1;
31        int mid;
32        while (awal <= akhir) {
33            mid = (awal + akhir) / 2;
34            if (arr[mid] > max) {
35                akhir = mid - 1;
36            } else if (arr[mid] == max) {
37                startindex = mid;
38                akhir = mid - 1;
39            } else {
40                awal = mid + 1;
41            }
42        }
43        int endIndex = -1;
44        awal = 0;
45        akhir = arr.length - 1;
46        while (awal <= akhir) {
47            mid = (awal + akhir) / 2;
48            if (arr[mid] > max) {
49                akhir = mid - 1;
50            } else if (arr[mid] == max) {
51                endIndex = mid;
52                awal = mid + 1;
53                count++;
54            } else {
55                awal = mid + 1;
56            }
57        }
58        System.out.println("Ada " + count + " nilai terbesar");
59        System.out.print("Nilai tersebut ada pada indeks ke ");
60        if (startindex != 1 && endIndex != 1) {
61            for (int i = 0; i + startindex <= endIndex; i++) {
62                if (i > 0) {
63                    System.out.print(" dan ");
64                }
65                System.out.print(i + startindex);
66            }
67        }
68    }
69 }
70 }
```

Output

```
Run: SoalNomor3Ridwan x
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe"
Data sebelum diurutkan
[12, 17, 2, 1, 70, 50, 90, 17, 2, 90]
Data setelah diurutkan
[1, 2, 2, 12, 17, 17, 50, 70, 90, 90]
Nilai terbesar : 90
Ada 2 nilai terbesar
Nilai tersebut ada pada indeks ke 8 dan 9
Process finished with exit code 0
```