

Laporan Praktikum Alogaritma dan Struktur Data

Jobsheet 7 : Searching

Alogaritma dan Struktur Data

Dosen Pembimbing : Triana Fatmawati, S.T,M.T



Nama : Eka Putri Natalya Kabelen
Nim : 2341760107
Kelas : SIB 1E
Prodi : D-IV Sistem Informasi Bisnis

JURUSAN TEKNOLOGI INFORMASI

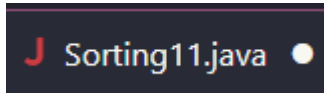
POLITEKNIK NEGERI MALANG

Tahun Ajaran 2024

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

```
Sorting11.java > ...
1  /**
2   * Sorting11
3   */
4  public class Sorting11 {
5
6      public static void sequentialSearch(int []arr, int key) {
7          for (int i = 0; i < arr.length; i++) {
8              if (i == key) {
9                  System.out.println("Data ditemukan pada index ke-" + i);
10             }
11         }
12         System.out.println(x:"Data tidak ditemukan");
13     }
14 }
```

3. Tambahkan fungsi main sebagai berikut


```
14     }
15     Run | Debug
16     public static void main(String[] args) {
17         int[] daftarNilai = {10, 5, 20, 15, 80, 45};
18         sequentialSearch(daftarNilai, key:5);
19     }
20 }
```

4. Hasil Run dari kode program

```
redhat.java\jdt_ws\JOBSHEET 7_8c06f48f\bin' 'Sorting11'
Data ditemukan pada index ke-5
Data tidak ditemukan
PS C:\Users\ASUS\Documents\SEMESTER 2\Alogaritma&StrukturData\JOBSHEET 7>
```

❖ Langkah-langkah Percobaan Sequential Search

1. Buat class Mahasiswa, kemudian deklarasikan atribut berikut ini:

 Mahasiswa11.java

```
J Mahasiswa11.java > ...  
1  public class Mahasiswa11 {  
2  
3      String nama;  
4      int nim, umur;  
5      double ipk;
```

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

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

3. Buatlah method tampil bertipe void.

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

4. Buat class baru dengan nama PencarianMhs seperti di bawah ini!

```
J PencarianMhs11.java > PencarianMhs11 > tampil()  
1  public class PencarianMhs11 {  
2  
3      Mahasiswa11 listMhs[] = new Mahasiswa11[5];  
4      int idx;
```

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

```
6      void tambah(Mahasiswa11 m) {
7          if(idx < listMhs.length) {
8              listMhs[idx] = m;
9              idx ++;
10         } else {
11             System.out.println(x:"Data sudah penuh!!");
12         }
13     }
```

6. 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.

```
15     void tampil() {
16         for(Mahasiswa11 m : listMhs) {
17             m.tampil();
18             System.out.println(x:"=====");
19         }
20     }
```

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

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

8. Buatlah method Tampilpoisisi bertipe void dan Deklarasikan isi dari method Tampilpoisisi.

```

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

```

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

```

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

```

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

```

MahasiswaMain11.java > MahasiswaMain11 >
1     import java.util.Scanner;
2     public class MahasiswaMain11 {

```

11. 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.

```

4     public static void main(String[] args) {
5         Scanner s = new Scanner(System.in);
6         Scanner sl = new Scanner(System.in);
7
8         PencarianMhs11 data = new PencarianMhs11 ();
9         int jumMhs = 5;
10
11         System.out.println(x:"=====");
12         System.out.println(x:"Masukkan data mahasiswa secaraurut dari NIM terkecil : ");
13         for (int i = 0; i < jumMhs; i++) {
14             System.out.println(x:"=====");
15             System.out.print(s:"Nim\t : ");
16             int nim = s.nextInt();
17             System.out.print(s:"Nama\t : ");
18             String nama = sl.nextLine();
19             System.out.print(s:"Umur\t : ");
20             int umur = s.nextInt();
21             System.out.print(s:"IPK\t : ");
22             double ipk = s.nextDouble();
23
24             Mahasiswa11 m = new Mahasiswa11 (nama, nim, umur, ipk);
25             data.tambah(m);

```

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

```
27 System.out.println(x:"=====");
28 System.out.println(x:"Data Keseluruhan Mahasiswa : ");
29 data.tampil();
```

13. 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.

Lakukan pemanggilan method Tampilposisi dari class PencarianMhs.

Lakukan pemanggilan method TampilData dari class PencarianMhs.

```
31 System.out.println(x:"=====");
32 System.out.println(x:"=====");
33 System.out.println(x:"Pencarian data : ");
34 System.out.println(x:"Masukkan NIM Mahasiswa yang dicari: ");
35 System.out.print(s:"NIM : ");
36 int cari = s.nextInt();
37 System.out.println(x:"Menggunakan sequential Search");
38 int posisi = data.FindSeqSearch(cari);
39 data.TampilPosisi(cari, posisi);
40 data.TampilData(cari, posisi);
41 }
42 }
43 }
```

14. Hasil Run kode program

```
=====
Masukkan data mahasiswa secara urut dari NIM terkecil :
=====
Nim      : 2017
Nama     : Dewi L
Umur     : 23
IPK      : 3.5
=====
Nim      : 2018
Nama     : Sinta S
Umur     : 22
IPK      : 4
=====
Nim      : 2019
Nama     : Danang A
Umur     : 22
IPK      : 3.7
=====
Nim      : 2020
Nama     : Budi P
Umur     : 20
IPK      : 2.9
=====
Nim      : 2021
Nama     : Vania S
Umur     : 20
IPK      : 3.0
=====
```

```

=====
Data Keseluruhan Mahasiswa :
Nim = 2017
Nama = Dewi L
Umur = 23
IPK = 3.5
=====
Nim = 2018
Nama = Sinta S
Umur = 22
IPK = 4.0
=====
Nim = 2019
Nama = Danang A
Umur = 22
IPK = 3.7
=====
Nim = 2020
Nama = Budi P
Umur = 20
IPK = 2.9
=====
Nim = 2021
Nama = Vania S
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 S
Umur     : 22
IPK      : 4.0
PS C:\Users\ASUS\Documents\SEMESTER 2\Alogaritma&StrukturData\JOBSHEET 7>

```

6.2.3. Pertanyaan

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

```

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

```

Jawab :

```

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

```

```

JUC_WS\JOBSHEET 7_80061481\BIN
Data ditemukan pada index ke-2

```

2. Jelaskan perbedaan metod TampilData dan Tampilposisi pada class PencarianMhs

Jawab :

Perbedaannya yaitu jika Tampil data digunakan untuk menampilkan data mahasiswa sedangkan ,Tampil proses digunakan untuk menampilkan posisi index yang dicari.

3. Jelaskan fungsi break pada kode program dibawah ini! Algoritma dan Struktur Data 2023-2024 Tim Ajar Algoritma dan Struktur Data 2023-2024 Jurusan Teknologi Informasi-Politeknik Negeri Malang .

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

Jawab :

Fungsi dari break adalah untuk menghentikan eksekusi dari loop (baik itu for, while, atau do-while) secara paksa ketika suatu kondisi tertentu terpenuhi, tanpa harus menunggu loop mencapai kondisi terminasi normalnya. Pada array tersebut dengan atribut nim samadengan data yang di cari. Jika data sama dengan posisi berisi j maka pencarian akan berhenti.

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

Jawab :

Masih bisa berjalan, tetapi efisiensi pencarian dapat terpengaruh jika data tidak terurut. Untuk mencari hasil yang benar dalam pencarian sequential search, data biasanya harus terurut..Hasil yang dikeluarkan benar tetapi tidak berurutan mulai dari yang terkecil hingga sampai yang terbesar.

6.3. Searching / Pencarian Menggunakan Binary Search

6.3.1. Langkah-langkah Percobaan Binary Search menggunakan Array

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

```
int[] sortedNilai = {5, 5, 10, 20, 30, 40, 50 };
int index = binarySearchAsc (sortedNilai, key:5);

if(index != -1) {
    System.out.println("Data ditemukan pada indeks ke-" + index);
} else {
    System.out.println("Data tidak ditemukan");
}
```

3. Hasil Run dari kode program

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

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.

```
52     public int FindBinarySearch (int cari, int left, int right) {
53         int mid;
54         if (right >= left ) {
55             mid = (left + right) / 2;
56             if (cari == listMhs[mid].nim) {
57                 return (mid);
58             } else if (listMhs[mid].nim > cari) {
59                 return FindBinarySearch(cari, left, mid - 1 );
60             } else {
61                 return FindBinarySearch(cari, mid + 1, right);
62             }
63         }
64         return -1;
65     }
66 }
```

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

```
42     System.out.println(x:"=====");
43     System.out.println(x:"menggunakan binary Search");
44     posisi = data.FindBinarySearch(cari, left:0, jumMhs - 1);
45     data.TampilPosisi(cari, posisi);
46     data.TampilData(cari,posisi);
47
48 }
```

3. Hasil Run dari kode program

```
=====
Masukkan data mahasiswa secara urut dari NIM terkecil :
=====
Nim      : 2017
Nama     : Dewi L
Umur     : 23
IPK      : 3.5
=====
Nim      : 2018
Nama     : Sinta s
Umur     : 22
IPK      : 4
=====
Nim      : 2019
Nama     : Danang a
Umur     : 22
IPK      : 3.7
=====
Nim      : 2020
Nama     : Budi p
Umur     : 20
IPK      : 3.7
=====
Nim      : 2021
Nama     : Vania s
Umur     : 20
IPK      : 3.0
=====
```

```
=====
Data Keseluruhan Mahasiswa :
```

```
Nim = 2017
```

```
Nama = Dewi L
```

```
Umur = 23
```

```
IPK = 3.5
```

```
=====
Nim = 2018
```

```
Nama = Sinta s
```

```
Umur = 22
```

```
IPK = 4.0
```

```
=====
Nim = 2019
```

```
Nama = Danang a
```

```
Umur = 22
```

```
IPK = 3.7
```

```
=====
Nim = 2020
```

```
Nama = Budi p
```

```
Umur = 20
```

```
IPK = 3.7
```

```
=====
Nim = 2021
```

```
Nama = Vania s
```

```
Umur = 20
```

```
IPK = 3.0
```

```
=====
Pencarian data :
```

```
Masukkan NIM Mahasiswa yang dicari:
```

```
NIM : 2018
```

```
Menggunakan sequential Search
```

```
Data : 2018ditemukan pada indeks 1
```

```
Nim      : 2018
```

```
Nama     : Sinta s
```

```
Umur     : 22
```

```
IPK      : 4.0
```

```
=====
menggunakan binary Search
```

```
Data : 2018ditemukan pada indeks 1
```

```
Nim      : 2018
```

```
Nama     : Sinta s
```

```
Umur     : 22
```

```
IPK      : 4.0
```

6.3.3. Pertanyaan

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

Jawab :

Berikut adalah kode program yang melakukan proses divide

```
53         int mid;  
54         if (right >= left ) {  
55             mid = (left + right) / 2;
```

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

Jawab :

Berikut adalah kode program yang melakukan proses conquer

```
54         if (right >= left ) {  
55             mid = (left + right) / 2;  
56             if (cari == listMhs[mid].nim) {  
57                 return (mid);  
58             } else if (listMhs[mid].nim > cari) {  
59                 return FindBinarySearch(cari, left, mid - 1 );  
60             } else {  
61                 return FindBinarySearch(cari, mid + 1, right);
```

3. Jika data Nim yang dimasukkan tidak urut. Apakah program masih dapat berjalan?

Mengapa demikian!

Jawab :

Masih dapat berjalan. Tetapi output program tidak berurutan mulai dari yang terkecil hingga yang terbesar, dan inputan yang kita cari tidak ditemukan

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 :

```
52     public int FindBinarySearch (int cari, int left, int right) {  
53         int mid;  
54         if (right >= left ) {  
55             mid = (left + right) / 2;  
56             if (listMhs[mid].nim == cari) {  
57                 return (mid);  
58             } else if (listMhs[mid].nim < cari) {  
59                 return FindBinarySearch(cari, left, mid - 1 );  
60             } else {  
61                 return FindBinarySearch(cari, mid + 1, right);  
62             }  
63         }  
64         return -1;  
65     }
```

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

Jawab :

```

1  /**
2   * PencarianMhs11
3   */
4  public class PencarianMhs11 {
5
6      Mahasiswa11 listMhs[];
7      int idx;
8
9      public PencarianMhs11(int maxSize) {
10         listMhs = new Mahasiswa11[maxSize];
11     }
12     void tambah(Mahasiswa11 m) {
13         if(idx < listMhs.length) {
14             listMhs[idx] = m;
15             idx++;
16         } else {
17             System.out.println("Data sudah penuh!!");
18         }
19     }
20
21     void tampil() {
22         for (int i = 0; i < idx; i++) {
23             listMhs[i].tampil();
24             System.out.println("-----");
25         }
26     }
27
28     public int FindSeqSearch (int cari) {
29         int posisi = -1;
30         for (int j = 0; j < idx; j++) {
31             if (listMhs[j].nim == cari) {
32                 posisi = j;
33                 break;
34             }
35         }
36         return posisi;
37     }
38     public int FindBinarySearch(int cari, int Left, int right) {
39         int mid;
40         if (right >= Left) {
41             mid = (left + right) / 2;
42             if (cari == listMhs[mid].nim) {
43                 return (mid);
44             }else if (listMhs[mid].nim < cari) {
45                 return FindBinarySearch(cari, Left, mid - 1);
46             }else{
47                 return FindBinarySearch(cari, mid + 1, right);
48             }
49         }
50         return -1;
51     }
52     public void TampilPosisi (int x, int pos) {
53         if (pos!= -1) {
54             System.out.println("data : " + x + "\nditemukan pada indeks " + (
55 pos+1));
56         }else {
57             System.out.println("data " + x + "tidak ditemukan");
58         }
59     }
60
61     public void TampilData (int x, int pos) {
62         if (pos!= -1) {
63             System.out.println("Nim\t : " + x);
64             System.out.println("Nama\t : " + listMhs[pos].nama);
65             System.out.println("Umur\t : " + listMhs[pos].umur);
66             System.out.println("IPK\t : " + listMhs[pos].ipk);
67         } else {
68             System.out.println("data " + x + "tidak ditemukan ");
69         }
70     }
71 }
72

```

```

1  import java.util.Scanner;
2
3  /**
4   * MahasiswaMain11
5   */
6  public class MahasiswaMain11 {
7
8      public static void main(String[] args) {
9          Scanner s = new Scanner(System.in);
10         Scanner s1 = new Scanner(System.in);
11
12         int jumMhs = 5;
13
14         System.out.println("Masukkan jumlah Mahasiswa: ");
15         int maxSize = s.nextInt();
16
17         PencarianMhs11 data = new PencarianMhs11(maxSize);
18         System.out.println("-----");
19         System.out.println("Masukkan data mahasiswa secaraurut dari NIM terkecil : ");
20         for (int i = 0; i < maxSize; i++) {
21             System.out.println("-----");
22             System.out.print("Nim\t : ");
23             int nim = s.nextInt();
24             System.out.print("Nama\t : ");
25             String nama = s1.nextLine();
26             System.out.print("Umur\t : ");
27             int umur = s.nextInt();
28             System.out.print("IPK\t : ");
29             double ipk = s.nextDouble();
30
31             Mahasiswa11 m = new Mahasiswa11(nama, nim, umur, ipk);
32             data.tambah(m);
33         }
34         System.out.println("-----");
35         System.out.println("Data Keseluruhan Mahasiswa : ");
36         data.tampil();
37
38         System.out.println("-----");
39         System.out.println("Pencarian data : ");
40         System.out.println("Masukkan NIM Mahasiswa yang dicari: ");
41         System.out.print("NIM : ");
42         int cari = s.nextInt();
43         System.out.println("Menggunakan sequential Search");
44         int posisi = data.FindSeqSearch(cari);
45         data.TampilPosisi(cari, posisi);
46         data.TampilData(cari, posisi);
47
48         System.out.println("=====");
49         System.out.println("Menggunakan binary Search");
50         posisi = data.FindBinarySearch(cari, 0, maxSize - 1);
51         data.TampilPosisi(cari, posisi);
52         data.TampilData(cari, posisi);
53     }
54 }
55

```

Hasil run dari kode program modif diatas

```
Masukkan jumlah Mahasiswa:
2
-----
Masukkan data mahasiswa secara urut dari NIM terkecil :
-----
Nim      : 2018
Nama     : putri
Umur     : 18
IPK      : 3.8
-----
Nim      : 2019
Nama     : valen
Umur     : 20
IPK      : 4.0
-----
```

```
-----
Data Keseluruhan Mahasiswa :
Nim = 2018
Nama = putri
Umur = 18
IPK = 3.8
-----
Nim = 2019
Nama = valen
Umur = 20
IPK = 4.0
-----
-----
Pencarian data :
Masukkan NIM Mahasiswa yang dicari:
NIM : 2019
Menggunakan sequential Search
data : 2019
ditemukan pada indeks 2
Nim      : 2019
Nama     : valen
Umur     : 20
IPK      : 4.0
=====
Menggunakan binary Search
data 2019tidak ditemukan
data 2019tidak ditemukan
```

6.4. Percobaan Pengayaan Divide and Conquer

6.4.1. Langkah-langkah Percobaan Merge Sort

1. Tambahkan class MergeSorting pada package tersebut ,
Pada class MergeSorting buatlah method mergeSort yang menerima parameter data array yang akan diurutkan,
Buatlah method merge untuk melakukan proses penggabungan data dari bagian kiri dan kanan,

```
J MergeSorting11.java > MergeSorting11 > merge(int[], int, int, int)
1  public class MergeSorting11 {
2
3      public void mergeSort(int[] data) {
4          sort(data, left:0, data.length - 1);
5      }
```

2. Implementasikan proses merge sebagai berikut.

```
6
7      public void merge (int data[], int left, int middle, int right) {
8          int[] temp = new int[data.length];
9          for (int i = left; i <= right; i++) {
10             temp[i] = data[i];
11          }
12          int a = left;
13          int b = middle + 1;
14          int c = left;
15
16          while (a <= middle && b <= right) {
17              if (temp[a] <= temp[b]) {
18                  data[c] = temp[a];
19                  a++;
20              } else {
21                  data[c] = temp[b];
22                  b++;
23              }
24              c++;
25          }
26
27          while (a <= middle) {
28              data[c] = temp[a];
29              a++;
30              c++;
31          }
32      }
```

3. Buatlah method sort lalu Implementasikan kode berikut pada method sort

```
34      public void sort (int data[], int left, int right) {
35          if (left < right) {
36              int middle = (left + right) / 2;
37              sort(data, left, middle);
38              sort(data, middle + 1, right);
39              merge(data, left, middle, right);
40          }
41      }
```


4. Pada method `mergeSort`, panggil method `sort` dengan parameter data yang ingin diurutkan serta range data awal sampai dengan akhir. Tambahkan method `printArray` sekaligus.

```
42  
43     public void printArray(int arr[]){  
44         int n = arr.length;  
45         for (int i = 0; i < n; i++) {  
46             System.out.println(arr[i] + " ");  
47         }  
48         System.out.println();  
49     }  
50 }  
51
```

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

```
J SortMain11.java > ...  
1  public class SortMain11 {  
2  
3      Run | Debug  
4      public static void main(String[] args) {  
5          int data[] = {10,40,30,50,70,20,100,90};  
6          System.out.println(x:"Sorting dengan merge sort");  
7          MergeSorting11 mSort = new MergeSorting11();  
8          System.out.println(x:"-----");  
9          System.out.println(x:"Data awal");  
10         System.out.println(x:"-----");  
11         mSort.printArray(data);  
12         mSort.mergeSort(data);  
13         System.out.println(x:"-----");  
14         System.out.println(x:"Setelah di urutkan");  
15         System.out.println(x:"-----");  
16         mSort.printArray(data);  
17     }  
18 }
```

6. Hasil Run dari kode program

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

5.5.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!

```
1  /**
2   * Mahasiswa11
3   */
4  public class Mahasiswa11 {
5
6      String nama;
7      int nim, umur;
8      double ipk;
9
10     Mahasiswa11 (String n, int ni, int u, double i) {
11         nama = n;
12         nim = ni;
13         umur = u;
14         ipk = i;
15     }
16
17     void tampil() {
18         System.out.println("Nim = " + nim);
19         System.out.println("Nama = " + nama);
20         System.out.println("Umur = " + umur );
21         System.out.println("IPK = " + ipk);
22         System.out.println("=====");
23     }
24
25     public int compareTo(String cari) {
26         throw new UnsupportedOperationException("Unimplemented method 'compareTo'");
27     }
28 }
29
```

```

1  /**
2   * PencarianMhs11
3   */
4  public class PencarianMhs11 {
5
6      Mahasiswa11 listMhs[] = new Mahasiswa11 [5];
7      int idx;
8
9      void tambah(Mahasiswa11 m) {
10         if(idx < listMhs.length) {
11             listMhs[idx] = m;
12             idx++;
13         } else {
14             System.out.println("Data sudah penuh!!");
15         }
16     }
17
18     void tampil() {
19         for(Mahasiswa11 m : listMhs) {
20             if (m != null) {
21                 m.tampil();
22                 System.out.println("=====");
23             }
24         }
25     }
26
27     public int FindSeqSearch (int cari) {
28         int posisi = -1;
29         for (int j = 0; j < listMhs.length; j++) {
30             if (listMhs[j].nim == cari) {
31                 posisi = j;
32                 break;
33             }
34         }
35         return posisi;
36     }
37
38     public void TampilPosisi (int x, int pos) {
39         if (pos != -1) {
40             System.out.println("Data : " + x + "ditemukan pada indeks " + pos);
41         } else {
42             System.out.println("Data " + x + "tidak ditemukan");
43         }
44     }
45
46     public void TampilData (int x, int pos) {
47         if (pos != -1) {
48             System.out.println("Nim\t : " + x);
49             System.out.println("Nama\t : " + listMhs[pos].nama);
50             System.out.println("Umur\t : " + listMhs[pos].umur);
51             System.out.println("IPK\t : " + listMhs[pos].ipk);
52         } else {
53             System.out.println("data " + x + "tidak ditemukan ");
54         }
55     }
56
57     public int FindBinarySearch (String cari, int Left, int right) {
58         if (right >= Left) {
59             int mid = Left + (right - Left) / 2;
60
61             if (listMhs[mid].nama.equalsIgnoreCase(cari)) {
62                 return mid;
63             }
64
65             if (listMhs[mid].nama.compareToIgnoreCase(cari) > 0) {
66                 return FindBinarySearch(cari, Left, mid - 1);
67             }
68
69             return FindBinarySearch(cari, mid + 1, right);
70         }
71
72         return -1;
73     }
74
75     void sortDataByNama() {
76         for (int i = 0; i < idx - 1; i++) {
77             for (int j = 0; j < idx - i - 1; j++) {
78                 if (listMhs[j].nama.compareToIgnoreCase(listMhs[j + 1].nama) > 0) {
79                     Mahasiswa11 temp = listMhs[j];
80                     listMhs[j] = listMhs[j + 1];
81                     listMhs[j + 1] = temp;
82                 }
83             }
84         }
85     }
86 }

```

```

1  import java.util.Scanner;
2
3  public class MahasiswaMain11 {
4      public static void main(String[] args) {
5          Scanner s = new Scanner(System.in);
6          Scanner sl = new Scanner(System.in);
7
8          PencarianMhs11 data = new PencarianMhs11();
9          int jumMhs = 5;
10
11         int[] daftarNilai = {10, 5, 20, 15, 45};
12
13         System.out.println("=====");
14         System.out.println("Data Nilai : ");
15         for (int i = 0; i < daftarNilai.length; i++) {
16             System.out.println("Nilai " + (i+1) + " : " + daftarNilai[i]);
17         }
18
19         System.out.println("=====");
20         System.out.println("Masukkan data mahasiswa secaraurut dari NIM terkecil : ");
21         for (int i = 0; i < jumMhs; i++) {
22             System.out.println("=====");
23             System.out.print("Nim\t : ");
24             int nim = s.nextInt();
25             System.out.print("Nama\t : ");
26             String nama = sl.nextLine();
27             System.out.print("Umur\t : ");
28             int umur = s.nextInt();
29             System.out.print("IPK\t : ");
30             double ipk = s.nextDouble();
31
32             Mahasiswa11 m = new Mahasiswa11 (nama, nim, umur, ipk);
33             data.tambah(m);
34         }
35
36         System.out.println("=====");
37         System.out.println("Data Keseluruhan Mahasiswa : ");
38         data.tampil();
39
40         System.out.println("=====");
41         System.out.println("=====");
42         System.out.println("Pencarian data : ");
43         System.out.println("Masukkan Nama Mahasiswa yang dicari : ");
44         System.out.print("Nama : ");
45         String cariNama = sl.nextLine();
46
47         System.out.println("Menggunakan binary Search");
48         int posisi = data.FindBinarySearch (cariNama, 0, jumMhs - 1);
49
50         if (posisi != -1) {
51             // Cek apakah ada hasil lebih dari satu
52             int posisiAkhir = posisi;
53             int posisiAwal = posisi;
54             while (posisiAwal > 0 && data.listMhs[posisiAwal - 1].nama.equalsIgnoreCase(cariNama)) {
55                 posisiAwal--;
56             }
57             while (posisiAkhir < jumMhs - 1 && data.listMhs[posisiAkhir + 1].nama.equalsIgnoreCase(cariNama)) {
58                 posisiAkhir++;
59             }
60             if (posisiAwal != posisiAkhir) {
61                 System.out.println("Data ditemukan pada indeks " + posisiAwal + " hingga " + posisiAkhir);
62                 System.out.println("Hasil pencarian lebih dari satu.");
63             } else {
64                 System.out.println("Data ditemukan pada indeks " + posisi);
65                 data.TampilData(data.listMhs[posisi].nim, posisi);
66             }
67         } else {
68             System.out.println("Data tidak ditemukan");
69         }
70     }
71     public static int sequentialSearch(int[] arr, int key) {
72         for (int i = 0; i < arr.length; i++) {
73             if (arr[i] == key) {
74                 return i;
75             }
76         }
77         return - 1;
78     }
79 }

```

Hasil Run modif dari kode program diatas

```
=====
Data Nilai :
Nilai 1 : 10
Nilai 2 : 5
Nilai 3 : 20
Nilai 4 : 15
Nilai 5 : 45
=====
Masukkan data mahasiswa secara urut dari NIM terkecil :
=====
Nim      : 2019
Nama     : putri
Umur     : 18
IPK      : 3.8
=====
Nim      : 2020
Nama     : valen
Umur     : 20
IPK      : 4.0
=====
Nim      : 2021
Nama     : vian
Umur     : 21
IPK      : 3.5
=====
Nim      : 2022
Nama     : rosa
Umur     : 19
IPK      : 3.0
=====
Nim      : 2023
Nama     : lenka
Umur     : 20
IPK      : 2.7
=====
```

```
=====
Data Keseluruhan Mahasiswa :
Nim = 2019
Nama = putri
Umur = 18
IPK = 3.8
=====
Nim = 2020
Nama = valen
Umur = 20
IPK = 4.0
=====
Nim = 2021
Nama = vian
Umur = 21
IPK = 3.5
=====
Nim = 2022
Nama = rosa
Umur = 19
IPK = 3.0
=====
Nim = 2023
Nama = lenka
Umur = 20
IPK = 2.7
=====
```

```
=====
Pencarian data :
Masukkan Nama Mahasiswa yang dicari:
Nama : vian
Menggunakan binary Search
Data ditemukan pada indeks 2
Nim      : 2021
Nama     : vian
Umur     : 21
IPK      : 3.5
PS C:\Users\ASUS\Documents\SEMESTER 2\Alogaritma
```