

# ALGORITMA STRUKTUR DATA

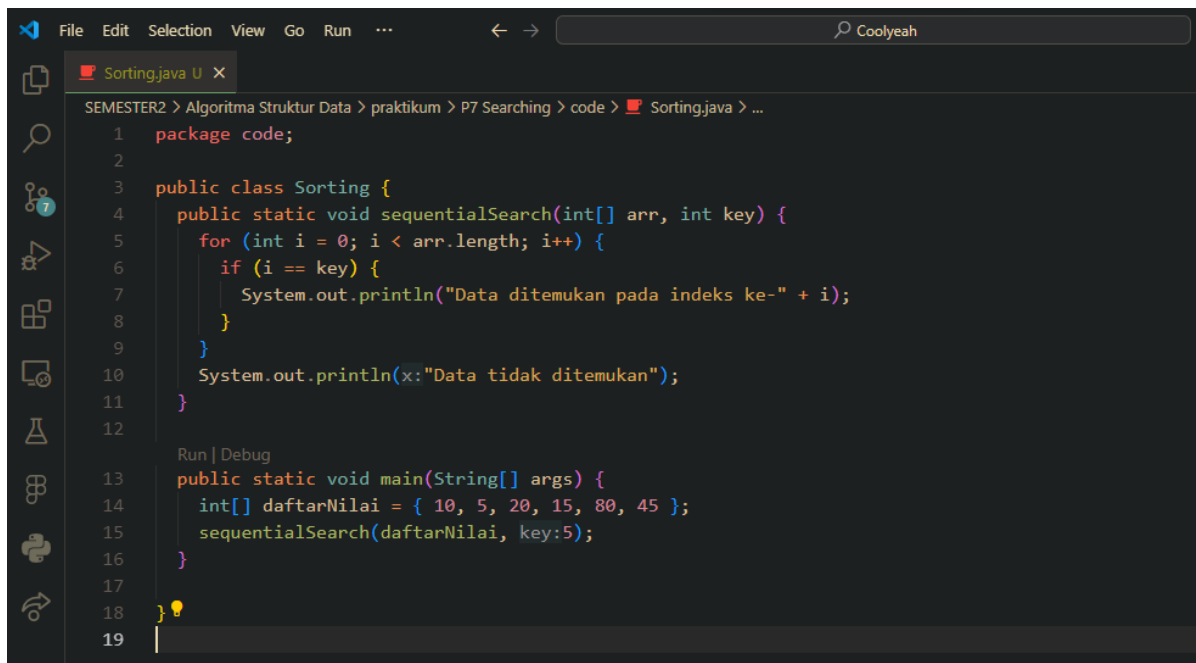
## Praktikum – Searching

Lavina 2341760062

### Praktikum 1: Searching / Pencarian Menggunakan Algoritma Sequential Search

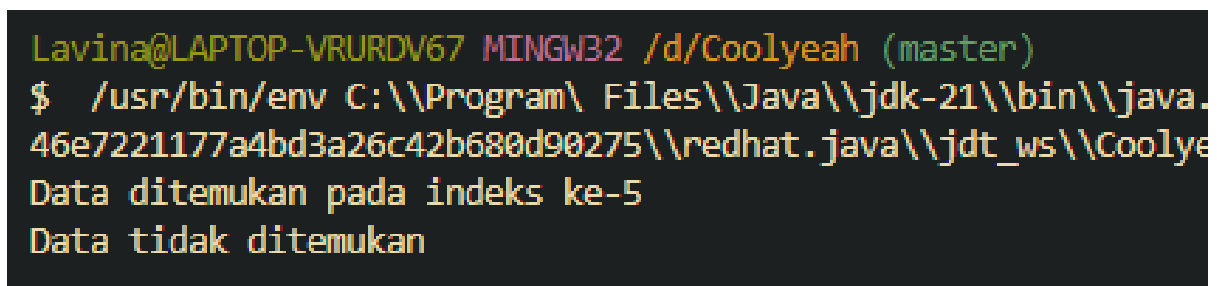
#### Sequential Search Menggunakan Array

Sorting.java



```
1 package code;
2
3 public class Sorting {
4     public static void sequentialSearch(int[] arr, int key) {
5         for (int i = 0; i < arr.length; i++) {
6             if (i == key) {
7                 System.out.println("Data ditemukan pada indeks ke-" + i);
8             }
9         }
10        System.out.println(x:"Data tidak ditemukan");
11    }
12
13    public static void main(String[] args) {
14        int[] daftarNilai = { 10, 5, 20, 15, 80, 45 };
15        sequentialSearch(daftarNilai, key:5);
16    }
17
18 }
19
```

Output:



```
Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyeh (master)
$ /usr/bin/env C:\\Program\\ Files\\Java\\jdk-21\\bin\\java.
46e7221177a4bd3a26c42b680d90275\\redhat.java\\jdt_ws\\Coolyeh
Data ditemukan pada indeks ke-5
Data tidak ditemukan
```

## Sequential Search Menggunakan Array of Object

### Mahasiswa.java

```
File Edit Selection View Go Run ... TestSearching
Mahasiswa.java x PencarianMhs.java MahasiswaMain.java 1
src > minggu7 > Mahasiswa.java > Mahasiswa
1 package minggu7;
2
3 /**
4  * Mahasiswa
5  */
6 public class Mahasiswa {
7
8     int nim, umur;
9     String nama;
10    double ipk;
11
12    Mahasiswa(int ni, String n, int u, double i) {
13        nim = ni;
14        nama = n;
15        umur = u;
16        ipk = i;
17    }
18
19    void tampil() {
20        System.out.printf(format:"NIM = %d\nNama = %s\nUmur = %d\nIPK = %.2f\n", nim, nama, umur, ipk);
21    }
22 }
```

### PencarianMhs.java

```
1 package minggu7;
2
3 public class PencarianMhs {
4     Mahasiswa listMhs[] = new Mahasiswa[5];
5     int idx;
6
7     void tambah(Mahasiswa m) {
8         if (idx < listMhs.length) {
9             listMhs[idx] = m;
10            idx++;
11        } else {
12            System.out.println("Data sudah penuh !");
13        }
14    }
15
16    void tampil() {
17        for (Mahasiswa m : listMhs) {
18            m.tampil();
19            System.out.println("-----");
20        }
21    }
22
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    }
}
```

```

34 public 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 }
41
42 public void TampilData(int x, int pos) {
43     if (pos != -1) {
44         System.out.printf("NIM\t : %d\nNama\t : %s\nUmur\t : %d\nIPK\t : %.2f\n", x, listMhs[pos].nama, listMhs[pos].umur,
45             listMhs[pos].ipk);
46     } else {
47         System.out.println("data " + x + " tidak ditemukan");
48     }
49 }
50 }
51

```

## MahasiswaMain.java

```

1 package minggu7;
2
3 public class PencarianMhs {
4     Mahasiswa listMhs[] = new Mahasiswa[5];
5     int idx;
6
7     void tambah(Mahasiswa m) {
8         if (idx < listMhs.length) {
9             listMhs[idx] = m;
10            idx++;
11        } else {
12            System.out.println("Data sudah penuh !");
13        }
14    }
15
16    void tampil() {
17        for (Mahasiswa m : listMhs) {
18            m.tampil();
19            System.out.println("-----");
20        }
21    }
22
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    }
33
34    public 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    }
41
42    public void TampilData(int x, int pos) {
43        if (pos != -1) {
44            System.out.printf("NIM\t : %d\nNama\t : %s\nUmur\t : %d\nIPK\t : %.2f\n", x, listMhs[pos].nama, listMhs[pos].umur,
45                listMhs[pos].ipk);
46        } else {
47            System.out.println("data " + x + " tidak ditemukan");
48        }
49    }
50 }
51

```

Output:

```
-----  
Masukkan data mahasiswa secara urut  
-----  
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  
-----
```

```
=====
```

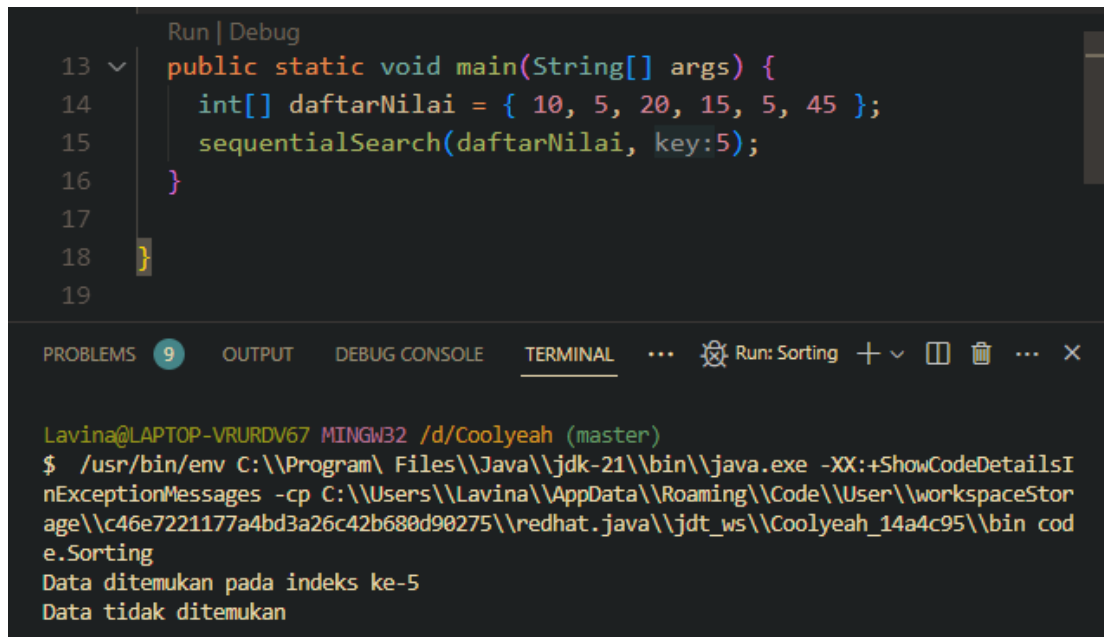
Data Keseluruhan Mahasiswa			
NIM	=	2017	
Nama	=	Dewi Lestari	
Umur	=	23	
IPK	=	3.50	
-----			
NIM	=	2018	
Nama	=	Sinta Sanjaya	
Umur	=	22	
IPK	=	4.00	
-----			
NIM	=	2019	
Nama	=	Danang Adi	
Umur	=	22	
IPK	=	3.70	
-----			
Nama	=	Budi Prakarsa	
Umur	=	20	
IPK	=	2.90	
-----			
NIM	=	2021	
Nama	=	Vania Siti	
Umur	=	20	
IPK	=	3.00	
-----			

```
-----  
-----  
Pencarian Data  
Masukkan NIM Mahasiswa yang dicari  
NIM : 2018  
Menggunakan sequential search  
data : 2018ditemukan pada indeks 1  
NIM      : 2018  
Nama     : Sinta Sanjaya  
Umur     : 22  
IPK      : 4.00
```

## 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);
}
```



The screenshot shows an IDE with a code editor and a terminal. The code editor displays the following code:

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

The terminal shows the following output:

```
Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyeah (master)
$ /usr/bin/env C:\Program Files\Java\jdk-21\bin\java.exe -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\Lavina\AppData\Roaming\Code\User\workspaceStorage\c46e7221177a4bd3a26c42b680d90275\redhat.java\jdt_ws\Coolyeah_14a4c95\bin code.Sorting
Data ditemukan pada indeks ke-5
Data tidak ditemukan
```

2. Jelaskan perbedaan metod TampilData dan Tampilposisi pada class PencarianMhs !

**Jawab :** Method TampilData berfungsi untuk menampilkan detail dari data yang dicari sedangkan method TampilPosisi berfungsi untuk menampilkan indeks dimana data yang dicari ditemukan.

3. Jelaskan fungsi break pada kode program dibawah ini!

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

**Jawab :** Fungsinya adalah untuk mengakhiri iterasi ketika 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!

**Jawab :** Karena proses pencarian datanya dilakukan secara urut dari dari indeks yang pertama hingga yang terakhir, jika data sudah ditemukan maka iterasi akan berakhir, jika belum maka iterasi akan terus berlanjut, inilah kenapa algoritma sequential search tidak efektif untuk data yang besar, tetapi jika hanya untuk kumpulan data yang sederhana seperti 5 mahasiswa, algoritma ini sudah sesuai untuk digunakan.

## Percobaan 2: Searching / Pencarian Menggunakan Binary Search

### Binary Search menggunakan Array

Sorting.java

```
public static int binarySearch(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;
}
```

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

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

Output :

```
Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyyeah (master)
$ cd d:\\Coolyyeah ; /usr/bin/env C:\\Program\\ Files\\
orkspaceStorage\\c46e7221177a4bd3a26c42b680d90275\\re
Data ditemukan pada indeks ke-1
```

PencarianMhs.java

```
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;
}
```

MahasiswaMain.java

```
System.out.println(x:"=====");
System.out.println(x:"Menggunakan binary search");
posisi = data.FindBinarySearch(cari, left:0, jmlMhs - 1);
data.TampilPosisi(cari, posisi);
data.TampilData(cari, posisi);
```

Output :

```
-----  
Masukkan data mahasiswa secara urut  
-----  
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	=	3.50	
-----			
NIM	=	2018	
Nama	=	Sinta Sanjaya	
Umur	=	22	
IPK	=	4.00	
-----			
NIM	=	2019	
Nama	=	Danang Adi	
Umur	=	22	
IPK	=	3.70	
-----			
Nama	=	Budi Prakarsa	
Umur	=	20	
IPK	=	2.90	
-----			
NIM	=	2021	
Nama	=	Vania Siti	
Umur	=	20	
IPK	=	3.00	
-----			

```
-----  
-----  
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      : 4.00  
=====
```

Menggunakan binary search	
data	: 2018 ditemukan pada indeks 1
NIM	: 2018
Nama	: Sinta Sanjaya
Umur	: 22
IPK	: 4.00



## Pertanyaan

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

**Jawab :**

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

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

**Jawab :**

```
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);  
}
```

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

**Jawab :** Tidak bisa, karena binary search mengandalkan pembagian array menjadi dua bagian secara berulang untuk mempersempit ruang pencarian. Pada data yang tidak urut, pembagian ini tidak akan menghasilkan subarray yang berisi elemen yang lebih kecil atau lebih besar, sehingga algoritma akan gagal menemukan elemen yang dicari.

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 !

**Jawab :** Hasilnya tidak sesuai karena kode yang sebelumnya untuk mencari data yang ascending, berikut perbaikan untuk mencari data yang descending :

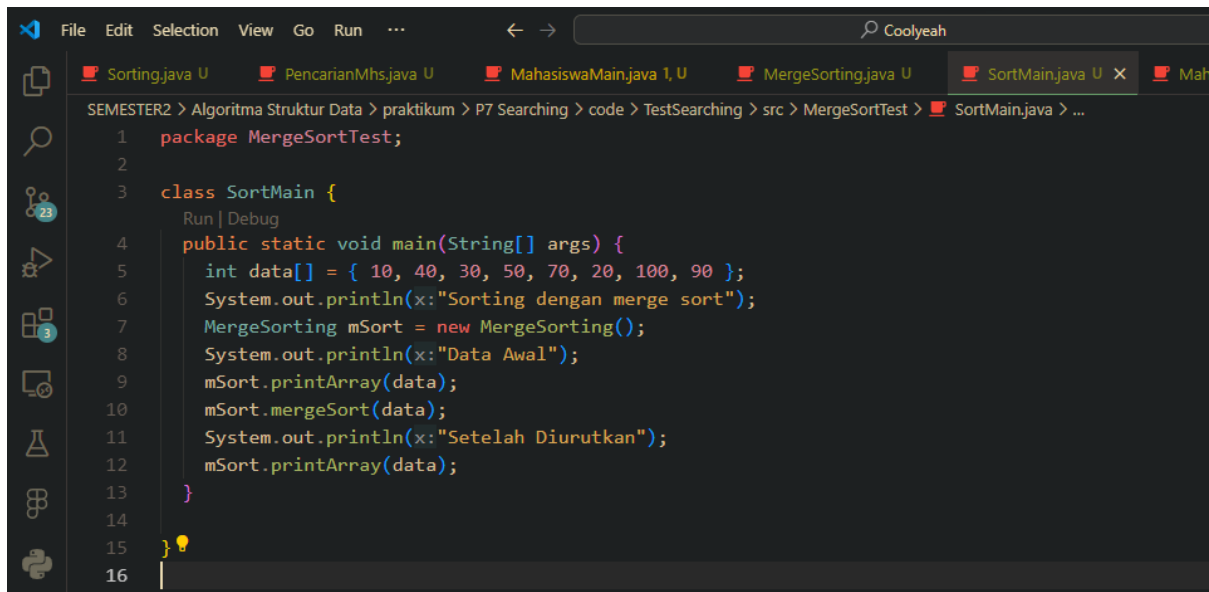
```
} else if (cari > listMhs[mid].nim) {  
    return FindBinarySearch(cari, left, mid - 1);
```

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

```
System.out.print(s:"Masukkan jumlah mahasiswa : ");  
int jmlMhs = sc.nextInt();  
sc.nextLine();
```

### Percobaan 3: Pengayaan Divide and Conquer

#### SortMain.java



```
1 package MergeSortTest;
2
3 class SortMain {
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         MergeSorting mSort = new MergeSorting();
8         System.out.println(x:"Data Awal");
9         mSort.printArray(data);
10        mSort.mergeSort(data);
11        System.out.println(x:"Setelah Diurutkan");
12        mSort.printArray(data);
13    }
14
15 }
16
```

#### MergeSorting.java



```
1 package MergeSortTest;
2
3 public class MergeSorting {
4     public void mergeSort(int[] data) {
5         sort(data, 0, 0);
6     }
7
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
14        int a = left;
15        int b = middle + 1;
16        int c = left;
17
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    }
29 }
```

```

27     }
28     int s = middle - a;
29     for (int i = 0; i <= s; i++) {
30         data[c + i] = temp[a + i];
31     }
32 }
33
34 private 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 }
42
43 public void printArray(int arr[]) {
44     int n = arr.length;
45     for (int i = 0; i < n; i++) {
46         System.out.print(arr[i] + " ");
47     }
48     System.out.println();
49 }
50
51 }

```

Output :

```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Lavina@LAPTOP-VRURDV67 MINGW32 /d/Coolyeah (master)
$ C:\Program Files\Java\jdk-21\bin\java.exe -XX:+ShowCodeDetailsInExceptionMessages -cp C:\d3a26c42b680d90275\redhat.java\jdt_ws\Coolyeah_14a4c95\bin MergeSortTest.SortMain
Sorting dengan merge sort
Data Awal
10 40 30 50 70 20 100 90
Setelah Diurutkan
10 40 30 50 70 20 100 90

```

## Latihan Praktikum

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!

MahasiswaMain.java

```
1  package minggu7;
2
3  import java.util.Scanner;
4
5  public class MahasiswaMain {
6      public static void main(String[] args) {
7          Scanner sc = new Scanner(System.in);
8
9          System.out.print("Masukkan jumlah mahasiswa : ");
10         int jmlMhs = sc.nextInt();
11         PencarianMhs data = new PencarianMhs(jmlMhs);
12         sc.nextLine();
13
14         System.out.println("-----");
15         System.out.println("Masukkan data mahasiswa secara urut");
16         for (int i = 0; i < data.jmlMhs; i++) {
17             System.out.println("-----");
18             System.out.print("NIM\t : ");
19             int nim = sc.nextInt();
20             sc.nextLine();
21             System.out.print("Nama\t : ");
22             String nama = sc.nextLine();
23             System.out.print("Umur\t : ");
24             int umur = sc.nextInt();
25             sc.nextLine();
26             System.out.print("IPK\t : ");
27             double ipk = sc.nextDouble();
28             sc.nextLine();
29
30             Mahasiswa m = new Mahasiswa(nim, nama, umur, ipk);
31             data.tambah(m);
32         }
33
34         System.out.println("=====");
35         System.out.println("Data Keseluruhan Mahasiswa");
36         data.tampil();
37
38         System.out.println("_____");
39         System.out.println("_____");
40         System.out.println("Pencarian Data");
41         System.out.println("Masukkan Nama Mahasiswa yang dicari");
42         System.out.print("Nama : ");
43         String cari = sc.nextLine();
```

```

50     System.out.println("\nMenggunakan binary search");
51     int posisi = data.FindBinarySearch(cari, 0, data.jmlMhs - 1);
52     data.TampilPosisi(cari, posisi);
53     data.TampilData(cari, posisi);
54
55 }
56 }

```

PencarianMhs.java

```

1  package minggu7;
2
3  public class PencarianMhs {
4      int jmlMhs;
5      Mahasiswa listMhs[];
6      int idx = 0;
7
8      PencarianMhs(int jmlMhs) {
9          this.jmlMhs = jmlMhs;
10         listMhs = new Mahasiswa[jmlMhs];
11     }
12
13     void tambah(Mahasiswa m) {
14         if (idx >= listMhs.length) {
15             System.out.println("Data sudah penuh !");
16             return;
17         }
18         int i = idx;
19         while (i > 0 && listMhs[i - 1].nama.compareTo(m.nama) > 0)
20             i--;
21
22         for (int j = idx; j > i; j--) {
23             listMhs[j] = listMhs[j - 1];
24         }
25
26         listMhs[i] = m;
27         idx++;
28     }

```

## Method FindBinarySearch

```
1 public int FindBinarySearch(String cari, int left, int right) {
2     int mid;
3     if (right >= left) {
4         mid = (left + right) / 2;
5         if (cari.compareTo(listMhs[mid].nama) == 0) {
6             if ((mid > 0 && cari.equals(listMhs[mid - 1].nama))
7                 ||
8                 (mid < listMhs.length - 1 && cari.equals(listMhs[mid + 1].nama)))
9             System.err.println("Nama yang di cari lebih dari 1!!!");
10            return (mid);
11        } else if (cari.compareTo(listMhs[mid].nama) < 0) {
12            return FindBinarySearch(cari, left, mid - 1);
13        } else {
14            return FindBinarySearch(cari, mid + 1, right);
15        }
16    }
17    return -1;
18 }
```

Output :

```
Masukkan jumlah mahasiswa : 3
-----
Masukkan data mahasiswa secaraurut
-----
NIM      : 2022
Nama     : Lavina
Umur     : 19
IPK      : 4
-----
NIM      : 2019
Nama     : Yonanda
Umur     : 19
IPK      : 4
-----
NIM      : 2017
Nama     : Yonanda
Umur     : 20
IPK      : 4
=====
Data Keseluruhan Mahasiswa
NIM = 2022
Nama = Lavina
Umur = 19
IPK = 4.00
-----
NIM = 2019
Nama = Yonanda
Umur = 19
IPK = 4.00
-----
```

```
-----
NIM = 2017
Nama = Yonanda
Umur = 20
IPK = 4.00
-----
-----
Pencarian Data
Masukkan Nama Mahasiswa yang dicari
Nama : Yonanda

Menggunakan binary search
Nama yang di cari lebih dari 1!!!
Data      : Yonanda ditemukan pada indeks 1

NIM      : 2019
Nama     : Yonanda
Umur     : 19
IPK      : 4.00
```