

Langkah-langkah 6.2.1

1. Membuat folder baru dengan nama Praktikum06. Buat file dengan nama Sorting.java

```
1 public class Sorting17{
```

2. Menambahkan method sequentialSearch() yang melakukan pencarian data bertipe integer di dalam array of integer

```
2 public static void sequentialSearch(int[] arr, int key) {  
3     for (int i = 0; i < arr.length; i++){  
4         if (i == key) {  
5             System.out.println("Data ditemukan dalam indeks ke-" + i);  
6         }  
7     }  
8  
9     System.out.println(x:"Data tidak ditemukan");  
10 }  
11
```

3. Menambahkan fungsi main

```
12 public static void main(String[] args) {  
13     int[] daftarNilai = { 10, 5, 20, 15, 80, 45 };  
14     sequentialSearch(daftarNilai, key:5);  
15 }  
16
```

4. Hasil

```
Data ditemukan dalam indeks ke-5  
Data tidak ditemukan
```

Langkah-langkah 6.2.2

1. Membuat packages baru dengan nama minggu7. lalu membuat class Mahasiswa

```
▼ minggu7  
Mahasiswa.java
```

2. Mendeklarasikan atribut dibawah ini

```
3 public class Mahasiswa {  
4     int nim;  
5     String nama;  
6     int umur;  
7     double ipk;  
8 }
```

3. Membuat konstruktor dengan nama Mahasiswa dengan parameter (int ni, String n, int u, double i) kemudian isi konstruktor tersebut dengan kode berikut

```
9      Mahasiswa(int ni, String n, int u, double i){
10          nim = ni;
11          nama = n;
12          umur = u;
13          ipk = i;
14      }
15
```

4. Membuat 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      }
22
```

5. Membuat class baru dengan nama PencarianMhs seperti di bawah ini!

```
3      public class PencarianMhs {
4          Mahasiswa listMhs[] = new Mahasiswa[5];
5          int idx;
6
```

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

```
7      void tambah(Mahasiswa m) {
8          if(idx < listMhs.length) {
9              listMhs[idx] = m;
10             idx++;
11         } else {
12             System.out.println(x:"Data sudah penuh !!");
13         }
14     }
15
```

7. Menambahkan method tampil() di dalam class PencarianMhs! Method tampil() digunakan untuk menampilkan semua data mahasiswa-mahasiswa yang ada di dalam class tersebut!

```
16      void tampil() {
17          for(Mahasiswa m : listMhs) {
18              m.tampil();
19              System.out.println(x:"-----");
20          }
21      }
22
```

8. Menambahkan method FindSeqSearch bertipe integer dengan parameter cari bertipe integer

```
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
```

9. Membuat method Tampilposisi bertipe void dan Deklarasikan isi dari method Tampilposisi.

```
35 public void Tampilposisi(int x,int pos)
36 {
37     if (pos!= -1) {
38         System.out.println("data " + x + "ditemukan pada indeks " + pos);
39     } else {
40         System.out.println("data " + x + "tidak ditemukan");
41     }
42 }
43
```

10. Membuat method TampilData bertipe void dan Deklarasikan isi dari method TampilData.

```
44 public void TampilData(int x,int pos)
45 {
46     if(pos!= -1) {
47         System.out.println("Nim\t : " + x);
48         System.out.println("Nama\t : "+listMhs[pos].nama);
49         System.out.println("Umur\t : "+listMhs[pos].umur);
50         System.out.println("Ipk\t : "+listMhs[pos].ipk);
51     } else {
52         System.out.println("data " + x + "tidak ditemukan");
53     }
54 }
55 }
56
```

11. Membuat class baru dengan nama MahasiswaMain

 MahasiswaMain.java

12. Menambahkan method main

```
5 public class MahasiswaMain {
6     Run | Debug
7     public static void main(String[] args) {
```

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

```
7 Scanner s = new Scanner(System.in);
8 Scanner s1 = new Scanner(System.in);
9
10 PencarianMhs data = new PencarianMhs();
11 int jumMhs = 5;
12
13 System.out.println(x:"-----");
14 System.out.println(x:"Masukkan data mahasiswa secara Urut dari Nim Terkecil");
15 for(int i = 0; i < jumMhs; i++){
16     System.out.println(x:"-----");
17     System.out.print(s:"Nim\t: ");
18     int nim = s.nextInt();
19     System.out.print(s:"Nama\t: ");
20     String nama = s1.nextLine();
21     System.out.print(s:"Umur\t: ");
22     int umur = s.nextInt();
23     System.out.print(s:"IPK\t: ");
24     double ipk = s.nextDouble();
25
26     Mahasiswa m = new Mahasiswa(nim, nama, umur, ipk);
27     data.tambah(m);
28 }
```

14. Panggil method tampil()

```
29 System.out.println(x:"-----");
30 System.out.println(x:"Data keseluruhan Mahasiswa : ");
31 data.tampil();
```

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

adalah variable cari

```
32     System.out.println(x:"_____");
33     System.out.println(x:"_____");
34     System.out.println(x:"Pencarian Data: ");
35     System.out.println(x:"Masukkan Nim Mahasiswa yang dicari: ");
36     System.out.print (s:"NIM : ");
37     int cari = s.nextInt();
38     System.out.println(x:"menggunakan sequential Search");
39     int posisi = data. FindSeqSearch (cari);
```

16. Melakukan pemanggilan method Tampilposisi dari class PencarianMhs

```
41     data.Tampilposisi(cari, posisi);
```

17. Melakukan pemanggilan method TampilData dari class PencarianMhs.

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

18. Hasil

```
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 2018ditemukan pada indeks 1
Nim      : 2018
Nama     : Sinta Sanjaya
Umur     : 22
Ipk      : 4.0
```

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);
}
```

2. Jelaskan perbedaan metod TampilData dan Tampilposisi pada class PencarianMhs

3. Jelaskan fungsi break pada kode program dibawah ini!

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

4. 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. Data tidak ditemukan tampil karena pada method sequentialSearch terdapat print di bawah sendiri
2. **TampilData** memiliki fungsi untuk menampilkan data yang telah di temukan pada method search
- TampilPosisi** memiliki fungsi untuk menampilkan detail data yang ditemukan yaitu posisi index data pada array
3. Untuk menghentikan perulangan jika kondisi terpenuhi
4. Program masih dapat berjalan, hasilnya juga benar, hal ini karena fungsi searching tidak akan mencari data dari awal array hingga akhir

6.3.1. Langkah-langkah Percobaan Binary Search menggunakan Array

1. Menambahkan method `binarySearchAsc()` pada file `Sorting.java`

```
18 public static int binarySearchAsc (int[] arr, int key){
19     int start = 0, end = arr.length - 1;
20
21     while (start <= end) {
22         int mid = start + (end - start) / 2;
23
24         if (arr[mid] == key) {
25             return mid;
26         }
27         if (arr[mid] < key) {
28             start = mid + 1;
29         }
30         else {
31             end = mid - 1;
32         }
33     }
34
35     return -1;
36 }
```

2. Menambahkan baris program untuk menguji method `binarySearchAsc()` pada fungsi `main()`

```
16 int[] sortedNilai = { 5, 5, 10, 20, 30, 40, 50 };
17 int index = binarySearchAsc (sortedNilai, key:5);
18
19 if (index != -1) {
20     System.out.println("Data ditemukan pada indeks ke-" + index);
21 }
22 else {
23     System.out.println(x:"Data tidak ditemukan");
24 }
25 }
26 }
```

3. Hasil

```
Data ditemukan dalam indeks ke-5
Data tidak ditemukan
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.

```

57     public int FindBinarySearch (int cari, int left, int right){
58         int mid;
59         if (right >= left) {
60             mid = (left + right) / 2;
61             if (cari == listMhs[mid].nim) {
62                 return (mid);
63             } else if (listMhs [mid].nim > cari) {
64                 return FindBinarySearch (cari, left, mid - 1);
65             } else {
66                 return FindBinarySearch (cari, mid + 1, right);
67             }
68         }
69         return -1;
70     }
71 }
72
73

```

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

```

45     System.out.println(x:"=====");
46     System.out.println(x:"menggunakan binary Search");
47     posisi = data.FindBinarySearch (cari, left:0, jumMhs - 1);
48     data.Tampilposisi(cari, posisi);
49     data.TampilData(cari, posisi);
50 }
51
52 }
53

```

3. Hasil

Data keseluruhan Mahasiswa :	Masukkan data mahasiswa secara Urut dari Nim Terkecil
Nim = 2017 Nama = Dewi Lestari Umur = 23 Ipk = 3.5	Nim : 2017 Nama : Dewi Lestari Umur : 23 IPK : 3.5
Nim = 2018 Nama = Sinta Sanjaya Umur = 22 Ipk = 4.0	Nim : 2018 Nama : Sinta Sanjaya Umur : 22 IPK : 4
Nim = 2019 Nama = Danang Adi Umur = 22 Ipk = 3.7	Nim : 2019 Nama : Danang Adi Umur : 22 IPK : 3.7
Nim = 2020 Nama = Budi Prakarsa Umur = 20 Ipk = 2.9	Nim : 2020 Nama : Budi Prakarsa Umur : 20 IPK : 2.9
Nim = 2021 Nama = Vania Siti Umur = 20 Ipk = 3.0	Nim : 2021 Nama : Vania Siti 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 Sanjaya
Umur : 22
Ipk : 4.0
=====
menggunakan binary Search
data 2018ditemukan pada indeks 1
Nim : 2018
Nama : Sinta Sanjaya
Umur : 22
Ipk : 4.0
=====

```


6.3.3. Pertanyaan

1. Tunjukkan pada kode program yang mana proses divide dijalankan!
2. Tunjukkan pada kode program yang mana proses conquer dijalankan!
3. Jika data Nim yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian!
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. `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);  
}
```

- 2.
3. Program bisa di run meskipun data tidak urut, karena searching akan mencari nilai dari data urutan awal hingga akhir

```
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);  
}
```

4. Hasilnya tidak sesuai, dengan melakukan ubahan seperti dibawah ini , maka searching akan dapat dilakukan

```
public PencarianMhs(int jumMhs) {  
    listMhs = new Mahasiswa[jumMhs];  
}
```

- 5.

```
int jumMhs = 2;   
System.out.println("Masukkan berapa banyak data yang ingin dimasukkan : ");  
System.out.println("-----");  
PencarianMhs cari = new PencarianMhs(jumMhs);
```

Langkah-langkah 6.4.1

1. Membuat class MergeSorting dan membuat method merge

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


2. Membuat method merge()

```
7 private 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     int s = middle - a;  
27     for (int i = 0; i <= s; i++) {  
28         data[c + i] = temp[a + i];  
29     }  
30 }
```

3. Membuat method sort() dan menambahkan method printArray()

```
32 private void sort(int data[], int left, int right) {  
33     if (left < right) {  
34         int middle = (left + right) / 2;  
35         sort(data, left, middle);  
36         sort(data, middle + 1, right);  
37         merge(data, left, middle, right);  
38     }  
39 }  
40  
41  
42 public void printArray(int arr[]) {  
43     int n = arr.length;  
44     for (int i = 0; i < n; i++) {  
45         System.out.print(arr[i] + " ");  
46     }  
47     System.out.println();  
48 }  
49  
50 }  
51
```

4. Hasil

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

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!

Jawaban

```
32     System.out.println(x:"-----");
33     System.out.println(x:"Data keseluruhan Mahasiswa : ");
34     data.tampil();
35     System.out.println(x:"_____");
36     System.out.println(x:"_____");
37     System.out.println(x:"Pencarian Data: ");
38     System.out.println(x:"Masukkan Nama Mahasiswa yang dicari: ");
39     System.out.print (s:"Nama: ");
40     String cari = s.next();
41     System.out.println(x:"-----");
42     // int posisi = data. FindSeqSearch (cari);
43     data.countSearchName(cari);
44     int posisi = data.FindBinarySearch(cari, left:0, jumMhs - 1, count:0 );
45     System.out.println(x:"-----");
46     System.out.println(x:"menggunakan binary Search");
47     data.TampilposisiNama(cari, posisi);
48     data.TampilDataNama(cari, posisi);
49
50
51 }
52
```

```
77 public int FindBinarySearch(String cari, int left, int right, int count) {
78     if (right >= left) {
79         int mid = (left + right) / 2;
80         int comparison = listMhs[mid].nama.compareTo(cari);
81
82         if (comparison < 0) {
83             return FindBinarySearch(cari, mid + 1, right, count);
84         } else if (comparison > 0) {
85             return FindBinarySearch(cari, mid - 1, left, count);
86         } else {
87             return mid;
88         }
89     }
90     return -1;
91 }
92
```

```
58 public void TampilposisiNama(String x, int pos) {
59     if (pos != -1) {
60         System.out.println("data : " + x + " ditemukan pada indeks " + pos);
61     } else {
62         System.out.println("data " + x + " tidak ditemukan");
63     }
64 }
65
66 public void TampilDataNama(String x, int pos) {
67     if (pos != -1) {
68         System.out.println("Nim\t : " + x);
69         System.out.println("Nama\t : " + listMhs[pos].nama);
70         System.out.println("Umur\t : " + listMhs[pos].umur);
71         System.out.println("IPK\t : " + listMhs[pos].ipk);
72     } else {
73         System.out.println("data " + x + " tidak ditemukan");
74     }
75 }
76
```

```
94 public void countSearchName(String cari) {
95     int count = 0;
96     for(int i=0; i<listMhs.length; i++){
97         if(cari.equalsIgnoreCase(listMhs[i].nama)){
98             count += 1;
99         }
100     }
101
102     if (count > 1) {
103         System.out.println(x:"Data yang ditemukan lebih dari 1");
104         return;
105     }
106 }
107
```

