

LAPORAN PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA
JOBSHEET 7



Oleh:

FAUZIYYAH ADELIA RAMANDA

NIM. 2341760145

SIB-1F / 10

D-IV SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

6.2 Searching/ Pencarian Menggunakan Algoritma Sequential Search

6.2.1 Code:

```
package Pertemuan06;

import javax.sound.sampled.SourceDataLine;

public class Sorting10 {
    public static void sequentialSearch(int[] arr, int key){
        for (int i=0; i<arr.length; i++) {
            if (i == key){
                System.out.println("Data ditemukan pada indkes ke-" + i);
            }
        }
        System.out.println(x:"Data tidak ditemukan");
    }
    Run | Debug
    public static void main(String[] args){
        int[] daftarNilai = {10, 5, 20, 15, 80, 45};
        sequentialSearch(daftarNilai, key:5);
    }
}
```

Output:

```
01d\01n - Pertemuan06.Sorting10
Data ditemukan pada indkes ke-5
Data tidak ditemukan
D:\C:\Users\UGEP\Documents\Kuliah\vitiv
```

6.2.2 Sequential Search Menggunakan Array of Object

```
Mahasiswa10.java > {} JOBSHEET7
1 package JOBSHEET7;
2
3 public class Mahasiswa10 {
4     int nim;
5     String nama;
6     int umur;
7     double ipk;
8
9     Mahasiswa10(int ni, String n, int u, double i){
10         nim = ni;
11         nama = n;
12         umur = u;
13         ipk = i;
14     }
15     void tampil(){
16         System.out. println("Nim =" + nim);
17         System.out. println("Nama =" + nama);
18         System.out. println("Umur =" + umur);
19         System.out. println("IPK =" + ipk);
20     }
21 }
22 }
```

```

PencarianMhs.java | 1 | K0100177
package K0100177;

public class PencarianMhs {
    Mahasiswa[] listMhs; // new Mahasiswa[] (5);
    int jml;

    void tambah(Mahasiswa m) {
        if (jml + listMhs.length == listMhs.length) {
            listMhs = new Mahasiswa[jml + listMhs.length];
        }
        listMhs[jml] = m;
        jml++;
        System.out.println("Data sudah tersimpan");
    }

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

    public int findIndexSearch(int cari) {
        int awal = 0;
        for (int i = 0; i < listMhs.length; i++) {
            if (listMhs[i].nama.equals(cari)) {
                awal = i;
                break;
            }
        }
        return awal;
    }

    public void tampilKecili(int a, int pec) {
        if (a == 0) {
            System.out.println("Nama : " + a + " ditemukan pada baris " + pec);
        } else {
            System.out.println("Nama : " + a + " tidak ditemukan");
        }
    }

    public void tampilData(int a, int pec) {
        if (a == 0) {
            System.out.println("Nama : " + a);
            System.out.println("Usia : " + listMhs[0].usia);
            System.out.println("Rata-rata : " + listMhs[0].rata);
            System.out.println("IPK : " + listMhs[0].ipk);
        } else {
            System.out.println("Nama : " + a + " tidak ditemukan");
        }
    }
}

MahasiswaMhs.java
package K0100177;

import java.util.Scanner;

public class MahasiswaMhs {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        Scanner ts = new Scanner(System.in);

        PencarianMhs data = new PencarianMhs();
        int jumlah = 5;

        System.out.println("-----");
        System.out.println("Masukkan data mahasiswa secara urut dari Nm terkecil :");
        for (int i = 0; i < jumlah; i++) {
            System.out.println("-----");
            System.out.println("Nama : ");
            String nama = s.next();
            System.out.println("Usia : ");
            String usia = s.nextInt();
            System.out.println("Rata-rata : ");
            int rata = s.nextInt();
            System.out.println("IPK : ");
            double ipk = s.nextDouble();

            Mahasiswa m = new Mahasiswa(nama, usia, rata, ipk);
            data.tambah(m);
        }

        System.out.println("-----");
        System.out.println("Data keseluruhan Mahasiswa :");
        data.tampil();

        System.out.println("-----");
        System.out.println("Pencarian Data :");
        System.out.println("Masukkan Nm Mahasiswa yang dicari :");
        System.out.println("Nm : ");
        int cari = s.nextInt();
        System.out.println("Masukkan baris awal :");
        int awal = data.findIndexSearch(cari);

        data.tampilKecili(cari, awal);
        data.tampilData(cari, awal);
    }
}

```

6.3.2 Verifikasi Hasil Percobaan

```

-----
Masukkan data mahasiswa secara urut dari Nm terkecil:
Nm : 2340
Nama : Dewi Lestari
Usia : 23
IPK : 3.5
-----
Nm : 2342
Nama : Sinta Sanjaya
Usia : 22
IPK : 3.7
-----
Nm : 2345
Nama : Donang Adi
Usia : 25
IPK : 3.8
-----
Nm : 2348
Nama : Budi Prakarsa
Usia : 28
IPK : 3.9
-----
Nm : 2349
Nama : Vania Siti
Usia : 29
IPK : 3.9
-----
Data keseluruhan Mahasiswa:
Nm +2340
Nama +Dewi Lestari
Usia +23
IPK +3.5
-----
Nm +2342
Nama +Sinta Sanjaya
Usia +22
IPK +3.7
-----
Nm +2345
Nama +Donang Adi
Usia +25
IPK +3.8
-----
Nm +2348
Nama +Budi Prakarsa
Usia +28
IPK +3.9
-----
Nm +2349
Nama +Vania Siti
Usia +29
IPK +3.9

```

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);
}
```

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

2. Jelaskan perbedaan metod **TampilData** dan **Tampilposisi** pada class PencarianMhs
 - TampilData untuk menampilkan data mahasiswa berdasarkan indeks yang diberikan
 - TampilPosisi untuk menampilkan indeks dari data mahasiswa yang dicari
3. Jelaskan fungsi **break** pada kode program dibawah ini

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

- Fungsi break yaitu untuk menghentikan proses pencarian data mahasiswa 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!
 - Program masih bisa berjalan meskipun data NIM tidak terurut dari kecil ke yang terbesar. Hasil yang dikeluarkan tetap benar, asalkan algoritma pencarian yang digunakan tidak bergantung pada urutan data

6.1. Searching / Pencarian Menggunakan Binary Search

6.3.1. Langkah-langkah Percobaan Binary Search menggunakan Array

Code:

```
package com.m27;  
  
import java.util.Scanner;  
  
public class BinarySearch {  
    public static void sequentialSearch(int[] arr, int key){  
        for (int i=0; i<arr.length; i++){  
            if (i == key){  
                System.out.println("Data ditemukan pada indeks ke-" + i);  
            }  
        }  
        System.out.println("Data tidak ditemukan");  
    }  
  
    public static void main(String[] args){  
        int[] daftarNilai = {10, 5, 20, 15, 1, 40};  
        sequentialSearch(daftarNilai, 0);  
        int[] sortedNilai = {5, 10, 20, 30, 40, 50};  
        int index = binarySearch(sortedNilai, 0);  
  
        if (index != -1){  
            System.out.println("Data ditemukan pada indeks ke-" + index);  
        }  
        else{  
            System.out.println("Data tidak ditemukan");  
        }  
    }  
  
    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;  
    }  
}
```

Output:

```
Data ditemukan pada indeks ke-5  
Data tidak ditemukan  
Data ditemukan pada indeks ke-1
```

6.3.2 Langkah-langkah Percobaan Binary Search menggunakan Array of Object

```
}  
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;  
}
```

```

        int cari = s.nextInt();
        System.out.println(x:"menggunakan sequential Search");
        int posisi = data.FindSeqSearch(cari);
        System.out.println(x:"=====");
        System.out.println(x:"menggunakan binary Search");
        posisi = data.FindBinarySearch(cari, left:0, jumMhs - 1);

        data.Tampilposisi(cari, posisi);
        data.TampilData(cari, posisi);
    }
}

```

6.3.2. Pertanyaan

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

```

    if (right > left) {
        mid = (left + right) / 2;
        if (cari == listMhs[mid].nim) {

```

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

```

        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;
}

```

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

```

=====
Masukkan data mahasiswa secara urut dari Nim terkecil
=====
Nim : 2340
Nama : Dewi Lestari
Umur : 23
IPK : 3.5
=====
Nim : 2342
Nama : Sinta Sanjaya
Umur : 22
IPK : 3.7
=====
Nim : 2345
Nama : Gungang Adi
Umur : 25
IPK : 3.8
=====
Nim : 2348
Nama : Budi Prakarsa
Umur : 28
IPK : 3.9
=====
Nim : 2349
Nama : Wania Siti
Umur : 29
IPK : 3.9
=====

Data keseluruhan Mahasiswa:
Nim >2340
Nama >Dewi Lestari
Umur >23
IPK >3.5
=====
Nim >2342
Nama >Sinta Sanjaya
Umur >22
IPK >3.7
=====
Nim >2345
Nama >Gungang Adi
Umur >25
IPK >3.8
=====
Nim >2348
Nama >Budi Prakarsa
Umur >28
IPK >3.9
=====
Nim >2349
Nama >Wania Siti
Umur >29
IPK >3.9

```

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.

6.2. Percobaan Pengayaan Divide and Conquer

6.4.1. Langkah-langkah Percobaan Merge Sort

```
public class SortMain {  
    Run | Debug  
    public static void main(String[] args){  
        int data[] = {10,40,30,50,70,20,100,90};  
        System.out.println(x:"sorting dengan merge sort");  
        MergeSorting10 mSort = new MergeSorting10();  
        System.out.println(x:"data awal");  
        mSort.printArray(data);  
        mSort.mergeSort(data);  
        System.out.println(x:"setelah diurutkan");  
        mSort.printArray(data);  
    }  
}
```

6.4.2. Verifikasi Hasil Percobaan

```
sorting dengan merge sort  
data awal  
10403050702010090  
setelah diurutkan  
10203040507090100  
C:\Users\UC5F\Documents\Modul
```

6.3. 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!

[illegible]


```

// Import the express module
const express = require('express');

// Create an express application
const app = express();

// Set the port to 3000
const port = 3000;

// Start the server
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});

// Import the body-parser module
const bodyParser = require('body-parser');

// Use body-parser to parse incoming request bodies
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));

// Import the mongoose module
const mongoose = require('mongoose');

// Connect to the MongoDB database
mongoose.connect('mongodb://localhost:27020/notes', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});

// Import the note model
const Note = mongoose.model('Note');

// Create a new note
const createNote = (req, res) => {
  const { title, content } = req.body;

  if (!title || !content) {
    res.status(400).send('Title and content are required');
  }

  const newNote = new Note({ title, content });
  newNote.save().then(() => {
    res.status(201).send('Note created successfully');
  }, (err) => {
    res.status(500).send('Error creating note');
  });
};

// Get all notes
const getAllNotes = (req, res) => {
  Note.find().then((notes) => {
    res.status(200).send(notes);
  }, (err) => {
    res.status(500).send('Error fetching notes');
  });
};

// Get a note by ID
const getNoteById = (req, res) => {
  const { id } = req.params;

  Note.findById(id).then((note) => {
    if (!note) {
      res.status(404).send('Note not found');
    }
    res.status(200).send(note);
  }, (err) => {
    res.status(500).send('Error fetching note');
  });
};

// Update a note
const updateNote = (req, res) => {
  const { id } = req.params;
  const { title, content } = req.body;

  Note.findByIdAndUpdate(id, { title, content }, { new: true }).then((note) => {
    if (!note) {
      res.status(404).send('Note not found');
    }
    res.status(200).send(note);
  }, (err) => {
    res.status(500).send('Error updating note');
  });
};

// Delete a note
const deleteNote = (req, res) => {
  const { id } = req.params;

  Note.findByIdAndDelete(id).then(() => {
    res.status(200).send('Note deleted successfully');
  }, (err) => {
    res.status(500).send('Error deleting note');
  });
};

// Export the routes
module.exports = {
  createNote,
  getAllNotes,
  getNoteById,
  updateNote,
  deleteNote,
};

```

```
Data ditemukan pada indeks ke-5
Data tidak ditemukan
=====
Masukkan berapa banyak data yang ingin ditambahkan : 1
=====
Masukkan data mahasiswa secara urut dari nim terkecil
=====
Nim      : 234
Nama     : ujii
Umur     : 20
IPK      : 3.9
=====
Data keseluruhan mahasiswa :
Nim ~234
Nama ~ujii
Umur ~20
IPK ~3.9
=====
=====
Pencarian data :
Masukkan nama mahasiswa yang dicari : ujii
=====
=====
Menggunakan binary search
Data      : ujii ditemukan pada indeks 0
Nim       : ujii
Nama      : ujii
Umur      : 20
IPK       : 3.9
```

