

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA
JOBSHEET 6



NAMA : DIMAS ADI BAYU SAMUDRA

KELAS : 1A

NO. ABSEN : 08

NIM : 2341720169

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2024

PRAKTIKUM

Percobaan 1

Code Buku

```
package jobsheet6;

public class Buku08 {
    int kodeBuku;
    String judul;
    int tahunTerbit;
    String pengarang;
    int stock;

    public Buku08(int kodeBuku, String judul, int tahunTerbit, String
pengarang, int stock){
        this.kodeBuku = kodeBuku;
        this.judul = judul;
        this.tahunTerbit = tahunTerbit;
        this.pengarang = pengarang;
        this.stock = stock;
    }

    void tampil(){

    }

    void tampilDataBuku(){
        System.out.println("Kode buku : " + kodeBuku);
        System.out.println("Judul buku : " + judul);
        System.out.println("tahun terbit buku : " + tahunTerbit);
        System.out.println("Pengarang buku : " + pengarang);
        System.out.println("Stock buku : " + stock);
    }
}
```

Code PencarianBuku

```
package jobsheet6;

public class PencarianBuku08 {
    Buku08 listBuku08[] = new Buku08[5];
    int idx;

    void tambah(Buku08 m){
        if (idx < listBuku08.length) {
            listBuku08[idx] = m;
            idx++;
        } else {
            System.out.println("Data Sudah Penuh");
        }
    }
}
```

```

void tampil(){
    for (Buku08 m : listBuku08) {
        m.tampilDataBuku();
    }
}

public int FindSeqSearch(int cari){
    int posisi = -1;
    for (int j = 0; j < listBuku08.length; j++) {
        if (listBuku08[j].kodeBuku == cari) {
            j = posisi;
            break;
        }
    }
    return posisi;
}

public void tampilPosisi(int x, int pos){
    if (pos != -1) {
        System.out.println("Data   : " + x + "   Ditemukan pada index : "
+ pos );
    } else {
        System.out.println("Data   : " + x + "   tidak Ditemukan" );
    }
}

public void TampilData(int x, int pos) {
    if (pos != -1) {
        System.out.println("Kode Buku\t: " + x );
        System.out.println("Judul\t\t: " + listBuku08[pos].judul);
        System.out.println("Tahun Terbit\t: " +
listBuku08[pos].tahunTerbit);
        System.out.println("Pengarang\t\t: " +
listBuku08[pos].pengarang);
        System.out.println("Stock\t\t: " + listBuku08[pos].stock);
    } else {
        System.out.println("data " + x + " tidak ditemukan");
    }
}
}

```

Code mainBuku

```

package jobsheet6;

import java.util.Scanner;

public class mainBuku08 {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        Scanner s1 = new Scanner(System.in);

        PencarianBuku08 data = new PencarianBuku08();
        int jumBuku = 5;
    }
}

```

```

        System.out.println("_____");
        System.out.println("Masukkan data Buku secara Urut dari KodeBuku
Terkecil : ");
        for (int i = 0; i < jumBuku; i++) {
            System.out.println("_____");
            System.out.print("Kode Buku \t: ");
            int kodeBuku = s.nextInt();
            System.out.print("Judul buku \t: ");
            String judulBuku = s1.nextLine();
            System.out.print("Tahun Terbit \t: ");
            int tahunTerbit = s.nextInt();
            System.out.print("Pengarang \t: ");
            String pengarang = s1.nextLine();
            System.out.print("Stock \t\t: ");
            int stock= s.nextInt();

            Buku08 m= new Buku08(kodeBuku, judulBuku, tahunTerbit, pengarang,
stock);
            data.tambah(m);
        }

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

        System.out.println("_____");
        System.out.println("Pencarian Data ");
        System.out.println("Masukkan Kode Buku yang dicari: ");
        System.out.print("Kode Buku : ");
        int cari = s.nextInt();
        System.out.println("menggunakan sequential Search");
        int posisi = data.FindSeqSearch(cari);
        data.tampilPosisi(cari, posisi);
        data.TampilData(cari, posisi);

    }
}

```

Hasil

Masukkan data Buku secara Urut dari KodeBuku Terkecil :

Kode Buku : 111
Judul buku : Alogritma
Tahun Terbit : 2019
Pengarang : Wahyuni
Stock : 5

Kode Buku : 123
Judul buku : Big Data
Tahun Terbit : 2020
Pengarang : Susilo
Stock : 3

Kode Buku : 125
Judul buku : Desain UI
Tahun Terbit : 2021
Pengarang : Supriadi
Stock : 3

Kode Buku : 126
Judul buku : Web Programming
Tahun Terbit : 2022
Pengarang : Pustaka Adi
Stock : 2

Kode Buku : 127
Judul buku : Etika Mahasiswa
Tahun Terbit : 2023
Pengarang : Darmawan Adi
Stock : 2

Data keseluruhan Mahasiswa :

Data keseluruhan Buku :

Kode buku : 111
Judul buku : Alogritma
tahun terbit buku : 2019
Pengarang buku : Wahyuni
Stock buku : 5
Kode buku : 123
Judul buku : Big Data
tahun terbit buku : 2020
Pengarang buku : Susilo
Stock buku : 3
Kode buku : 125
Judul buku : Desain UI
tahun terbit buku : 2021
Pengarang buku : Supriadi
Stock buku : 3
Kode buku : 126
Judul buku : Web Programming
tahun terbit buku : 2022
Pengarang buku : Pustaka Adi
Stock buku : 2
Kode buku : 127
Judul buku : Etika Mahasiswa
tahun terbit buku : 2023
Pengarang buku : Darmawan Adi
Stock buku : 2

Jika Data Ditemukan

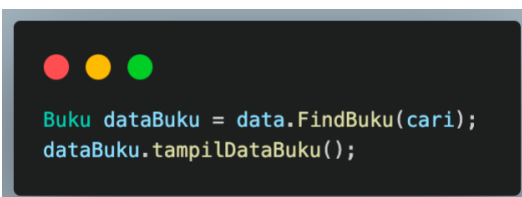
```
Pencarian Data
Masukkan Kode Buku yang dicari:
Kode Buku : 111
menggunakan sequential Search
Data : 111 Ditemukan pada index : 0
Kode Buku      : 111
Judul   : Algoritma
Tahun Terbit  : 2019
Pengarang    : Wahyuni
Stock       : 5
```

Jika Data Tidak Ditemukan

```
Pencarian Data
Masukkan Kode Buku yang dicari:
Kode Buku : 124
menggunakan sequential Search
Data : 124 tidak Ditemukan
data 124 tidak ditemukan
```

Pertanyaan

1. Jelaskan fungsi break yang ada pada method FindSeqSearch!
2. Jika Data Kode Buku yang dimasukkan tidak teratur dari kecil ke besar. Apakah program masih dapat berjalan? Apakah hasil yang dikeluarkan benar? Tunjukkan hasil screenshot untuk bukti dengan kode Buku yang acak. Jelaskan Mengapa hal tersebut bisa terjadi?
3. Buat method baru dengan nama FindBuku menggunakan konsep sequential search dengan tipe method dari FindBuku adalah BukuNoAbsen. Sehingga Anda bisa memanggil method tersebut pada class BukuMain seperti gambar berikut :



```
Buku dataBuku = data.FindBuku(cari);
dataBuku.tampilDataBuku();
```

Jawaban :

1. break digunakan untuk keluar dari loop setelah menemukan elemen yang cocok, sehingga tidak perlu memeriksa elemen yang tersisa.
2. Iya bisa, program masih dapat berjalan meskipun data kode buku tidak teratur dari kecil ke besar. Program masih dapat berjalan dengan baik meskipun data kode buku tidak teratur. Hasil yang dikeluarkan .

Masukkan data Buku secara Urut dari KodeBuku Terkecil :

Kode Buku : 123
Judul buku : Big Data
Tahun Terbit : 2020
Pengarang : Susilo
Stock : 3

Kode Buku : 127
Judul buku : Etika Mahasiswa
Tahun Terbit : 2023
Pengarang : Darmawan Adi
Stock : 2

Kode Buku : 125
Judul buku : Desain UI
Tahun Terbit : 2021
Pengarang : Supriadi
Stock : 3

Kode Buku : 111
Judul buku : Algoritma
Tahun Terbit : 2019
Pengarang : Wahyuni
Stock : 5

Kode Buku : 126
Judul buku : Web Programming
Tahun Terbit : 2022
Pengarang : Pustaka Adi
Stock : 2

Data keseluruhan Mahasiswa :

Data keseluruhan Buku :

Kode buku : 123
Judul buku : Big Data
tahun terbit buku : 2020
Pengarang buku : Susilo
Stock buku : 3
Kode buku : 127
Judul buku : Etika Mahasiswa
tahun terbit buku : 2023
Pengarang buku : Darmawan Adi
Stock buku : 2
Kode buku : 125
Judul buku : Desain UI
tahun terbit buku : 2021
Pengarang buku : Supriadi
Stock buku : 3
Kode buku : 111
Judul buku : Algoritma
tahun terbit buku : 2019
Pengarang buku : Wahyuni
Stock buku : 5
Kode buku : 126
Judul buku : Web Programming
tahun terbit buku : 2022
Pengarang buku : Pustaka Adi
Stock buku : 2

Pencarian Data

Masukkan Kode Buku yang dicari:

Kode Buku : 123

menggunakan sequential Search

Data : 123 Ditemukan pada index : 0

Kode Buku : 123

Judul : Big Data

Tahun Terbit : 2020

Pengarang : Susilo

Stock : 3

3. Code tambahan pada Class PencarianBuku

```
public Buku08 FindBuku(int cari) {  
    for (int j = 0; j < listBuku08.length; j++) {  
        if (listBuku08[j].kodeBuku == cari) {  
            return listBuku08[j];  
        }  
    }  
    return null;  
}
```

Code tambahan pada mainBuku

```
Buku08 dataBuku = data.FindBuku(cari);  
dataBuku.tampilDataBuku();
```

Hasil

```
-----  
Pencarian Data  
Masukkan Kode Buku yang dicari:  
Kode Buku : 111  
menggunakan sequential Search  
Data : 111 Ditemukan pada index : 0  
Kode Buku      : 111  
Judul          : Algoritma  
Tahun Terbit   : 2019  
Pengarang      : Wahyuni  
Stock         : 5  
Kode buku : 111  
Judul buku : Algoritma  
tahun terbit buku : 2019  
Pengarang buku : Wahyuni  
Stock buku : 5
```


Percobaan 2

Tamabahan Code pada PencarianBuku

```
public int FindBinarySearch(int cari, int left, int right) {
    int mid;
    if (right >= left) {
        mid = (right) / 2;
        if (cari == listBuku08[mid].kodeBuku) {
            return (mid);
        } else if (listBuku08[mid].kodeBuku > cari) {
            return FindBinarySearch(cari, left, mid);
        } else {
            return FindBinarySearch(cari, mid, right);
        }
    }
    return -1;
}
```

Tambahan Code pada mainBuku

```
System.out.println("=====");
System.out.println("menggunakan binary Search");
posisi = data.FindBinarySearch(cari, 0, jumBuku - 1);
data.tampilPosisi(cari, posisi);
data.TampilData(cari, posisi);
```

Hasil

```
Pencarian Data
Masukkan Kode Buku yang dicari:
Kode Buku : 111
menggunakan sequential Search
Data : 111 Ditemukan pada index : 0
Kode Buku      : 111
Judul          : Algoritma
Tahun Terbit   : 2019
Pengarang      : Wahyuni
Stock          : 5
Kode buku : 111
Judul buku : Algoritma
tahun terbit buku : 2019
Pengarang buku : Wahyuni
Stock buku : 5
=====
menggunakan binary Search
Data : 111 Ditemukan pada index : 0
Kode Buku      : 111
Judul          : Algoritma
Tahun Terbit   : 2019
Pengarang      : Wahyuni
Stock          : 5
```

Pertanyaan

1. Tunjukkan pada kode program yang mana proses divide dijalankan!
2. Tunjukkan pada kode program yang mana proses conquer dijalankan!
4. Jika data Kode Buku yang dimasukkan tidak urut. Apakah program masih dapat berjalan? Mengapa demikian! Tunjukkan hasil screenshoot untuk bukti dengan kode Buku yang acak. Jelaskan

Mengapa hal tersebut bisa terjadi?

3. Jika Kode Buku yang dimasukkan dari Kode Buku 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 search agar hasilnya sesuai!

Jawaban

1. kode program proses divide

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

2. kode program proses conquer dijalankan

```
if (cari == listBuku08[mid].kodeBuku) {
    return (mid);
} else if (listBuku08[mid].kodeBuku > cari) {
    return FindBinarySearch(cari, left, mid - 1);
} else {
    return FindBinarySearch(cari, mid + 1, right);
}
```

3. Jika data kode buku yang dimasukkan tidak terurut, program masih akan berjalan, tetapi tidak akan menghasilkan hasil yang benar karena asumsi dasar pencarian biner adalah data yang terurut.

4. Code yang diubah

```
public int FindBinarySearch(int cari, int left, int right) {
    if (left <= right) {
        int mid = left + (right - left) / 2;

        if (listBuku08[mid].kodeBuku == cari) {
            return mid;
        } else if (listBuku08[mid].kodeBuku < cari) {
            return FindBinarySearch(cari, left, mid - 1);
        } else {
            return FindBinarySearch(cari, mid + 1, right);
        }
    }
    return -1;
}
```

Hasil

```
=====
menggunakan binary Search
Data : 20213 Ditemukan pada index : 2
Kode Buku      : 20213
Judul          : Desain UI
Tahun Terbit   : 2021
Pengarang      : Supriadi
Stock          : 3
```

Percobaan 3

Code MergeSorting

```
package jobsheet6.MergeSortTest;

public class MergeSorting08 {
    public void mergeSort(int [] data){
        sort(data, 0, data.length-1);
    }

    public void merge(int data[], int left, int middle, int right){
        int[] temp = new int[data.length];
        for (int i = left; i <= right; i++) {
            temp[i] = data[i];
        }

        int a = left;
        int b = middle + 1;
        int c = left;

        while (a <= middle && b <= right) {
            if (temp[a] <= temp[b]) {
                data[c] = temp[a];
                a++;
            } else {
                data[c] = temp[b];
                b++;
            }
            c++;
        }

        int s = middle - a;
        for (int i = 0; i <= s; i++) {
            data[c + i] = temp[a + i];
        }
    }

    public void sort(int data[], int left, int right){
        if (left < right) {
            int middle = (left + right) / 2;
            sort(data, left, middle);
            sort(data, middle + 1, right);
            merge(data, left, middle, right);
        }
    }

    public void printArray(int arr[]){
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
    }
}
```

Code mainMergeSort

```
package jobsheet6.MergeSortTest;

public class mainMergeSort {
    public static void main(String[] args) {
        int data[] = {10,40,30,50,70,20,100,90};
        System.out.println("sorting dengan merge sort");
        MergeSorting08 mSort= new MergeSorting08();
        System.out.println("data awal");
        mSort.printArray(data);
        mSort.mergeSort(data);
        System.out.println();
        System.out.println("setelah diurutkan");
        mSort.printArray(data);
    }
}
```

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
```

Latihan Praktikum 1

Code yang di ubah

```
public int findBinarySearch(String cari, int left, int right) {
    if (right >= left) {
        int mid = (left + right) / 2;
        int comparison = cari.compareTo(listBuku08[mid].kodeBuku);
        if (comparison == 0) {
            return mid;
        } else if (comparison < 0) {
            return findBinarySearch(cari, left, mid - 1);
        } else {
            return findBinarySearch(cari, mid + 1, right);
        }
    }
    return -1;
}
```

```
public int findSeqSearch(String cari) {
    int posisi = -1;
    for (int j = 0; j < listBuku08.length; j++) {
        if (listBuku08[j].kodeBuku.equals(cari)) {
            posisi = j;
            break;
        }
    }
    return posisi;
}
```

```
System.out.print("Cari buku berdasarkan judul: ");
String cariJudul = s1.nextLine();

int posisi = data.findSeqSearchJudul(cariJudul);
System.out.println("sequential search: ");
data.TampilData(cariJudul, posisi);

System.out.println("binary search: ");
posisi = data.findJudulBinarySearch(cariJudul, 0,
data.listBuku08.length - 1);
data.TampilData(cariJudul, posisi);
```

Hasil

```
Cari buku berdasarkan judul: Algoritma
sequential search:
Kode Buku      : Algoritma
Judul          : Algoritma
Tahun Terbit   : 2019
Pengarang      : Wahyuni
Stock          : 5
binary search:
Kode Buku      : Algoritma
Judul          : Algoritma
Tahun Terbit   : 2019
Pengarang      : Wahyuni
Stock          : 5
```

Latihan Praktikum 2

Code yang di ubah

```
public int findBinarySearch(String cari, int left, int right) {
    if (right >= left) {
        int mid = (left + right) / 2;
        int comparison = cari.compareTo(listBuku08[mid].kodeBuku);

        if (comparison == 0) {
            return mid;
        } else if (comparison < 0) {
            return findBinarySearch(cari, left, mid - 1);
        } else {
            return findBinarySearch(cari, mid + 1, right);
        }
    }
    return -1;
}

int findJudulBinarySearch(String cari, int left, int right) {
    if (right >= left) {
        int mid = left + (right - left) / 2;

        int comparison =
listBuku08[mid].judul.compareToIgnoreCase(cari);

        if (comparison == 0) {
            return mid;
        } else if (comparison < 0) {
            return findJudulBinarySearch(cari, mid + 1, right);
        } else {
            return findJudulBinarySearch(cari, left, mid - 1);
        }
    }
    return -1;
}

void sort() {
    int length = listBuku08.length;
    for (int i = 0; i < length - 1; i++) {
        for (int j = 0; j < length - i - 1; j++) {
            if (listBuku08[j].judul.compareToIgnoreCase(listBuku08[j +
1].judul) > 0) {
                Buku08 temp = listBuku08[j];
                listBuku08[j] = listBuku08[j + 1];
                listBuku08[j + 1] = temp;
            }
        }
    }
}

int findSeqSearchJudul(String cari) {
    int count = 0;
```

```

        int posisi = -1;
        for (int j = 0; j < listBuku08.length; j++) {
            if
(listBuku08[j].judul.toLowerCase().contains(cari.toLowerCase())) {
                posisi = j;
                count++;
            }
        }
        if (count > 1) {
            System.out.println(count + " data ditemukan dengan judul yang
sama!");
        }
        return posisi;
    }
}

```

Hasil

```

Cari buku berdasarkan judul: b
sequential search:
Kode Buku      : b
Judul          : b
Tahun Terbit   : 2011
Pengarang      : b
Stock          : 2
binary search:
Kode Buku      : b
Judul          : b
Tahun Terbit   : 2011
Pengarang      : b
Stock          : 2

```