

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA
JOBSHEET 16



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

Kode LoopCollection

```
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;

public class LoopCollection {

    public static void main(String[] args) {
        List<Object> l = new ArrayList<>();
        l.add(1);
        l.add(2);
        l.add(3);
        l.add("Cireng");
        System.out.printf("Elemen 0: %s total semua elemen : %d elemen
terakhir : %s\n", l.get(0), l.size(), l.get(l.size() - 1));

        l.add(4);
        l.remove(0);
        System.out.printf("Elemen 0: %s total semua elemen : %d elemen
terakhir : %s\n", l.get(0), l.size(), l.get(l.size() - 1));

        List<String> names = new LinkedList<>();
        names.add("Noureen");
        names.add("Akhleema");
        names.add("Uwais");
        names.add("Al-Qarni");

        System.out.printf("Elemen 0: %s total semua elemen : %d elemen
terakhir : %s\n", names.get(0), names.size(), names.get(names.size() - 1));
        names.set(0, "My kid");
        System.out.printf("Elemen 0: %s total semua elemen : %d elemen
terakhir : %s\n", names.get(0), names.size(), names.get(names.size() - 1));
        System.out.println("Names : " + names.toString());
    }
}
```

Hasil run

```
Elemen 0: 1 total semua elemen : 4 elemen terakhir : Cireng
Elemen 0: 2 total semua elemen : 4 elemen terakhir : 4
Elemen 0: Noureen total semua elemen : 4 elemen terakhir : Al-Qarni
Elemen 0: My kid total semua elemen : 4 elemen terakhir : Al-Qarni
Names : [My kid, Akhleema, Uwais, Al-Qarni]
```

Pertanyaan Percobaan

1. Perhatikan baris kode 25-36, mengapa semua jenis data bisa ditampung ke dalam sebuah ArrayList?
2. Modifikasi baris kode 25-36 sehingga data yang ditampung hanya satu jenis atau spesifik tipe tertentu!
3. Ubah kode pada baris kode 38 menjadi seperti ini

```
LinkedList<String> names = new LinkedList<>();
```

4. Tambahkan juga baris berikut ini, untuk memberikan perbedaan dari tampilan yang sebelumnya

```
names.push("Mei-mei");
System.out.printf("Elemen 0: %s total elemen: %s elemen terakhir: %s\n",
    names.getFirst(), names.size(), names.getLast());
System.out.println("Names: " + names.toString());
```

5. Dari penambahan kode tersebut, silakan dijalankan dan apakah yang dapat Anda jelaskan

Jawaban

1. ArrayList<Object> dapat menampung berbagai jenis data karena Object adalah kelas dasar dari semua kelas di Java. Hal ini memungkinkan daftar tersebut untuk menyimpan objek dari tipe apa pun, seperti Integer, String, atau objek lainnya
2. Kode yg dirubah

```
List<Integer> l = new ArrayList<>();
    l.add(1);
    l.add(2);
    l.add(3);
    l.add(4);
    System.out.printf("Elemen 0: %d total semua elemen : %d elemen
terakhir : %d\n", l.get(0), l.size(), l.get(l.size() - 1));

    l.add(5);
    l.remove(0);
    System.out.printf("Elemen 0: %d total semua elemen : %d elemen
terakhir : %d\n", l.get(0), l.size(), l.get(l.size() - 1));
```

Hasil run

```
Elemen 0: 1 total semua elemen : 4 elemen terakhir : 4
Elemen 0: 2 total semua elemen : 4 elemen terakhir : 5
```

- 3.

```
LinkedList<String> names = new LinkedList<>();
```

- 4.

```
Elemen 0: Mei - mei total semua elemen : 5 elemen terakhir : Al-Qarni
Names : [Mei - mei, My kid, Akhleema, Uwais, Al-Qarni]
```

5. ArrayList akan menampilkan elemen-elemen Integer dan memperbarui jumlah elemen setelah penghapusan elemen pertama.

LinkedList akan menampilkan elemen-elemen String dan memperbarui jumlah serta urutan elemen setelah penambahan elemen baru menggunakan push().

Penambahan elemen menggunakan push() pada LinkedList menambahkan elemen ke awal daftar, sehingga "Mei-mei" akan menjadi elemen pertama.

Percobaan 2

Kode LoopCollection 2

```
import java.util.Iterator;
import java.util.Stack;
public class LoopCollection2 {
    public static void main(String[] args) {
        Stack<String> fruits = new Stack<>();
        fruits.push("Banana");
        fruits.push("Orange");
        fruits.add("Watermelon");
        fruits.add("Leci");
        fruits.push("Salak");
        for (String fruit : fruits) {
            System.out.printf("%s ", fruit);
        }
        System.out.println("\n" + fruits.toString());
        while (!fruits.empty()) {
            System.out.printf("%s ", fruits.pop());
        }

        fruits.push("Melon");
        fruits.push("Durian");
        System.out.println("");

        for (Iterator<String> it = fruits.iterator(); it.hasNext();) {
            String fruit = it.next();
            System.out.printf("%s ", fruit);
        }

        System.out.println("");
        fruits.stream().forEach(e -> {
            System.out.printf("%s ", e);
        });

        System.out.println("");
        for (int i = 0; i < fruits.size(); i++) {
            System.out.printf("%s ", fruits.get(i));
        }
    }
}
```

Hasil run

```
Banana Orange Watermelon Leci Salak  
[Banana, Orange, Watermelon, Leci, Salak]  
Salak Leci Watermelon Orange Banana  
Melon Durian  
Melon Durian  
Melon Durian
```

Pertanyaan Percobaan

1. Apakah perbedaan fungsi push() dan add() pada objek fruits?
2. Silakan hilangkan baris 43 dan 44, apakah yang akan terjadi? Mengapa bisa demikian?
3. Jelaskan fungsi dari baris 46-49?
4. Silakan ganti baris kode 25, Stack<String> menjadi List<String> dan apakah yang terjadi?

Mengapa bisa demikian?

5. Ganti elemen terakhir dari objek fruits menjadi "Strawberry"!
6. Tambahkan 3 buah seperti "Mango", "guava", dan "avocado" kemudian dilakukan sorting!

Jawaban

1. push(): Menambahkan elemen ke bagian atas tumpukan (stack). Ini adalah metode khusus untuk kelas Stack.

add(): Menambahkan elemen ke akhir daftar. Ini adalah metode yang umum digunakan dalam berbagai koleksi Java, termasuk ArrayList.

2. Jika kedua baris ini dihilangkan, Stack<String> fruits akan kosong setelah loop while (!fruits.empty()) { ... } karena semua elemen akan dihapus oleh metode pop().

push() digunakan untuk menambahkan elemen baru ke Stack. Menghapus baris tersebut berarti tidak ada elemen yang ditambahkan kembali setelah penghapusan menggunakan pop(), sehingga Stack akan tetap kosong.

3. Fungsi: Kode ini menggunakan iterator untuk mengiterasi melalui semua elemen di dalam fruits dan mencetak setiap elemen. Ini adalah cara umum untuk mengiterasi melalui koleksi di Java.

4. Mengganti Stack<String> dengan List<String> mengubah tipe deklarasi variabel fruits menjadi tipe antarmuka List, yang tidak memiliki metode push() dan pop(), kode akan error karena push dan pop harus di tambahi casting agar tidak error.

5.

```
Watermelon Leci Strawberry  
[Watermelon, Leci, Strawberry]  
Strawberry Leci Watermelon  
Melon Durian  
Melon Durian  
Melon Durian
```

6.

```
Banana Orange Watermelon Leci Salak  
[Banana, Orange, Watermelon, Leci, Salak]  
Salak Leci Watermelon Orange Banana  
Melon Durian  
Melon Durian  
Melon Durian  
Fruits after sorting:  
Mango Melon Strawberry avocado guava
```

Percobaan 3

Kode Mahasiswa

```
public class Mahasiswa {  
  
    String nim;  
    String nama;  
    String notelp;  
  
    public Mahasiswa() {}  
  
    public Mahasiswa(String nim , String nama, String notelp){  
        this.nim = nim;  
        this.nama = nama;  
        this.notelp = notelp;  
    }  
    @Override  
    public String toString(){  
        return "Mahasiswa {" + "nim = " + nim + ", nama = " + nama + ",  
notelp = " + notelp + "}";  
    }  
}
```

Kode ListMahasiswa

```
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.List;  
  
public class ListMahasiswa {  
    List<Mahasiswa> mahasiswas = new ArrayList<>();  
    public void tambah(Mahasiswa... mahasiswa){  
        mahasiswas.addAll(Arrays.asList(mahasiswa));  
    }  
  
    public void hapus (int index){  
        mahasiswas.remove(index);  
    }  
  
    public void update(int index, Mahasiswa mhs){
```

```

        mahasiswa.set(index, mhs);
    }

    public void tampil(){
        mahasiswa.stream().forEach(mhs -> {
            System.out.println("" + mhs.toString());
        });
    }

    int linearSearch(String nim){
        for (int i = 0; i < mahasiswa.size(); i++) {
            if (nim.equals(mahasiswa.get(i).nim)) {
                return i;
            }
        }
        return -1;
    }

    public static void main(String[] args) {
        ListMahasiswa lm = new ListMahasiswa();
        Mahasiswa m = new Mahasiswa("201234", "Noureen", "021xx1");
        Mahasiswa m1 = new Mahasiswa("201235", "Akhleema", "021xx2");
        Mahasiswa m2 = new Mahasiswa("201236", "Shannumm", "021xx3");

        lm.tambah(m, m1, m2);
        lm.tampil();
        lm.update(lm.linearSearch("201235"), new Mahasiswa("201235" ,
"Akhleema Lela " , "021xx2"));
        System.out.println("");
        lm.tampil();
    }
}

```

Hasil run

```

Mahasiswa {nim = 201234, nama = Noureen, notelp = 021xx1}
Mahasiswa {nim = 201235, nama = Akhleema, notelp = 021xx2}
Mahasiswa {nim = 201236, nama = Shannumm, notelp = 021xx3}

Mahasiswa {nim = 201234, nama = Noureen, notelp = 021xx1}
Mahasiswa {nim = 201235, nama = Akhleema Lela , notelp = 021xx2}
Mahasiswa {nim = 201236, nama = Shannumm, notelp = 021xx3}

```

Pertanyaan Percobaan

1. Pada fungsi tambah() yang menggunakan unlimited argument itu menggunakan konsep apa? Dan kelebihan apa?
2. Pada fungsi linearSearch() di atas, silakan diganti dengan fungsi binarySearch() dari collection!
3. Tambahkan fungsi sorting baik secara ascending ataupun descending pada class tersebut!

Jawaban

1. Fungsi tambah() menggunakan konsep varargs yang memungkinkan kita menambahkan sejumlah argumen yang tidak terbatas saat memanggil fungsi tersebut. Ini membuat penggunaan fungsi lebih fleksibel karena tidak perlu membatasi jumlah argumen secara khusus dalam definisi fungsi.
2. Untuk mengganti fungsi linearSearch() dengan binarySearch(), kita bisa menggunakan metode binarySearch() dari Collections. Namun, perlu diingat bahwa binarySearch() hanya bekerja pada data yang sudah diurutkan. Artinya, jika data Mahasiswa belum diurutkan, perlu diurutkan terlebih dahulu sebelum menggunakan binarySearch().
3. Untuk menambahkan fungsi sorting baik secara naik (ascending) maupun turun (descending) pada kelas tersebut, kita bisa menggunakan metode sort() dari Collections. Atau, jika ingin, kita juga bisa menulis kode untuk algoritma pengurutan sendiri seperti bubble sort atau selection sort. Pastikan untuk memberikan opsi kepada pengguna untuk mengurutkan data sesuai dengan kebutuhan mereka.

Tugas

Kode Mahasiswa08

```
package Tugas;

public class Mahasiswa08 {
    String nim, nama, notelp;

    public Mahasiswa08() {}

    public Mahasiswa08(String nim, String nama, String notelp) {
        this.nim = nim;
        this.nama = nama;
        this.notelp = notelp;
    }
}
```

Kode Nilai08

```
package Tugas;

public class Nilai08 {
    String nim, nama, mataKuliah, sks;
    double nilai;

    public Nilai08(String nim, String nama, String mataKuliah, String sks,
double nilai) {
        this.nim = nim;
        this.nama = nama;
        this.mataKuliah = mataKuliah;
        this.sks = sks;
        this.nilai = nilai;
    }
}
```


Kode MataKuliah08

```
package Tugas;

public class MataKuliah08 {
    String kode, mataKuliah, sks;

    public MataKuliah08(){}

    public MataKuliah08(String kode, String mataKuliah, String sks){
        this.kode = kode;
        this.mataKuliah = mataKuliah;
        this.sks = sks;
    }
}
```

Kode Main08

```
package Tugas;

import java.util.Scanner;
import java.util.Queue;
import java.util.LinkedList;

public class Main08 {
    public static Queue<Mahasiswa08> mahasiswas = new LinkedList<>();
    public static Queue<MataKuliah08> mataKuliahs = new LinkedList<>();
    public static Queue<Nilai08> nilais = new LinkedList<>();

    public static Scanner sc = new Scanner(System.in);

    public static void menu() {

        System.out.println("*****");
        System.out.println("SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER");

        System.out.println("*****");
        System.out.println("1. Input Nilai");
        System.out.println("2. Tampil Nilai");
        System.out.println("3. Mencari Nilai Mahasiswa");
        System.out.println("4. Urut Data Nilai");
        System.out.println("5. Keluar");

        System.out.println("*****");
        System.out.print("Pilih Menu: ");
    }

    public static void inputNilai() {
        System.out.println("--- Masukkan Data ---");
        System.out.print("Masukkan nilai : ");
        double nilai = sc.nextDouble();
        sc.nextLine();
    }
}
```

```

        System.out.println("DAFTAR MAHASISWA");

System.out.println("*****");
        System.out.println("NIM            Nama            Telf");
        mahasiswa.forEach((mhs) -> {
            System.out.printf("%-3s            %-12s            %10s\n",
mhs.nim, mhs.nama, mhs.notelp);
        });

        System.out.print("Masukkan NIM Mahasiswa : ");
        String nim = sc.nextLine();

        System.out.println("DAFTAR MATA KULIAH");

System.out.println("*****");
        System.out.println("Kode            Mata Kuliah
SKS");
        mataKuliah.forEach((mk) -> {
            System.out.printf("%-5s            %-40s            %-2s\n",
mk.kode, mk.mataKuliah, mk.sks);
        });

        System.out.print("Masukkan Kode Mata Kuliah : ");
        String kode = sc.nextLine();
        String nama = "";
        for (Mahasiswa08 mhs : mahasiswa) {
            if (mhs.nim.equals(nim)) {
                nama = mhs.nama;
            }
        }

        String mataKuliah = "";
        String sks = "";
        for (MataKuliah08 mk : mataKuliah) {
            if (mk.kode.equals(kode)) {
                mataKuliah = mk.mataKuliah;
                sks = mk.sks;
            }
        }
        nilai.add(new Nilai08(nim, nama, mataKuliah, sks, nilai));
    }

    public static void tampilNilai() {
        System.out.println("DAFTAR NILAI MAHASISWA");

System.out.println("*****");
        System.out.println("NIM            Nama            Mata Kuliah
SKS            Nilai");
        nilai.forEach((n) -> {

```

```

        System.out.printf("%-5s          %-10s          %-10s\n", n.nim, n.nama, n.mataKuliah, n.sks, n.nilai);
    });
}

public static void cariNilai() {
    tampilNilai();
    System.out.print("Masukkan NIM Mahasiswa : ");
    String nim = sc.nextLine();
    boolean found = false;
    for (Nilai08 n : nilais) {
        if (n.nim.equals(nim)) {
            System.out.println("NIM          Nama          Mata
Kuliah          SKS          Nilai");
            System.out.printf("%-5s          %-10s          %-10s\n", n.nim, n.nama, n.mataKuliah, n.sks, n.nilai);
            found = true;
        }
    }
    if (!found) {
        System.out.println("Data tidak ditemukan");
    }
}

public static void sortData() {
    Nilai08[] nilaiArr = new Nilai08[nilais.size()];
    int i = 0;
    for (Nilai08 n : nilais) {
        nilaiArr[i] = n;
        i++;
    }
    for (int j = 0; j < nilaiArr.length - 1; j++) {
        for (int k = 0; k < nilaiArr.length - j - 1; k++) {
            if (nilaiArr[k].nilai < nilaiArr[k + 1].nilai) {
                Nilai08 temp = nilaiArr[k];
                nilaiArr[k] = nilaiArr[k + 1];
                nilaiArr[k + 1] = temp;
            }
        }
    }
    nilais.clear();
    for (Nilai08 n : nilaiArr) {
        nilais.add(n);
    }
}

public static void main(String[] args) {
    mahasiswa.add(new Mahasiswa08("20001", "Thalhah", "021xxx"));
    mahasiswa.add(new Mahasiswa08("20002", "Zubair", "021xxx"));
    mahasiswa.add(new Mahasiswa08("20003", "Abdul-Rahman", "021xxx"));
    mahasiswa.add(new Mahasiswa08("20004", "Sa'ad", "021xxx"));
    mahasiswa.add(new Mahasiswa08("20005", "Sa'id", "021xxx"));
}

```

```

        mahasiswa.add(new Mahasiswa08("20006", "Ubaidah", "021xxx"));
        mataKuliahs.add(new MataKuliah08("00001", "Internet Of Things",
"3"));
        mataKuliahs.add(new MataKuliah08("00002", "Algoritma dan Struktur
Data", "2"));
        mataKuliahs.add(new MataKuliah08("00003", "Algoritma dan
Pemograman", "2"));
        mataKuliahs.add(new MataKuliah08("00004", "Praktikum Algoritma dan
Struktur Data", "3"));
        mataKuliahs.add(new MataKuliah08("00005", "Praktikum Algoritma dan
Pemograman", "3"));
        int menu;
        do {
            menu();
            menu = sc.nextInt();
            sc.nextLine();
            switch (menu) {
                case 1:
                    inputNilai();
                    break;
                case 2:
                    tampilNilai();
                    break;
                case 3:
                    cariNilai();
                    break;
                case 4:
                    sortData();
                    break;
                case 5:
                    System.out.println("Exit...");
                    break;
                default:
                    System.out.println("Menu tidak ada");
            }
        }while (menu != 5);
    }
}

```

Hasil run

```
--- Masukkan Data ---
Masukkan nilai : 90
DAFTAR MAHASISWA
*****
NIM      Nama      Telf
20001    Thalhah    021xxx
20002    Zubair      021xxx
20003    Abdul-Rahman  021xxx
20004    Sa'ad        021xxx
20005    Sa'id        021xxx
20006    Ubaidah      021xxx
Masukkan NIM Mahasiswa : 20001
DAFTAR MATA KULIAH
*****
Kode     Mata Kuliah      SKS
00001    Internet Of Things    3
00002    Algoritma dan Struktur Data 2
00003    Algoritma dan Pemograman    2
00004    Praktikum Algoritma dan Struktur Data 3
00005    Praktikum Algoritma dan Pemograman    3
Masukkan Kode Mata Kuliah : 00003
*****
```

```
Pilih Menu: 2
DAFTAR NILAI MAHASISWA
*****
NIM      Nama      Mata Kuliah      SKS      Nilai
20001    Thalhah    Algoritma dan Pemograman    2      90.0
20002    Zubair      Internet Of Things          3      20002.0
*****
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
*****
```

```
NIM      Nama      Mata Kuliah      SKS      Nilai
20001    Thalhah    Algoritma dan Pemograman    2      90.0
20002    Zubair      Internet Of Things          3      20002.0
Masukkan NIM Mahasiswa : 20002
NIM      Nama      Mata Kuliah      SKS      Nilai
20002    Zubair      Internet Of Things          3      20002.0
*****
```

```
3. Keluar
*****
Pilih Menu: 4
*****
```

```
Pilih Menu: 5
Exit...
```

2. Kode tambahan hapusData

```
public static void hapusData() {
    tampilNilai();
    System.out.print("Masukkan NIM Mahasiswa : ");
    String nim = sc.nextLine();
    boolean found = false;
    for (Nilai08 n : nilais) {
        if (n.nim.equals(nim)) {
            nilais.remove(n);
            found = true;
        }
    }
    if (!found) {
```

```
        System.out.println("Data tidak ditemukan");  
    }  
}
```

Pada main
case 5:

```
        hapusData();  
        break;
```

Hasil run tamabahan

```
*****  
Pilih Menu: 5  
DAFTAR NILAI MAHASISWA  
*****  
NIM          Nama          Mata Kuliah          SKS          Nilai  
20002        Zubair        Internet Of Things    3            90.0  
Masukkan NIM Mahasiswa : 20002  
*****
```