

LAPORAN HASIL PRAKTIKUM
MATA KULIAH ALGORITMA DAN STRUKTUR DATA

PERTEMUAN 16 : COLLECTION



KAYLA RACHMAUDINA SATITI PUTRI

2341760103

D-IV SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI POLITEKNIK
NEGERI MALANG

2024

JOBSHEET 16

Collection

16.1. Tujuan Praktikum

Setelah melakukan praktikum ini, mahasiswa mampu:

1. memahami bentuk-bentuk collection dan hierarkinya;
2. menerapkan collection sesuai dengan fungsi dan jenisnya;
3. menyelesaikan kasus menggunakan collection yang sesuai.

16.2. Kegiatan Praktikum 1

16.2.1. Percobaan 1

Pada percobaan 1 ini akan dicontohkan penggunaan collection untuk menambahkan sebuah

elemen, mengakses elemen, dan menghapus sebuah elemen.

1. Buatlah sebuah class ContohList yang main method berisi kode program seperti di bawah ini

```
1 List<Object> l = new ArrayList<>();
2 l.add(1);
3 l.add(2);
4 l.add(3);
5 l.add("Cireng");
6 System.out.printf("Elemen 0: %d total elemen: %d elemen terakhir: %s\n", l.get(0), l.size(), l.get(l.size() - 1));
7
8 l.add(4);
9 l.remove(0);
10 System.out.printf("Elemen 0: %d total elemen: %d elemen terakhir: %s\n", l.get(0), l.size(), l.get(l.size() - 1));
```

2. Tambahkan kode program untuk menggunakan collection dengan aturan penulisan kode program seperti berikut

```
12 List<String> names = new LinkedList<>();
13 names.add("Noureen");
14 names.add("Akhleema");
15 names.add("Shannum");
16 names.add("Uwais");
17 names.add("Al-Qarni");
18
19 System.out.printf("Elemen 0: %s total elemen: %d elemen terakhir: %s\n",
20     names.get(0), names.size(), names.get(names.size() - 1));
21 names.set(0, "My kid");
22 System.out.printf("Elemen 0: %s total elemen: %d elemen terakhir: %s\n",
23     names.get(0), names.size(), names.get(names.size() - 1));
24 System.out.println("Names: " + names.toString());
```

16.2.2. Verifikasi Hasil Percobaan

```

Elemen 0: 1 total elemen: 4 elemen terakhir: Cireng
Elemen 0: 2 total elemen: 4 elemen terakhir: 4
Elemen 0: Nourreen total elemen: 5 elemen terakhir: Al-Qarni
Elemen 0: My kid total elemen: 5 elemen terakhir: Al-Qarni
Names: [My kid, Akhleema, Shannum, Uwais, Al-Qarni]

```

16.2.3. Pertanyaan Percobaan

1. Perhatikan baris kode 25-36, mengapa semua jenis data bisa ditampung ke dalam sebuah ArrayList?

Karena menggunakan tipe Object, yang merupakan superclass dari semua tipe di Java.

2. Modifikasi baris kode 25-36 sehingga data yang ditampung hanya satu jenis atau spesifik tipe tertentu!

```

1 List<String> l = new ArrayList<>();
2 l.add("1");
3 l.add("2");
4 l.add("3");
5 l.add("Cireng");
6 System.out.printf("Elemen 0: %s total elemen: %d elemen terakhir: %s\n", l.get(0), l.size(), l.get(l.size() - 1));
7
8 l.add("4");
9 l.remove(0);
10 System.out.printf("Elemen 0: %s total elemen: %d elemen terakhir: %s\n", l.get(0), l.size(), l.get(l.size() - 1));
11

```

3. Ubah kode pada baris kode 38 menjadi seperti ini

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

```
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());

```

```

names.push(e:"Mei-mei");
System.out.printf(format:"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!

```

Elemen 0: Mei-mei total elemen: 6 elemen terakhir: Al-Qarni
Names: [Mei-mei, My kid, Akhleema, Shannum, Uwais, Al-Qarni]

```

Kode program tambahan di atas berfungsi untuk menambahkan data ke list.

16.3. Kegiatan Praktikum 2

16.3.1. Tahapan Percobaan

Pada praktikum 2 ini akan dibuat beberapa method untuk menampilkan beberapa cara yang dapat dilakukan untuk mengambil/menampilkan elemen pada sebuah collection. Silakan ikutilah Langkah-langkah di bawah ini

1. Buatlah class dengan nama LoopCollection serta tambahkan method main yang isinya adalah sebagai berikut.

```
Stack<String> fruits = new Stack<>();
fruits.push(item:"Banana");
fruits.add(e:"orange");
fruits.add(e:"Watermelon");
fruits.add(e:"Leci");
fruits.push(item:"Salak");

for (String fruit : fruits) {
    System.out.printf(format:"%s ", fruit);
}
System.out.println("\n" + fruits.toString());

while (!fruits.empty()) {
    System.out.printf(format:"%s ", fruits.pop());
}
```

2. Tambahkan potongan kode berikut ini dari yang sebelumnya agar proses menampilkan elemen pada sebuah stack bervariasi.

```
fruits.push(item:"Melon");
fruits.push(item:"Durian");
System.out.println(x:"");
for (Iterator<String> it = fruits.iterator(); it.hasNext();) {
    String fruit = it.next();
    System.out.printf(format:"%s ", fruit);
}
System.out.println(x:"");
fruits.stream().forEach(e -> {
    System.out.printf(format:"%s ", e);
});
System.out.println(x:"");
for (int i = 0; i < fruits.size(); i++) {
    System.out.printf(format:"%s ", fruits.get(i));
}
```

16.3.2. Verifikasi Hasil Percobaan

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

16.3.3. Pertanyaan Percobaan

1. Apakah perbedaan fungsi `push()` dan `add()` pada objek *fruits*?

`Push()` menambahkan elemen ke bagian atas list

`Add()` menambahkan elemen ke ahir list.

2. Silakan hilangkan baris 43 dan 44, apakah yang akan terjadi? Mengapa bisa demikian?

Tidak akan ada elemen yang ditambahkan ke stack setelah stack dikosongkan oleh loop `while (!fruits.empty())`. Akibatnya, iterasi menggunakan iterator dan stream akan tidak mencetak apa-apa karena stack *fruits* akan kosong.

3. Jelaskan fungsi dari baris 46-49?

Iterasi ini digunakan untuk mencetak setiap elemen di stack setelah elemen "Melon" dan "Durian" ditambahkan.

4. Silakan ganti baris kode 25, *Stack<String>* menjadi *List<String>* dan apakah yang terjadi?

Tidak ada yang berubah

Mengapa bisa demikian?

karena Stack adalah subkelas dari Vector dan mengimplementasikan List

5. Ganti elemen terakhir dari objek *fruits* menjadi "Strawberry"!

```
// Mengganti elemen terakhir menjadi "Strawberry"
fruits.set(fruits.size() - 1, element:"Strawberry");
```

Melon Strawberry

6. Tambahkan 3 buah seperti "Mango", "guava", dan "avocado" kemudian dilakukan sorting!

```
// Menambahkan tiga buah baru
fruits.add(e:"Mango");
fruits.add(e:"guava");
fruits.add(e:"avocado");

// Melakukan sorting
Collections.sort(fruits);

// Menampilkan elemen setelah sorting
System.out.println(x:"\nSetelah Sorting:");
for (String fruit : fruits) {
    System.out.printf(format:"%s ", fruit);
}
```

Setelah Sorting:
Mango Melon Strawberry avocado guava

16.4. Kegiatan Praktikum 3

16.4.1. Tahapan Percobaan

Pada praktikum 3 ini dilakukan uji coba untuk mengimplementasikan sebuah collection untuk menampung objek yang dibuat sesuai kebutuhan. Objek tersebut adalah sebuah objek mahasiswa dengan fungsi-fungsi umum seperti menambahkan, menghapus, mengubah, dan mencari.

1. Buatlah sebuah class Mahasiswa dengan attribute, konstruktor, dan fungsi.

```
String nim;  
String nama;  
String notelp;  
  
public Mahasiswa15() {}  
  
public Mahasiswa15(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 + '}';  
}
```

2. Selanjutnya, buatlah sebuah class ListMahasiswa

```
List<Mahasiswa15> mahasiswa15s = new ArrayList<>();
```

3. Method **tambah()**, **hapus()**, **update()**, dan **tampil()**.

```
public void tambah(Mahasiswa15 ... mahasiswa) {  
    mahasiswa15s.addAll(Arrays.asList(mahasiswa));  
}  
  
public void hapus(int index) {  
    mahasiswa15s.remove(index);  
}  
  
public void update(int index, Mahasiswa15 mhs) {  
    mahasiswa15s.set(index, mhs);  
}  
  
public void tampil() {  
    mahasiswa15s.stream().forEach(mhs -> {  
        System.out.println(mhs.toString());  
    });  
}
```

4. Untuk proses hapus, update membutuhkan fungsi pencarian terlebih dahulu

```
int linearSearch(String nim) {  
    for (int i = 0; i < mahasiswa15s.size(); i++) {  
        if (nim.equals(mahasiswa15s.get(i).nim)) {  
            return i;  
        }  
    }  
    return -1;  
}
```

5. Pada class yang sama, tambahkan main method

```
Mahasiswa15 lm = new Mahasiswa15();
Mahasiswa15 m = new Mahasiswa15(nim:"201234", nama:"Noureen", notelp:"021xx1");
Mahasiswa15 m1 = new Mahasiswa15(nim:"201235", nama:"Akhleema", notelp:"021xx2");
Mahasiswa15 m2 = new Mahasiswa15(nim:"201236", nama:"Shannum", notelp:"021xx3");
// menambahkan objek mahasiswa
lm.tambah(m, m1, m2);
// menampilkan list mahasiswa
lm.tampil();
// update mahasiswa
lm.update(lm.linearSearch(nim:"201235"), new Mahasiswa15(nim:"201235", nama:"Akhleema Lela", notelp:"021xx2"));
System.out.println(x);
lm.tampil();
```

16.4.2. Verifikasi Hasil Percobaan

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

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

16.4.3. Pertanyaan Percobaan

1. Pada fungsi tambah() yang menggunakan unlimited argument itu menggunakan konsep apa? Dan kelebihan apa?

Menggunakan varargs yang memungkinkan untuk menerima sejumlah argumen yang tidak terbatas dalam bentuk array. Kelebihannya memungkinkan kita untuk memanggil fungsi dengan sejumlah argumen yang berbeda tanpa perlu mendefinisikan beberapa metode dengan jumlah parameter yang berbeda.

2. Pada fungsi linearSearch() di atas, silakan diganti dengan fungsi binarySearch() dari collection!

```
// mengganti linear search dengan binary search
int binarySearch(String nim) {
    List<String> nims = new ArrayList<>();
    for (Mahasiswa15 m : mahasiswa15s) {
        nims.add(m.nim);
    }
    Collections.sort(nims);
    return Collections.binarySearch(nims, nim);
}
```

3. Tambahkan fungsi sorting baik secara ascending ataupun descending pada class tersebut!

```
public void sortAscending() {
    mahasiswa15s.sort((m1, m2) -> m1.nim.compareTo(m2.nim));
}

public void sortDescending() {
    mahasiswa15s.sort((m1, m2) -> m2.nim.compareTo(m1.nim));
}
```

Main class:

```
// Sorting ascending
lm.sortAscending();
System.out.println(x:"\nSetelah sorting ascending:");
lm.tampil();

// Sorting descending
lm.sortDescending();
System.out.println(x:"\nSetelah sorting descending:");
lm.tampil();
```

Output:

```
Setelah sorting ascending:
Mahasiswa{nim=201234, nama=Noureen, notelp=021xx1}
Mahasiswa{nim=201235, nama=Akhleema Lela, notelp=021xx2}
Mahasiswa{nim=201236, nama=Shannum, notelp=021xx3}

Setelah sorting descending:
Mahasiswa{nim=201236, nama=Shannum, notelp=021xx3}
Mahasiswa{nim=201235, nama=Akhleema Lela, notelp=021xx2}
Mahasiswa{nim=201234, nama=Noureen, notelp=021xx1}
```

16.5. Tugas Praktikum

1. Buatlah implementasi program daftar nilai mahasiswa semester, minimal memiliki 3 class yaitu Mahasiswa, Nilai, dan Mata Kuliah. Data Mahasiswa dan Mata Kuliah perlu melalui penginputan data terlebih dahulu.

Class mahasiswa

```
public class Mahasiswa {
    String nim;
    String nama;
    String telf;

    Mahasiswa(String nim, String nama, String telf) {
        this.nim = nim;
        this.nama = nama;
        this.telf = telf;
    }
}
```



```

public class Mahasiswa {
    Run | Debug
    public static void main(String[] args) {
        SistemNilai sistem = new SistemNilai();
        Scanner input = new Scanner(System.in);
        int pilihan;

        // insert data mahasiswa
        sistem.tambahMahasiswa(nim:"001", nama:"abdur", telf:"021xxx");
        sistem.tambahMahasiswa(nim:"002", nama:"ikram", telf:"021xxx");
        sistem.tambahMahasiswa(nim:"003", nama:"rahman", telf:"021xxx");
        sistem.tambahMahasiswa(nim:"004", nama:"ghofur", telf:"021xxx");
        sistem.tambahMahasiswa(nim:"005", nama:"ibda", telf:"021xxx");
        sistem.tambahMahasiswa(nim:"006", nama:"asbak", telf:"021xxx");

        sistem.tambahMataKuliah(kode:"01", nama:"Internet of Things", sks:3);
        sistem.tambahMataKuliah(kode:"02", nama:"Algoritma dan Struktur Data", sks:3);
        sistem.tambahMataKuliah(kode:"03", nama:"Algoritma dan Pemrograman", sks:3);
        sistem.tambahMataKuliah(kode:"04", nama:"Praktikum Algoritma dan Struktur Data");
        sistem.tambahMataKuliah(kode:"05", nama:"Praktikum Algoritma dan Pemrograman",);

        do {
            System.out.println();
            System.out.println(x:"*****");
            System.out.println(x:" SISTEM PENGOLAHAN DATA NILAI MAHASISWA ");
            System.out.println(x:"*****");
            System.out.println(x:"1. Input Nilai");
            System.out.println(x:"2. Tampil Nilai");
            System.out.println(x:"3. Mencari Nilai Mahasiswa");
            System.out.println(x:"4. Urut Data Nilai");
            System.out.println(x:"5. Keluar");
            System.out.println(x:"*****");
            System.out.print(s:"Pilih: ");
            pilihan = input.nextInt();
            input.nextLine();
        } while (pilihan < 1 || pilihan > 5);
    }
}

```

```

        switch (pilihan) {
            case 1:
                System.out.print(s:"Masukkan NIM: ");
                String nim = input.nextLine();
                System.out.print(s:"Masukkan Kode Mata Kuliah: ");
                String kodeMk = input.nextLine();
                System.out.print(s:"Masukkan Nilai: ");
                double nilai = input.nextDouble();
                sistem.inputNilai(nim, kodeMk, nilai);
                break;
            case 2:
                sistem.tampilNilai();
                break;
            case 3:
                System.out.print(s:"Masukkan NIM: ");
                nim = input.nextLine();
                sistem.cariNilaiMahasiswa(nim);
                break;
            case 4:
                sistem.urutDataNilai();
                break;
            case 5:
                return;
            default:
                System.out.println(x:"Pilih antara 1-7.");
        }
    } while (pilihan != 5);
}
}

```

Class matakuliah

```

public class MataKuliah {
    String kode;
    String nama;
    int sks;

    MataKuliah(String kode, String nama, int sks) {
        this.kode = kode;
        this.nama = nama;
        this.sks = sks;
    }
}

```

Class nilai

```

public class Nilai {
    Mahasiswa mahasiswa;
    MataKuliah matakuliah;
    double nilai;

    Nilai(Mahasiswa mahasiswa, MataKuliah matakuliah, double nilai) {
        this.mahasiswa = mahasiswa;
        this.matakuliah = matakuliah;
        this.nilai = nilai;
    }
}

```

Class sistem nilai

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

public class SistemNilai {
    List<Mahasiswa> daftarMahasiswa = new ArrayList<>();
    List<MataKuliah> daftarMataKuliah = new ArrayList<>();
    List<Nilai> daftarNilai = new ArrayList<>();
    Queue<Mahasiswa> antrianHapus = new LinkedList<>();

    void tambahMahasiswa(String nim, String nama, String telf) {
        Mahasiswa mahasiswa = new Mahasiswa(nim, nama, telf);
        daftarMahasiswa.add(mahasiswa);
    }

    void tambahMataKuliah(String kode, String nama, int sks) {
        MataKuliah matakuliah = new MataKuliah(kode, nama, sks);
        daftarMataKuliah.add(matakuliah);
    }

    void inputNilai(String nim, String kodeMk, double nilai) {
        Mahasiswa mahasiswa = daftarMahasiswa.stream().filter(m -> m.nim.equals(nim)).findFirst().orElse(null);
        MataKuliah matakuliah = daftarMataKuliah.stream().filter(mk -> mk.kode.equals(kodeMk)).findFirst().orElse(null);

        if (mahasiswa != null && matakuliah != null) {
            Nilai nilaiObj = new Nilai(mahasiswa, matakuliah, nilai);
            daftarNilai.add(nilaiObj);
        } else {
            if (mahasiswa == null) {
                System.out.println("Mahasiswa dengan NIM " + nim + " tidak ditemukan.");
            } else {
                System.out.println("Mata kuliah dengan kode " + kodeMk + " tidak ditemukan.");
            }
        }
    }

    void tampilNilai() {
        System.out.println("*****");
        System.out.println("          DAFTAR NILAI MAHASISWA          ");
        System.out.println("*****");
        for (Nilai nilai : daftarNilai) {
            System.out.println("NIM           : " + nilai.mahasiswa.nim);
            System.out.println("Nama           : " + nilai.mahasiswa.nama);
            System.out.println("Mata Kuliah    : " + nilai.matakuliah.nama);
            System.out.println("SKS           : " + nilai.matakuliah.sks);
            System.out.println("Nilai         : " + nilai.nilai);
            System.out.println("*****");
        }
    }

    void cariNilaiMahasiswa(String nim) {
        System.out.println("*****");
        System.out.println("          NILAI MAHASISWA NIM: " + nim);
        System.out.println("*****");
        for (Nilai nilai : daftarNilai) {
            if (nilai.mahasiswa.nim.equals(nim)) {
                System.out.println("Nama           : " + nilai.mahasiswa.nama);
                System.out.println("Mata Kuliah    : " + nilai.matakuliah.nama);
                System.out.println("SKS           : " + nilai.matakuliah.sks);
                System.out.println("Nilai         : " + nilai.nilai);
                System.out.println("*****");
            }
        }
    }

    void urutDataNilai() {
        daftarNilai.sort(Comparator.comparing(n -> n.mahasiswa.nama));
        tampilNilai();
    }
}
```

```

*****
SISTEM PENGOLAHAN DATA NILAI MAHASISWA
*****
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
*****
Pilih: 1
Masukkan NIM: 002
Masukkan Kode Mata Kuliah: 01
Masukkan Nilai: 95

*****
SISTEM PENGOLAHAN DATA NILAI MAHASISWA
*****
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
*****
Pilih: 2
*****
DAFTAR NILAI MAHASISWA
*****
NIM      : 005
Nama     : ibda
Mata Kuliah : Praktikum Algoritma dan Pemrograman
SKS      : 3
Nilai    : 90.0
*****
NIM      : 002
Nama     : ikram
Mata Kuliah : Internet of Things
SKS      : 3
Nilai    : 95.0
*****

```

```

Pilih: 3
Masukkan NIM: 005
*****
NILAI MAHASISWA NIM: 005
*****
Nama      : ibda
Mata Kuliah : Praktikum Algoritma dan Pemrograman
SKS       : 3
Nilai     : 90.0
*****

*****
SISTEM PENGOLAHAN DATA NILAI MAHASISWA
*****
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
*****
Pilih: 4
*****
DAFTAR NILAI MAHASISWA
*****
NIM      : 005
Nama     : ibda
Mata Kuliah : Praktikum Algoritma dan Pemrograman
SKS      : 3
Nilai    : 90.0
*****
NIM      : 002
Nama     : ikram
Mata Kuliah : Internet of Things
SKS      : 3
Nilai    : 95.0
*****

```

2. Tambahkan prosedur hapus data mahasiswa melalui implementasi Queue pada collections Tugas nomor 1!

```

void hapusMahasiswa(String nim) {
    Mahasiswa mahasiswa = daftarMahasiswa.stream().filter(m -> m.nim.equals(nim)).findFirst().orElse(null);
    if (mahasiswa != null) {
        daftarMahasiswa.remove(mahasiswa);
        daftarNilai.removeIf(nilai -> nilai.mahasiswa.equals(mahasiswa));
        System.out.println("Mahasiswa dengan NIM " + mahasiswa.nim + " telah dihapus.");
    } else {
        System.out.println("Mahasiswa dengan NIM " + nim + " tidak ditemukan.");
    }
}

```

```
SISTEM PENGOLAHAN DATA NILAI MAHASISWA
*****
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Hapus Mahasiswa
6. Keluar
*****
Pilih: 5
Masukkan NIM Mahasiswa yang akan dihapus: 001
Mahasiswa dengan NIM 001 telah dihapus.
```