LAPORAN PRAKTIKUM
MATA KULIAH ALGORITMA DAN STRUKTUR DATA

Dosen Pengampu : Triana Fatmawati, S.T, M.T

PERTEMUAN - 15 - Collection

Nama : **M. Zidna Billah Faza**

NIM : **2341760030**

Prodi : **D-IV Sistem Informasi Bisnis**

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2024

Percobaan 1

1) Buatlah sebuah class ContohList yang main methode berisi kode program seperti di bawah ini

```
List l = new ArrayList<>();

l.add(e:1);
l.add(e:2);
l.add(e:3);
l.add(e:"Cireng");

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

l.add(e:4);
l.remove(index:0);

System.out.printf(format:"Elemen 0 : %d total elemen : %d elemen terakhir : %s\n", l.get(index:0), l.size(),
        l.get(l.size() - 1));
```

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

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

names.add(e:"Noureen");
names.add(e:"Akhleema");
names.add(e:"Shannum");
names.add(e:"Uwais");
names.add(e:"Al-Qarni");

System.out.printf(format:"Elemen 0 : %s total elemen : %s elemen terakhir : %s\n", names.get(index:0), names.size(),
        names.get(names.size() - 1));

names.set(index:0, element:"My Kid");

System.out.printf(format:"Elemen 0 : %s total elemen : %s elemen terakhir : %s\n", names.get(index:0), names.size(),
        names.get(names.size() - 1));
System.out.println("Names : " + names.toString());
```

3) Verifikasi Hasil Percobaan

```
Elemen 0 : 1 total elemen : 4 elemen terakhir : Cireng
Elemen 0 : 2 total elemen : 4 elemen terakhir : 4
Elemen 0 : Noureen 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]
```

Pertanyaan Percobaan 1

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

Karena pada deklarasi ArrayList() tidak disebutkan parameter yang spesifik maka ketika mengisi ArrayList() dapat berupa banyak tipe data

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

```
List<Integer> LI = new ArrayList<>();

LI.add(e:100);
LI.add(e:200);
LI.add(e:300);

System.out.printf(format:"\nElemen 0 : %d total elemen : %d elemen terakhir : %s\n", LI.get(index:0), LI.size(),
        LI.get(LI.size() - 1));
```

```
Elemen 0 : 100 total elemen : 3 elemen terakhir : 300
```

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(e:"Mei-mei");

System.out.printf(format:"Elemen 0 : %s total elemen : %s elemen terakhir : %s\n", names.get(index:0), names.size(),
        names.get(names.size() - 1));
System.out.println("Names : " + names.toString());
```

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

Baris kode 38 telah diubah untuk menggunakan LinkedList<String> untuk objek names. Kemudian, baris kode 39 ditambahkan untuk memasukkan string "Mei-mei" ke dalam LinkedList menggunakan metode push().

Selanjutnya, pada baris kode 40, printf() dipanggil untuk mencetak elemen pertama (getFirst()), ukuran total (size()), dan elemen terakhir (getLast()) dari LinkedList names. Dan akhirnya, pada baris kode 41, toString() dipanggil untuk mencetak LinkedList names.

Percobaan 2

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.add(e:"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());
}

fruits.push(item:"Melon");
fruits.push(item:"Durian");
```

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));
}
```

3) Verifikasi Hasil Percobaan

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

Pertanyaan Percobaan 2

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

add() lebih cocok digunakan untuk menambahkan elemen kedalam *stack* dengan konsep LIFO, sedangkan push() lebih cocok digunakan untuk menambahkan elemen kedalam *list* dengan tidak memiliki konsep LIFO

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

Program tidak mencetak apapun karena tidak ada elemen yang ada di dalam Stack karena tidak ada elemen di dalam Stack

3) Jelaskan fungsi dari baris 46-49?

Kode tersebut berfungsi untuk mencetak semua elemen didalam stack dengan mengggunakan iterator, maka ketika iterator menemukan elemen didalma stack maka akan di print

4) Silakan ganti baris kode 25, Stack<String> menjadi List<String> dan apakah yang terjadi? Mengapa bisa demikian?

Maka akan terjadi error pada push() karena pada List tidak memiliki push() maka dapat diganti dengan add()

5) Ganti elemen terakhir dari dari objek fruits menjadi “Strawberry”!

```
fruits.pop();
fruits.add(e:"Strawberry");
```
```
Banana Orange Watermelon Leci Strawberry
[Banana, Orange, Watermelon, Leci, Strawberry]
```

6) Tambahkan 3 buah seperti “Mango”,”guava”, dan “avocado” kemudian dilakukan sorting!

```
fruits.add(e:"Mango");
fruits.add(e:"Guava");
fruits.add(e:"Avocado");

Collections.sort(fruits);
```
```
Avocado Banana Guava Leci Mango Orange Strawberry Watermelon
[Avocado, Banana, Guava, Leci, Mango, Orange, Strawberry, Watermelon]
```

Percobaan 3

1) Buatlah sebuah class Mahasiswa dengan attribute, kontruktor, dan fungsi sebagai berikut.

```
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 + ", No Telp = " + noTelp + '}';
}
```

2) Selanjutnya, buatlah sebuah class ListMahasiswa yang memiliki attribute seperti di bawah ini

```
List<Mahasiswa> mahasiswas = new ArrayList<>();
```

3) Method tambah(), hapus(), update(), dan tampil()secara berurut dibuat agar bisa melakukan operasi-operasi seperti yang telah disebutkan.

```
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) {
    mahasiswas.set(index, mhs);
}

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

4) Untuk proses hapus, update membutuhkan fungsi pencarian terlebih dahulu yang potongan kode programnya adalah sebagai berikut

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

5) Pada class yang sama, tambahkan main method seperti potongan program berikut dan amati hasilnya!

```
ListMahasiswa lm = new ListMahasiswa();

Mahasiswa m = new Mahasiswa(nim:"201234", nama:"Noureen", noTelp:"021xx1");
Mahasiswa m1 = new Mahasiswa(nim:"201235", nama:"Akleema", noTelp:"021xx2");
Mahasiswa m2 = new Mahasiswa(nim:"201236", nama:"Shannum", noTelp:"021xx3");

lm.tambah(m, m1, m2);
lm.tampil();
lm.update(lm.linearSearch(nim:"201235"), new Mahasiswa(nim:"201235", nama:"Akhleema Lela", noTelp:"021xx2"));

System.out.println(x:"");
lm.tampil();
```

6) Verifikasi Hasil Percobaan

```
Mahasiswa{ Nim = 201234, Nama = Noureen, No Telp = 021xx1}
Mahasiswa{ Nim = 201235, Nama = Akleema, No Telp = 021xx2}
Mahasiswa{ Nim = 201236, Nama = Shannum, No Telp = 021xx3}

Mahasiswa{ Nim = 201234, Nama = Noureen, No Telp = 021xx1}
Mahasiswa{ Nim = 201235, Nama = Akhleema Lela, No Telp = 021xx2}
Mahasiswa{ Nim = 201236, Nama = Shannum, No Telp = 021xx3}
```

Pertanyaan Percobaan 3

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

Pada fungsi tambah() menggunakan konsep varargs yang mempermudah penambahan objek Mahasiswa ke dalam list mahasiswas tanpa harus membatasi jumlah argumen yang diberikan saat pemanggilan metode.

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

```
int binarySearch(String nim) {

    Collections.sort(mahasiswas, (m1, m2) -> m1.nim.compareTo(m2.nim));

    int low = 0;
    int high = mahasiswas.size() - 1;

    while (low <= high) {
        int mid = (low + high) / 2;
        String midNim = mahasiswas.get(mid).nim;
        int cmp = midNim.compareTo(nim);

        if (cmp < 0) {
            low = mid + 1;
        } else if (cmp > 0) {
            high = mid - 1;
        } else {
            return mid;
        }
    }

    return -1;
}
```

```
lm.update(lm.binarySearch(nim:"201235"), new Mahasiswa(nim:"201235", nama:"Akhleema Lela", noTelp:"021xx2"));
```

```
Mahasiswa{ Nim = 201235, Nama = Akhleema Lela, No Telp = 021xx2}
```

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

Sorting berdasarkan NIM

```
public void ascendingSort() {
    Collections.sort(mahasiswas, Comparator.comparing(m -> m.nim));
}

public void descendingSort() {
    Collections.sort(mahasiswas, Comparator.comparing((Mahasiswa m) -> m.nim).reversed());
}
```

```
System.out.println(x:"Ascending Sort:");
lm.ascendingSort();
lm.tampil();

System.out.println(x:"\nDescending Sort:");
lm.descendingSort();
lm.tampil();
```

```
Ascending Sort:
Mahasiswa{ Nim = 201234, Nama = Noureen, No Telp = 021xx1}
Mahasiswa{ Nim = 201235, Nama = Akhleema Lela, No Telp = 021xx2}
Mahasiswa{ Nim = 201236, Nama = Shannum, No Telp = 021xx3}

Descending Sort:
Mahasiswa{ Nim = 201236, Nama = Shannum, No Telp = 021xx3}
Mahasiswa{ Nim = 201235, Nama = Akhleema Lela, No Telp = 021xx2}
Mahasiswa{ Nim = 201234, Nama = Noureen, No Telp = 021xx1}
```

Tugas Praktikum 1

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

- Class Matakuliah

```
public class Matakuliah {

    String kode;
    int sks;
    String namaMatkul;

    public Matakuliah(String kode, String namaMatkul, int sks) {
        this.kode = kode;
        this.sks = sks;
        this.namaMatkul = namaMatkul;
    }

    public String getKode() {
        return kode;
    }

    public String getNamaMatkul() {
        return namaMatkul;
    }

    public int getSks() {
        return sks;
    }

    public String toString() {
        return "| " + kode + "\t| " + namaMatkul + "\t\t\t| " + sks + "\t|";
    }

}
```

- Class Mahasiswa

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

public class Mahasiswa {

    String nim;
    String nama;
    String noTelp;
    List<Nilai> nilaiList;

    public Mahasiswa() {

    }

    public Mahasiswa(String nim, String nama, String noTelp) {
        this.nim = nim;
        this.nama = nama;
        this.noTelp = noTelp;
        nilaiList = new ArrayList<>();
    }

    @Override

    public String toString() {
        return "| " + nim + "\t\t\t| " + nama + "\t\t\t| " + noTelp + "\t\t|";
    }

    public String getNim() {
        return nim;
    }

    public String getNama() {
        return nama;
    }

    public void tambahNilai(Nilai nilai) {
        nilaiList.add(nilai);
    }

    public List<Nilai> getNilaiList() {
        return nilaiList;
    }

}
```

- Class Nilai

```
public class Nilai {
    Mahasiswa mahasiswa;
    Matakuliah matakuliah;
    double nilai;

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

    public Mahasiswa getMahasiswa() {
        return mahasiswa;
    }

    public Matakuliah getMatakuliah() {
        return matakuliah;
    }

    public double getNilai() {
        return nilai;
    }

    @Override

    public String toString() {
        return "| " + mahasiswa.getNim() + "\t| " + mahasiswa.getNama() + "\t\t| " +
                matakuliah.getNamaMatkul() + "\t| " + matakuliah.getSks() + "\t| " +
                nilai + "\t|";
    }
}
```

- Output Input Nilai

```
========================================================================
                SISTEM PENGOLAHAN DATA NILAI MAHASISWA
========================================================================

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Hapus Mahasiswa
6. Keluar

========================================================================
Masukkan pilihan          : 1
-----------------------------------------------------------
| kode  | Nama Matkul                          | SKS   |
-----------------------------------------------------------
| IOT   | Internet of Things                   | 3     |
| ASD   | Algortima Data                       | 2     |
| BSD   | Teori Basis Data                     | 2     |
| PSI   | Sistem Informasi                     | 2     |
-----------------------------------------------------------
Masukkan Kode Matakuliah    : PSI
-------------------------------------------------------------------------------------
| NIM                  | Nama Mahasiswa               | Nomor                |
-------------------------------------------------------------------------------------
| 2342                 | Slamet                       | 08123457             |
| 2343                 | Kurniawan                    | 08123458             |
| 2344                 | Riyatno                      | 08123459             |
-------------------------------------------------------------------------------------
Masukkan NIM Mahasiswa      : 2342
Masukkan Nilai              : 89
Nilai berhasil diinputkan untuk Mahasiswa Slamet
```

- Output Tampil Nilai

```
Masukkan pilihan          : 2
-------------------------------------------------------------------------------
| NIM                   | Nama Mahasiswa                | Nomor                 |
-------------------------------------------------------------------------------
| 2342                  | Slamet                        | 08123457              |
| 2343                  | Kurniawan                     | 08123458              |
| 2344                  | Riyatno                       | 08123459              |
-------------------------------------------------------------------------------
Masukkan NIM Mahasiswa      : 2342
-------------------------------------------------------------------------
| NIM   | Nama Mahasiswa        | Mata Kuliah           | SKS   | Nilai |
-------------------------------------------------------------------------
| 2342  | Slamet                | Internet of Things    | 3     | 82.0  |
| 2342  | Slamet                | Algortima Data        | 2     | 87.0  |
| 2342  | Slamet                | Sistem Informasi      | 2     | 89.0  |
-------------------------------------------------------------------------
```

- Output Mencari Nilai Mahasiswa

```
Masukkan pilihan          : 3
-------------------------------------------------------------------------------
| NIM                   | Nama Mahasiswa                | Nomor                 |
-------------------------------------------------------------------------------
| 2342                  | Slamet                        | 08123457              |
| 2343                  | Kurniawan                     | 08123458              |
| 2344                  | Riyatno                       | 08123459              |
-------------------------------------------------------------------------------
Masukkan NIM Mahasiswa      : 2342
-------------------------------------------------------------------------
| NIM   | Nama Mahasiswa        | Mata Kuliah           | SKS   | Nilai |
-------------------------------------------------------------------------
| 2342  | Slamet                | Internet of Things    | 3     | 82.0  |
| 2342  | Slamet                | Algortima Data        | 2     | 87.0  |
| 2342  | Slamet                | Sistem Informasi      | 2     | 89.0  |
-------------------------------------------------------------------------
```

- Output Urut Data Nilai

```
Masukkan pilihan          : 4
-------------------------------------------------------------------------
| NIM   | Nama Mahasiswa        | Mata Kuliah           | SKS   | Nilai |
-------------------------------------------------------------------------
| 2344  | Riyatno               | Algortima Data        | 2     | 90.0  |
| 2342  | Slamet                | Sistem Informasi      | 2     | 89.0  |
| 2344  | Riyatno               | Teori Basis Data      | 2     | 88.0  |
| 2342  | Slamet                | Algortima Data        | 2     | 87.0  |
| 2343  | Kurniawan             | Internet of Things    | 3     | 85.0  |
| 2344  | Riyatno               | Internet of Things    | 3     | 83.0  |
| 2342  | Slamet                | Internet of Things    | 3     | 82.0  |
-------------------------------------------------------------------------
```

- Output Hapus Mahasiswa

```
Masukkan pilihan          : 5
-------------------------------------------------------------------------------
| NIM                   | Nama Mahasiswa                | Nomor               |
-------------------------------------------------------------------------------
| 2342                  | Slamet                        | 08123457            |
| 2343                  | Kurniawan                     | 08123458            |
| 2344                  | Riyatno                       | 08123459            |
-------------------------------------------------------------------------------
Masukkan NIM Mahasiswa      : 2344
Mahasiswa dengan NIM 2344 berhasil dihapus.
```

```
-------------------------------------------------------------------------------
| NIM                   | Nama Mahasiswa                | Nomor               |
-------------------------------------------------------------------------------
| 2342                  | Slamet                        | 08123457            |
| 2343                  | Kurniawan                     | 08123458            |
-------------------------------------------------------------------------------
```

**full code ada di github*

Repository : https://github.com/zidnafaz/Praktikum-Algoritma-Struktur-Data