

TUGAS PRATIKUM 12

(Sebagai pemenuhan salah satu tugas mata kuliah

Pemograman Berorientasi Objek(Pratikum) Program Studi D3 Teknik Informatika)



Disusun oleh :

Rani Indrianna Sembiring (201511025)

Kelas : 2A

POLITEKNIK NEGERI BANDUNG

JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA

PRODI D3 TEKNIK INFORMATIKA

2021 – 2022

1. Buatlah class yang menggambarkan struktur data yang dapat menampung data-data pada file json tersebut, lalu gunakan Collection Framework dengan tipe LIST untuk menampung data tersebut.

```
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.ArrayList;

import com.google.gson.Gson;
import com.google.gson.JsonIOException;
import com.google.gson.JsonSyntaxException;
import com.google.gson.reflect.TypeToken;

public class JsonToObject {
    public static void main(String[] args) {
        Gson gson = new Gson();
        try {
            ArrayList<Barang> data = gson.fromJson(
                new FileReader("D:\\TUGAS PBO\\W12_PBO_PR\\src\\JavaCollectionFramework\\barang.json"),
                new TypeToken<ArrayList<Barang>>() {}.getType()
            );
            for(int x = 0; x<data.size(); x++) {
                System.out.println("Barang ke-" + (x+1) + " = " + data.get(x).toString());
            }
        } catch (JsonIOException | JsonSyntaxException | FileNotFoundException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

2. Buatlah aplikasi untuk menampilkan daftar produk yang akan dikirimkan dengan format sebagai berikut

ProductName | Quantity | Weight | Destination | ServicePackage | Service | Value | Total

```
@Override
public String toString() {
    String result = this.productName + " | " + this.quantity + " | "
        + this.weight + " | " + city.getDestination() + " | " +
        servicePackage.getService() + " | " + servicePackage.getValue() + " | " +
        ((priceItem*quantity)+servicePackage.getValue());
    return result;
}
```

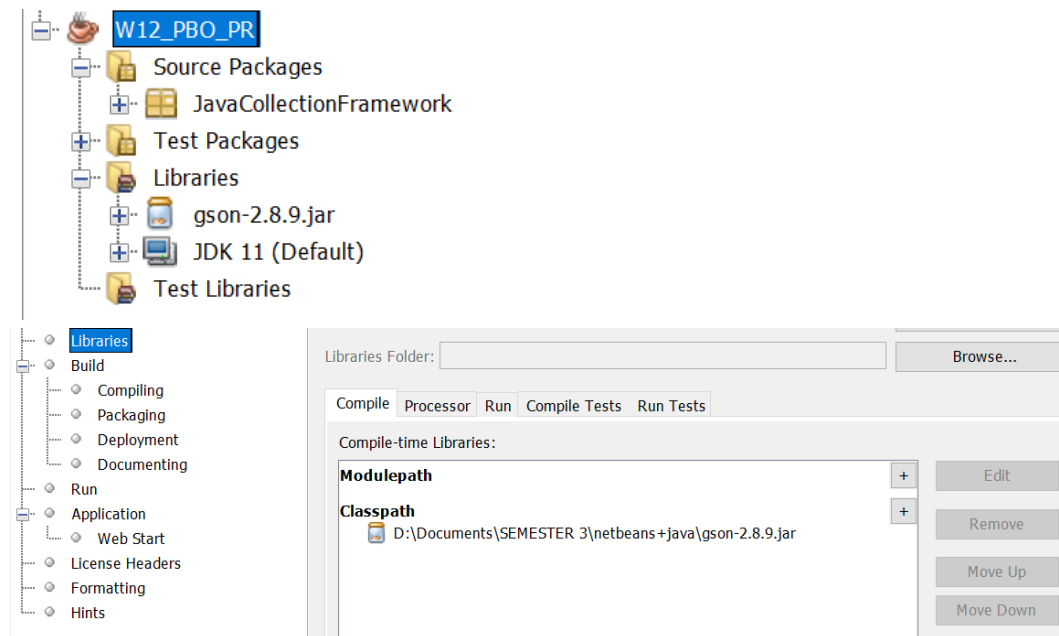
➤ Hasil Output

compile-single:

run-single:

```
Barang ke-1 = Celana Jeans | 1 | 2 | JAKARTA | Yakin Esok Sampai(YES) | 48000 | 128000
Barang ke-2 = Kemeja Lengan Panjang | 2 | 2 | JAKARTA | Regular (REG) | 26000 | 206000
BUILD SUCCESSFUL (total time: 1 second)
```

3. Gunakan library Json untuk melakukan parsing data dari JSON to Object



A. Bagaimana Struktur Class yang dibuat untuk menampung data dari file json?

Berikut class yang digunakan:

- Barang.java

```
public class Barang {  
    private Status status;  
    private ServicePackage servicePackage;  
    private City city;  
    private int weight, quantity, priceItem;  
    private String productName;  
  
    @Override  
    public String toString() {  
        String result = this.productName + " | " + this.quantity + " | "  
            + this.weight + " | " + city.getDestination() + " | " +  
            servicePackage.getService() + " | " + servicePackage.getValue() + " | "  
            + ((priceItem*quantity)+servicePackage.getValue());  
        return result;  
    }  
}
```

- City.java

```
public class City {  
    private String origin;  
    private String destination;  
  
    public String getOrigin() {  
        return origin;  
    }  
  
    public void setOrigin(String origin) {  
        this.origin = origin;  
    }  
  
    public String getDestination() {  
        return destination;  
    }  
  
    public void setDestination(String destination) {  
        this.destination = destination;  
    }  
}
```

- ServicePackage.java

```
public class ServicePackage {  
    private String service;  
    private int value;  
  
    public String getService() {  
        return service;  
    }  
  
    public void setService(String service) {  
        this.service = service;  
    }  
  
    public int getValue() {  
        return value;  
    }  
  
    public void setValue(int value) {  
        this.value = value;  
    }  
}
```

▪ Status.java

```
public class Status {
    private int code;
    private String description;

    public int getCode() {
        return code;
    }

    public void setCode(int code) {
        this.code = code;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }
}
```

▪ barang.json

```
[{
    "status": {
        "code": 0,
        "description": "OK"
    },
    "servicePackage": {
        "service": "Yakin Esok Sampai (YES)",
        "value": 48000
    },
    "city": {
        "origin": "SURABAYA",
        "destination": "JAKARTA"
    },
    "weight": 2,
    "quantity": 1,
    "priceItem": 80000,
    "productName": "Celana Jeans"
},
{
    "status": {
        "code": 0,
        "description": "OK"
    },
    "servicePackage": {
        "service": "Reguler (REG)",
        "value": 26000
    },
    "city": {
        "origin": "SURABAYA",
        "destination": "JAKARTA"
    },
    "weight": 2, "quantity": 2,
    "priceItem": 90000,
    "productName": "Kemeja Lengan Panjang"
}
]
```

B. Class List apa yang digunakan? ArrayList atau Linked List? Jelaskan alasannya

Jawab :

Pada project ini yang digunakan adalah ArrayList, dikarenakan pada program ini hanya untuk membaca dan menampilkan data dari barang.json dan juga memungkinkan akses baca dengan cepat sehingga pada saat mengambil suatu elemen ataupun memanggil indeks yang ditunjuk untuk menampilkan pada output.

Sedangkan dalam penggunaan LinkedList tidak terlalu dibutuhkan untuk menginsert, delete, ataupun update data karena pada kasus ini hanya menampilkan data dari barang.json sehingga dalam penggunaan LinkedList tidak akan terlalu di gunakan pada program ini.

C. Bagaimana cara parsing dari json ke object di java?

```
ArrayList<Barang> data = gson.fromJson(  
    new FileReader("D:\\TUGAS PBO\\W12_PBO_PR\\src\\JavaCollectionFramework\\barang.json"),  
    new TypeToken<ArrayList<Barang>>() {}.getType()  
);
```

Cara parsing dari json ke object di java yaitu dengan cara :

```
gson.fromJson(  
    new FileReader("Letak File barang.json")  
    new TypeToken<ArrayList<class Barang.java>>() {}.getType()  
);
```

Diinputkan kedalam variabel ArrayList<Barang>data

D. Screenshoot output

```
compile-single:
run-single:
Barang ke-1 = Celana Jeans | 1 | 2 | JAKARTA | Yakin Esok Sampai(YES) | 48000 | 128000
Barang ke-2 = Kemeja Lengan Panjang | 2 | 2 | JAKARTA | Reguler (REG) | 26000 | 206000
BUILD SUCCESSFUL (total time: 1 second)
```