# LAPORAN HASIL KUIS II PRAKTIKUM ALGORITMA DAN STRUKTUR DATA



# Oleh:

DZULFIKAR MUHAMMAD AL GHIFARI NIM. 2341760071

SIB-1F / 08

D-IV SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

#### Sistem restoran

Link github: https://github.com/dzulfikar05/PraktikumAlgoritma/tree/main/kuis2

### A. Java Collection

#### **Class Restaurant08**

```
public class Restaurant08 {
    string idItem;
    string namaItem;
    double harga;

public Restaurant08() {
    }

public Restaurant08(string idItem, String namaItem, double harga) {
        this.idItem = idItem;
        this.namaItem = namaItem;
        this.harga = harga;
    }

@override
public String toString() {
        return String.format(format:"| %-75 | %-205 | Rp. %-105", idItem, namaItem, harga);
    }
}
```

deklarasi variable yang terdiri dari idltem, namaltem, dan harga

```
String idItem;
String namaItem;
double harga;
```

Konstruktor bebas dan konstruktor berparameter sesuai dengan variable yang telah di deklarasikan

```
public Restaurant08() {
}

public Restaurant08(String idItem, String namaItem, double hargo) {
    this.idItem = idItem;
    this.namaItem = namaItem;
    this.harga = harga;
}
```

Menampilkan data dengan format kelebaran masing masing

```
@override
public string toString() {
    return String.format(format:"| %-75 | %-205 | Rp. %-105", idItem, namaItem, harga);
}
```

#### Class List08

```
ort java.utii.coile
olic class List08 [
 private ArrayList<Restaurant08> menu;
    menu = new ArrayList<>();
    menu.add(item);
System.out.println();
System.out.println(x:"Berhasil menambahkan menu.");
     System.out.println();
 public Restaurant08 get(String itemId) {
     for (Restaurant08 item : menu) {
   if (item.idItem.equals(itemId)) {
             return item;
   blic void printAll() {
    for (Restaurant08 product : menu) {
        System.out.println(product);
    Collections.sort(menu, (p1, p2) -> Double.compare(p2.harga, p1.harga));
   blic void delete(string itemId) {
  menu.removeIf(item -> item.idItem.equals(itemId));
     System.out.println("Menu dengan ID " +itemId+"Sudah Dihapus.");
     System.out.println();
```

Instansiasi variable menu yang mengimplementasikan arraylist dari class restauran08

```
private ArrayList<Restaurant08> menu;
public List08() {
    menu = new ArrayList<>();
}
```

Method add berfungsi untuk menambahkan data kedalam array list

```
public void add(Restaurant08 item) {
   menu.add(item);
   System.out.println();
   System.out.println();
   System.out.println();
   System.out.println();
}
```

Method get berfungsi untuk mencari data pada array list menggunakan parameter id

```
public Restaurant08 get(String itemId) {
    for (Restaurant08 item : menu) {
        if (item.idItem.equals(itemId)) {
            return item;
        }
    }
    return null;
}
```

Method printAll menampilkan semua data pada arraylist

Method sortDesc mengurutkan harga tertinggi -> terendah, kemudian menampilkan data dengan memanggil method printAll

```
public void sortDesc() {
   Collections.sort(menu, (p1, p2) -> Double.compare(p2.harga, p1.harga));
   printAll();
}
```

Method delete menghapus data dari arraylist menggunakan parameter id

```
public void delete(String itemId) {
    menu.removeIf(item -> item.idItem.equals(itemId));
    System.out.println();
    System.out.println("Menu dengan ID " +itemId+"Sudah Dihapus.");
    System.out.println();
}
```

#### **Class Main**

Instansiasi class list08 dan menambahkan data pada arraylist

```
public class Main08 {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        List08 MenuList = new List08();

        MenuList.add(new Restaurant08(idItem:"RM001", namaItem:"Ayam Goreng Padang", harga:25000));
        MenuList.add(new Restaurant08(idItem:"RM002", namaItem:"Mie Ayam Bakso ", harga:15000));
        MenuList.add(new Restaurant08(idItem:"RM003", namaItem:"Gudeg Jogja", harga:30000));
        MenuList.add(new Restaurant08(idItem:"RM004", namaItem:"Rujak Cingur", harga:20000));
        MenuList.add(new Restaurant08(idItem:"RM005", namaItem:"Rujak Cingur", harga:50000));
```

Menampilkan tampilan menu

Case 1 untuk menambahkan data pada arraylist, dengan melakukan input data yang disimpan pada variable, kemudian memanggil method add

Case 2 untuk menghapus menu, dengan melakukan input id yang akan digunakan sebagai parameter untuk menghapus data

Case 3 untuk mencari menu, dengan melakukan input id yang akan digunakan sebagai parameter untuk search data pada array list

Case 4 akan menampilkan semua data

Case 5 akan mengurutkan data bedasarkan harga tertinggi -> terendah, kemudian di tampilkan

Case 6 akan mengentikan program

### **Hasil Compile**

# Tampilan awal menu

```
MENU [Restourant : Dzulfikar/08]

1. Tambah Menu
2. Hapus Menu
3. Cari Menu
4. Tampilkan Semua Menu
5. Tampilkan Harga dari termahal (desc)
6. Keluar

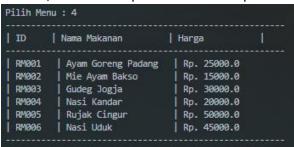
Pilih Menu :
```

Memilih menu1, yang berarti akan menambahkan menu

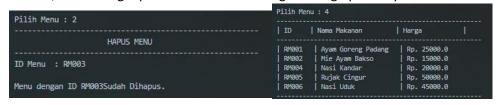
Menginput data menu, jika berhasil menambahkan menu, maka akan menampilkan message



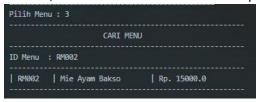
# Menu 4, akan menampilkan semua data pada arraylist



# Menu2, akan menghapus salah satu data dengan menginputkan parameter id



# Menu 3, akan mencari data berdasarkan parmeter id yang telah di inputkan



Menu 5, akan menampilkan data dengan urutan harga paling tinggi ke harga terendah

```
Pilih Menu : 5
| ID
         Nama Makanan
                                 Harga
                                  | Rp. 50000.0
| RM005
         | Rujak Cingur
                                   | Rp. 45000.0
 RM006
          Nasi Uduk
                                  | Rp. 25000.0
| Rp. 20000.0
| Rp. 15000.0
 RM001
          Ayam Goreng Padang
          | Nasi Kandar
 RM004
          | Mie Ayam Bakso
  RM002
```

# Menu 6, akan menghentikan program

```
Pilih Menu : 6
PS D:\1POLINEMA\2Genap 2023-2024\PraktikumAlgoritma\kuis2>
```

#### B. Linked List

#### Class RestaurantLL08

```
package LinkedList;

public class RestaurantLL08 {
    string idItem;
    string namaItem;
    double harga;

public RestaurantLL08() {
    }

public RestaurantLL08(string idItem, String namaItem, double harga) {
        this.idItem = idItem;
        this.namaItem = namaItem;
        this.harga = harga;
    }

@Override
public String toString() {
        return String.format(format:"| %-75 | %-205 | Rp. %-105", idItem, namaItem, harga);
    }
}
```

#### Deklarasi variable

```
String idItem;
String namaItem;
double harga;
```

Konstruktor bebas dan konstruktor berparameter sesuai dengan variable yang telah dideklarasikan

```
public RestaurantLL08() {
}

public RestaurantLL08(String idItem, String namaItem, double harga) {
    this.idItem = idItem;
    this.namaItem = namaItem;
    this.harga = harga;
}
```

Menampilkan data dengan format lebar masing masing

```
@Override
public string tostring() {
    return String.format(format:"| %-7s | %-20s | Rp. %-10s", idItem, namaItem, harga);
}
```

#### **Class Node08**

```
package LinkedList;

public class Node08 {
    RestaurantLL08 data;
    Node08 next;

public Node08(RestaurantLL08 data) {
    this.data = data;
    this.next = null;
    }
}
```

Deklarasi variable data yang berkorelasi dengan class RestauranLL08, dan variable next untuk node

```
RestaurantLL08 data;
Node08 next;
```

Konstruktor berparameter sesuai dengan variable yang telah di instansiasi

```
public Node08(RestaurantLL08 data) {
   this.data = data;
   this.next = null;
}
```

### Class LinkedList08

Instansiasi variable head

```
Mode08 head;
public LinkedList08() {
   head = null;
}
```

Method add, untuk menambahkan data

Memeriksa head, jika kosong maa akan membuat head baru, jika tidak ada maka akan menunjuk head selanjutnya

```
public void add(RestaurantLL08 data) {
    Mode08 newNode = new Node08(data);
    if (head == null) {
        head = newNode;
    } else {
        Node08 current = head;
        while (current.next != null) {
            current = current.next;
        }
        current.next = newNode;
}
```

Method get, untuk mencari data dengan parameter id Akan melakukan filter dengan perulangan dan di setiap perulangan akan dilakukan pencocokan data hingga menemukan id yang sama dengan parameter yang diberikan

```
public Restaurantil08 get(String idItem) {
    Node08 current = head;
    while (current != null) {
        if (current.data.idItem.equals(idItem)) {
            return current.data;
        }
        current = current.next;
    }
    return null;
}
```

Method printAll, akan menampikan semua data dengan fotmat lebaran yang ditentukan

Method delete, untuk menghapus data berdasarkan parameter id yang telah diberikan

```
public void delete(string idItem) {
    if (head == null) {
        return;
    }

    if (head.data.idItem.equals(idItem)) {
        head = head.next;
        return;
    }

    Mode08 current = head;
    while (current.next != null && | current.next.data.idItem.equals(idItem)) |
        current = current.next;
    }

    if (current.next != null) {
        current.next = current.next.next;
    }
}
```

Method sortDesc, untuk mengurutkan data berdasarkan harga tertinggi -> terendah. Melakukan perulangan dengan membandingkan data harga hingga data terurut dan kemudian akan ditampilkan

```
lic void sortDesc() {
  if (head == null || head.next == null) {
    return;
Node08 current = head;
while (current != null) {
     Node08 maxNode = current;
     Node08 next = current.next;
      while (next != null) {
           if (next.data.harga > maxNode.data.harga) {
               maxNode = next;
           next = next.next;
     if (maxNode != current) {
         double tempHarga = current.data.harga;
String tempIdItem = current.data.idItem;
          String tempNamaMakanan = current.data.namaItem;
         current.data.harga = maxNode.data.harga;
current.data.idItem = maxNode.data.idItem;
          current.data.namaItem = maxNode.data.namaItem;
          maxNode.data.harga = tempHarga;
maxNode.data.idItem = tempIdItem;
          maxNode.data.namaItem = tempNamaMakanan;
     current = current.next;
 printAll();
```

#### **Class Main**

Instansiasi class linkedlist08 dan menambahkan data pada linkedlist

#### Menampilkan tampilan menu

Case 1, menambahkan data dengan menginputkan data kemudian memanggil method add

Case 2, menhapus data dengan meninputkan parameter id kemudian mengirimkan pada method delete

Case 3, mencari data dengan meinputkan parameter id kemudian akan di kirimkan pada method get

Case 4 akan menampilkan semua data

Case 5 akan mengurutkan data bedasarkan harga tertinggi -> terendah, kemudian di tampilkan

Case 6 akan mengentikan program

# **Hasil Compile**

### Tampilan awal menu

```
MENU [Restourant : Dzulfikar/08]

1. Tambah Menu
2. Hapus Menu
3. Cari Menu
4. Tampilkan Semua Menu
5. Tampilkan Harga dari termahal (desc)
6. Keluar

Pilih Menu :
```

# Menu 1, menambahkan data dengan meninputkan semua data yang diperlukan

```
Pilih Menu : 1

TAMBAH MENU

ID Menu : RM908

Nama Menu : Roti bakar

Harga Menu : 10000
```

# Menu 4, menampilkan semua data

ID	Nama Makanan	Harga	
RM001	Ayam Goreng Padang	Rp. 25000.0	
RM002	Mie Ayam Bakso	Rp. 15000.0	
RM003	Gudeg Jogja	Rp. 30000.0	
RM004	Nasi Kandar	Rp. 20000.0	
RM005	Rujak Cingur	Rp. 50000.0	
RM008	Roti bakar	Rp. 10000.0	

# Menu 2, menghapus data berdasarkan parameter id yang telah di inputkan

```
Pilih Menu : 2

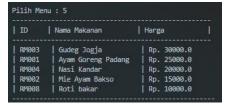
HAPUS MENU

ID Menu : RM005
```

# Menu 3, untuk mencari data berdasarkan parameter id yang telah di inputkan



# Menu 5, mengurutkan data dengan harga tertinggi -> terendan, kemudian ditampilkan



# Menu 6, menghentikan sistem

```
Pilih Menu : 6
PS D:\1POLINEMA\2Genap 2023-2024\PraktikumAlgoritma\kuis2>
```