

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:"| %-7s | %-20s | Rp. %-10s", idItem, namaItem, harga);  
    }  
}
```

deklarasi variable yang terdiri dari idItem, namaItem, 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 harga) {  
    this.idItem = idItem;  
    this.namaItem = namaItem;  
    this.harga = harga;  
}
```

Menampilkan data dengan format kelebaran masing masing

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

Class List08

```
import java.util.ArrayList;
import java.util.Collections;
public class List08 {

    private ArrayList<Restaurant08> menu;

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

    public void add(Restaurant08 item) {
        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;
            }
        }
        return null;
    }

    public void printAll() {
        System.out.println(x:"-----");
        System.out.printf(format:"| %-7s| %-20s | %-15s |", ...args:"ID", "Nama Makanan", "Harga");
        System.out.println(x:"\n-----");
        for (Restaurant08 product : menu) {
            System.out.println(product);
        }
    }

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

    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();
    }
}
```

Instansiasi variable menu yang mengimplementasikan arraylist dari class restoran08

```
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(x:"Berhasil menambahkan menu.");
    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

```
public void printAll() {
    System.out.println(x:"-----");
    System.out.printf(format:"| %-7s| %-20s | %-15s |", ...args:"ID", "Nama Makanan", "Harga");
    System.out.println(x:"\n-----");
    for (Restaurant08 product : menu) {
        System.out.println(product);
    }
}
```

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

Menampilkan tampilan menu

```
do {
    System.out.println(x:"-----");
    System.out.println(x:"          MENU    [Restourant : Dzulfikar/08]          ");
    System.out.println(x:"-----");
    System.out.println(x:"1. Tambah Menu");
    System.out.println(x:"2. Hapus Menu");
    System.out.println(x:"3. Cari Menu");
    System.out.println(x:"4. Tampilkan Semua Menu ");
    System.out.println(x:"5. Tampilkan Harga dari termahal (desc) ");
    System.out.println(x:"6. Keluar");
    System.out.println(x:"-----");

    System.out.print(s:"Pilih Menu : ");
    pil = scanner.nextInt();
    scanner.nextLine();
}
```

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

```
switch (pil) {
    case 1:
        System.out.println(x:"-----");
        System.out.println(x:"          TAMBAH MENU          ");
        System.out.println(x:"-----");
        System.out.print(s:"ID Menu  : ");
        String idMenu = scanner.nextLine();
        System.out.print(s:"Nama Menu : ");
        String namaMenu = scanner.nextLine();
        System.out.print(s:"Harga   : ");
        double harga = scanner.nextDouble();
        MenuList.add(new Restaurant08(idMenu, namaMenu, harga));
        break;
```

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

```
case 2:
    System.out.println(x:"-----");
    System.out.println(x:"          HAPUS MENU          ");
    System.out.println(x:"-----");
    System.out.print(s:"ID Menu  : ");
    idMenu = scanner.nextLine();
    MenuList.delete(idMenu);
    break;
```

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

```
case 3:
    System.out.println(x:"-----");
    System.out.println(x:"          CARI MENU          ");
    System.out.println(x:"-----");
    System.out.print(s:"ID Menu  : ");
    idMenu = scanner.nextLine();
    Restaurant08 restoran = MenuList.get(idMenu);
    if (restoran != null) {
        System.out.println(x:"-----");
        System.out.println(restoran);
        System.out.println(x:"-----");
    } else {
        System.out.println(x:"Restoran tidak ditemukan");
    }
    break;
```

Case 4 akan menampilkan semua data

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

Case 6 akan mengentikan program

```
case 4:
    MenuList.printAll();
    System.out.println(x:"-----");
    break;
case 5:
    MenuList.sortDesc();
    System.out.println(x:"-----");
    break;
case 6:
    return;
default:
```

Hasil Compile

Tampilan awal menu

```
=====
MENU [Restaurant : 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

```
Pilih Menu : 1
-----
TAMBAH MENU
-----
ID Menu : RM006
Nama Menu : Nasi Uduk
Harga Menu : 45000
Berhasil menambahkan menu.
```

Menu 4, akan menampilkan semua data pada arraylist

```
Pilih Menu : 4
-----
| 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 |
| RM006 | Nasi Uduk | Rp. 45000.0 |
-----
```

Menu2, akan menghapus salah satu data dengan menginputkan parameter id

```
Pilih Menu : 2
-----
HAPUS MENU
-----
ID Menu : RM003
Menu dengan ID RM003 Sudah Dihapus.
```

```
Pilih Menu : 4
-----
| ID | Nama Makanan | Harga |
-----
| RM001 | Ayam Goreng Padang | Rp. 25000.0 |
| RM002 | Mie Ayam Bakso | Rp. 15000.0 |
| RM004 | Nasi Kandar | Rp. 20000.0 |
| RM005 | Rujak Cingur | Rp. 50000.0 |
| RM006 | Nasi Uduk | Rp. 45000.0 |
-----
```

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

```
Pilih Menu : 3
-----
CARI MENU
-----
ID Menu : RM002
-----
| RM002 | Mie Ayam Bakso | Rp. 15000.0 |
-----
```


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

```
Pilih Menu : 5
```

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

Menu 6, akan menghentikan program

```
Pilih Menu : 6  
PS D:\POLINEMA\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:" | %-7s | %-20s | Rp. %-10s", 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 RestaurantLL08, 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

```
Node08 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) {
    Node08 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 RestaurantLL08 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 menampilkan semua data dengan fotmat lebar yang ditentukan

```
public void printAll() {
    Node08 current = head;
    System.out.println(x:"-----");
    System.out.printf(format:"| %-7s| %-20s | %-15s |", ...args:"ID", "Nama Makanan", "Harga");
    System.out.println(x:"\n-----");
    while (current != null) {
        System.out.println(current.data);
        current = current.next;
    }
}
```

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;
    }

    Node08 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

```
public 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

```
public class Main08 {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        LinkedList08 MenuList = new LinkedList08();

        MenuList.add(new RestaurantLL08(idItem:"RM001", namaItem:"Ayam Goreng Padang", harga:25000));
        MenuList.add(new RestaurantLL08(idItem:"RM002", namaItem:"Mie Ayam Bakso ", harga:15000));
        MenuList.add(new RestaurantLL08(idItem:"RM003", namaItem:"Gudeg Jogja", harga:30000));
        MenuList.add(new RestaurantLL08(idItem:"RM004", namaItem:"Nasi Kandar ", harga:20000));
        MenuList.add(new RestaurantLL08(idItem:"RM005", namaItem:"Rujak Cingur", harga:50000));
    }
}
```

Menampilkan tampilan menu

```
System.out.println(x:"=====");
System.out.println(x:"          MENU   [Restourant : Dzulfikar/08]          ");
System.out.println(x:"=====");
System.out.println(x:"1. Tambah Menu");
System.out.println(x:"2. Hapus Menu");
System.out.println(x:"3. Cari Menu");
System.out.println(x:"4. Tampilkan Semua Menu ");
System.out.println(x:"5. Tampilkan Harga dari termahal (desc) ");
System.out.println(x:"6. Keluar");
System.out.println(x:"=====");

System.out.print(s:"Pilih Menu : ");
pil = scanner.nextInt();
scanner.nextLine();
```

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

```
case 1:
    System.out.println(x:"-----");
    System.out.println(x:"                      TAMBAH MENU                      ");
    System.out.println(x:"-----");
    System.out.print(s:"ID Menu   : ");
    String idItem = scanner.nextLine();
    System.out.print(s:"Nama Menu : ");
    String namaItem = scanner.nextLine();
    System.out.print(s:"Harga Menu   : ");
    double harga = scanner.nextDouble();
    scanner.nextLine();
    MenuList.add(new RestaurantLL08(idItem, namaItem, harga));
    break;
```

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

```
case 2:
    System.out.println(x:"-----");
    System.out.println(x:"                      HAPUS MENU                      ");
    System.out.println(x:"-----");
    System.out.print(s:"ID Menu   : ");
    idItem = scanner.nextLine();
    MenuList.delete(idItem);
    break;
```

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

```
case 3:
    System.out.println(x:"-----");
    System.out.println(x:"                      CARI MENU                      ");
    System.out.println(x:"-----");
    System.out.print(s:"ID Menu   : ");
    idItem = scanner.nextLine();
    RestaurantLL08 menu = MenuList.get(idItem);
    if (menu != null) {
        System.out.println(x:"-----");
        System.out.println(menu);
        System.out.println(x:"-----");
    } else {
        System.out.println(x:"Menu tidak ditemukan");
    }
    break;
```

Case 4 akan menampilkan semua data

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

Case 6 akan mengentikan program

```
case 4:
    MenuList.printAll();
    System.out.println(x:"-----");
    break;
case 5:
    MenuList.sortDesc();
    System.out.println(x:"-----");
    break;
case 6:
    return;
```

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 : RM008
Nama Menu : Roti bakar
Harga Menu : 10000
```

Menu 4, menampilkan semua data

```
Pilih Menu : 4
=====
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

```
=====
Pilih Menu : 3
=====
CARI MENU
=====
ID Menu : RM008
=====
RM008 | Roti bakar | Rp. 10000.0
=====
```

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

```
Pilih Menu : 5
=====
ID | Nama Makanan | Harga |
=====
RM003 | Gudeg Jogja | Rp. 30000.0
RM001 | Ayam Goreng Padang | Rp. 25000.0
RM004 | Nasi Kandar | Rp. 20000.0
RM002 | Mie Ayam Bakso | Rp. 15000.0
RM008 | Roti bakar | Rp. 10000.0
=====
```

Menu 6, menghentikan sistem

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