

KUIS PRAKTIKUM ALGORITMA DAN STRUKTUR DATA



2022



Daftar Isi

KUIS.....	3
SOURCE CODE.....	3
OUTPUT NOMOR 1 TAMBAH ANTRIAN.....	8
OUTPUT NOMOR 2 CETAK ANTRIAN.....	9
OUTPUT NOMOR 3 HAPUS ANTRIAN.....	9
OUTPUT NOMOR 4 LAPORAN PESANAN (URUT BY NAMA).....	10
OUTPUT NOMOR 5 HITUNG TOTAL PENDAPATAN.....	10

Kuis

Source code

DoubleLinkedList_26

```
public class DoubleLinkedList_26 {
    Node_26 head;
    NodePesanan_26 headPesanan;
    int size;

    DoubleLinkedList_26() {
        head = null;
        size = 0;
    }

    public boolean isEmpty() {
        return head == null;
    }

    public void addFirst(Pembeli_26 item) {
        if (isEmpty()) {
            head = new Node_26(null, item, null);
        } else {
            Node_26 newNode = new Node_26(null, item, head);
            head.prev = newNode;
            head = newNode;
        }
        size++;
    }

    public void addLast(Pembeli_26 item) {
        if (isEmpty()) {
            addFirst(item);
        } else {
            Node_26 current = head;
            while (current.next != null) {
                current = current.next;
            }
            Node_26 newNode = new Node_26(current, item, null);
            current.next = newNode;
            size++;
        }
    }

    public void print() {
        if (!isEmpty()) {
            Node_26 tmp = head;
            System.out.println("No.\t\t>Nama Customer\t\t>No HP");
            while (tmp != null) {
                System.out.println("|" + tmp.data.antrian + "\t\t|" +
tmp.data.namaPembeli + "\t\t|" + tmp.data.NoHP);
                tmp = tmp.next;
            }
        } else {
            System.out.println("Linked Lists Kosong");
        }
    }

    public void removeFirst() throws Exception {
        if (isEmpty()) {
            throw new Exception("Antrian masih kosong, tidak dapat
dihapus");
        }
    }
}
```

```

    } else if (size == 1) {
        removeLast();
    } else {
        System.out.println(head.data.namaPembeli + " telah memesan
Menu");
        head = head.next;
        head.prev = null;
        size--;
    }
}

public void removeLast() throws Exception {
    if (isEmpty()) {
        throw new Exception("Antrian masih kosong, tidak dapat
dihapus");
    } else if (head.next == null) {
        head = null;
        size--;
        return;
    }
    Node_26 current = head;
    while (current.next.next != null) {
        current = current.next;
    }
    current.next = null;
    size--;
}
}

```

DoubleLinkedListPesanan_26

```

public class DoubleLinkedListPesanan_26 {
    NodePesanan_26 head;
    int size;

    DoubleLinkedListPesanan_26() {
        head = null;
        size = 0;
    }

    public boolean isEmpty() {
        return head == null;
    }

    public void addFirst(Pesanan_26 item) {
        if (isEmpty()) {
            head = new NodePesanan_26(null, item, null);
        } else {
            NodePesanan_26 newNode = new NodePesanan_26(null, item, head);
            head.prev = newNode;
            head = newNode;
        }
        size++;
    }

    public void addLast(Pesanan_26 item) {
        if (isEmpty()) {
            addFirst(item);
        } else {
            NodePesanan_26 current = head;
            while (current.next != null) {
                current = current.next;
            }
            NodePesanan_26 newNode = new NodePesanan_26(current, item,

```



```

null);
        current.next = newNode;
        size++;
    }
}

public void print() {
    if (!isEmpty()) {
        NodePesanan_26 tmp = head;
        System.out.println("No.\t\t>Nama Pesanan\t\t|Harga");
        while (tmp != null) {
            System.out.println("|" + tmp.data.kodePesanan + "\t\t|" +
tmp.data.namaPesanan + "\t\t|" + tmp.data.harga);
            tmp = tmp.next;
        }
        System.out.println("\nberhasil diisi");
    } else {
        System.out.println("Linked Lists Kosong");
    }
}

public void sortByNama() {
    NodePesanan_26 current, index;
    Pesanan_26 tmp;
    if (isEmpty()) {
        System.out.println("Data Linked list kosong");
    } else {
        for (current = head; current.next != null; current =
current.next) {
            for (index = current.next; index != null; index =
index.next) {
                if
(current.data.namaPesanan.compareTo(index.data.namaPesanan) > 0) {
                    tmp = current.data;
                    current.data = index.data;
                    index.data = tmp;
                }
            }
        }
    }
    print();
}

public int totalPendapatan() {
    int total = 0;
    if (!isEmpty()) {
        NodePesanan_26 tmp = head;
        while (tmp != null) {
            total += tmp.data.harga;
            tmp = tmp.next;
        }
    } else {
        System.out.println("Linked Lists Kosong");
    }
    return total;
}
}

```

Main_26

```
import java.util.Scanner;
```



```

public class Main_26 {
    public static Scanner input = new Scanner(System.in);
    public static void main(String[] args) throws Exception {
        System.out.println("===== QUIZ 2 PRAKTIKUM ASD TI - 1C DAN TI - 1F =====");
        System.out.println("dibuat oleh : Ridwan Caesar Rizqi Karisma Biwarni");
        System.out.println("NIM : 2141720183");
        System.out.println("Absen : 26");

        System.out.println("=====");
        System.out.println("Sistem Antrian Resto Royal Delish");
        DoubleLinkedList_26 dll = new DoubleLinkedList_26();
        DoubleLinkedListPesanan_26 dllPesanan = new DoubleLinkedListPesanan_26();
        int menu;
        // penambahan data tanpa input data melalui keyboard
        dll.addLast(new Pembeli_26("Ridwan", "085131232", 1));
        dll.addLast(new Pembeli_26("Caesar", "082142412", 2));
        dll.addLast(new Pembeli_26("Rizqi", "0842142142", 3));
        int noAntrian = 1;
        do {
            System.out.println("Menu");
            System.out.println("1. Tambah Antrian");
            System.out.println("2. Cetak Antrian");
            System.out.println("3. Hapus Antrian");
            System.out.println("4. Laporan Pengurutan pesanan by nama");
            System.out.println("5. Hitung total Pendapatan");
            System.out.println("6. Keluar");
            System.out.print("Pilih (1-6):");
            menu = input.nextInt();
            switch (menu) {
                case 1 -> {
                    System.out.println("-----");
                    System.out.println("Masukkan Data Pembeli");
                    System.out.println("-----");
                    System.out.println("Nomor Antrian : " + (dll.size + 1));

                    System.out.print("Nama Customer : ");
                    String namaCustomer = input.next();
                    input.nextLine();
                    System.out.print("Nomor Hp : ");
                    String noHP = input.nextLine();
                    Pembeli_26 newPembeli = new Pembeli_26(namaCustomer, noHP, (dll.size + 1));
                    dll.addLast(newPembeli);
                }
                case 2 -> {
                    System.out.println("+++++++");
                    System.out.println("Daftar Antrian Resto Royal Delish");

                    System.out.println("+++++++");
                    dll.print();
                }
                case 3 -> {
                    dll.removeFirst();
                    System.out.println("-----");
                    System.out.println("Transaksi input pesanan");
                    System.out.println("-----");
                    System.out.println("Nomor Pesanan : " + (dllPesanan.size + 1));

```



```

        System.out.print("Pesanan : ");
        String pesanan = input.next();
        input.nextLine();
        System.out.print("Harga : ");
        int harga = input.nextInt();
        Pesanan_26 newPesanan = new Pesanan_26((dllPesanan.size
+ 1),pesanan, harga);
        dllPesanan.addLast(newPesanan);
        dllPesanan.print();
    }
    case 4 -> {
        System.out.println("+++++++");
        System.out.println("Daftar Pesanan Masuk Resto Royal
Delish");

        System.out.println("+++++++");
        dllPesanan.sortByNama();
    }
    case 5 -> {
        System.out.println("+++++++");
        System.out.println("Total Pendapatan");
        System.out.println("+++++++");
        System.out.println("Pendapatan hari ini " +
dllPesanan.totalPendapatan());
    }
    case 6 -> {
        System.exit(0);
    }
    }
} while (menu >= 1 && menu <= 6);
}
}

```

Node_26

```

public class Node_26 {
    Pembeli_26 data;
    Node_26 prev, next;

    Node_26(Node_26 prev, Pembeli_26 data, Node_26 next) {
        this.prev = prev;
        this.data = data;
        this.next = next;
    }
}

```

NodePesanan_26

```

public class NodePesanan_26 {
    Pesanan_26 data;
    NodePesanan_26 prev, next;

    NodePesanan_26(NodePesanan_26 prev, Pesanan_26 data, NodePesanan_26
next) {
        this.prev = prev;
        this.data = data;
        this.next = next;
    }
}

```

Pembeli_26

```
public class Pembeli_26 {
    String namaPembeli;
    String NoHP;
    int antrian;

    Pembeli_26(String b, String c, int d) {
        namaPembeli = b;
        NoHP = c;
        antrian = d;
    }
}
```

Pesanan_26

```
public class Pesanan_26 {
    int kodePesanan;
    String namaPesanan;
    int harga;

    Pesanan_26(int a, String b, int c) {
        kodePesanan = a;
        namaPesanan = b;
        harga = c;
    }
}
```

Output Nomor 1 Tambah Antrian

```
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe"
===== QUIZ 2 PRAKTIKUM ASD TI - 1C DAN TI - 1F =====
dibuat oleh : Ridwan Caesar Rizqi Karisma Biwarni
NIM : 2141720183
Absen : 26
=====
Sistem Antrian Resto Royal Delish
Menu
1. Tambah Antrian
2. Cetak Antrian
3. Hapus Antrian
4. Laporan Pengurutan pesanan by nama
5. Hitung total Pendapatan
6. Keluar
Pilih (1-6):1
-----
Masukkan Data Pembeli
-----
Nomor Antrian : 4
Nama Customer : anto
Nomor Hp : 08121313
```


Output Nomor 2 Cetak Antrian

```

Menu
1. Tambah Antrian
2. Cetak Antrian
3. Hapus Antrian
4. Laporan Pengurutan pesanan by nama
5. Hitung total Pendapatan
6. Keluar
Pilih (1-6):2
+++++
Daftar Antrian Resto Royal Delish
+++++
No.      |Nama Customer      |No HP
|1       |Ridwan             |085131232
|2       |Caesar             |082142412
|3       |Rizqi              |0842142142
|4       |Ianto              |08121313

```

Output Nomor 3 Hapus Antrian

```

Menu
1. Tambah Antrian
2. Cetak Antrian
3. Hapus Antrian
4. Laporan Pengurutan pesanan by nama
5. Hitung total Pendapatan
6. Keluar
Pilih (1-6):3
Ridwan telah memesan Menu
-----
Transaksi input pesanan
-----
Nomor Pesanan : 1
Pesanan : Rendang
Harga : 10000
No.      |Nama Pesanan      |Harga
|1       |Rendang           |10000

```

Dan saat sukses mengembalikan maka antrian berkurang.

```
Menu
1. Tambah Antrian
2. Cetak Antrian
3. Hapus Antrian
4. Laporan Pengurutan pesanan by nama
5. Hitung total Pendapatan
6. Keluar
Pilih (1-6):2
+++++
Daftar Antrian Resto Royal Delish
+++++
No.      |Nama Customer      |No HP
|2       |Caesar             |082142412
|3       |Rizqi              |0842142142
|4       |Ianto              |08121313
```

Output Nomor 4 Laporan pesanan (urut by Nama)

```
Menu
1. Tambah Antrian
2. Cetak Antrian
3. Hapus Antrian
4. Laporan Pengurutan pesanan by nama
5. Hitung total Pendapatan
6. Keluar
Pilih (1-6):4
+++++
Daftar Pesanan Masuk Resto Royal Delish
+++++
No.      |Nama Pesanan      |Harga
|2       |Bakso             |10000
|1       |Rendang           |10000
```

Output Nomor 5 Hitung total pendapatan

Total pendapatan 20000 karena ada 2 pesanan bakso dan rendang yang masing-masing berharga 10000

```
Menu
1. Tambah Antrian
2. Cetak Antrian
3. Hapus Antrian
4. Laporan Pengurutan pesanan by nama
5. Hitung total Pendapatan
6. Keluar
Pilih (1-6):5
+++++
Total Pendapatan
+++++
Pendapatan hari ini 20000
```