

Kuis 2 Algoritma dan Strukur Data

Nama :Alex

Kelas : 1F

Absen : 03

```
1 package Kuis2_02_Alexander_1F;
2
3 public class pembeli03 {
4     public String kodepesanan;
5     public String namaPembeli;
6     public String phoneNumber;
7
8     public pembeli03(String kodepesanan, String namaPembeli, String phoneNumber) {
9         this.kodepesanan = kodepesanan;
10        this.namaPembeli = namaPembeli;
11        this.phoneNumber = phoneNumber;
12    }
13 }
14
```

```
1 package Kuis2_02_Alexander_1F;
2
3 public class Node03 {
4     Node03 prev;
5     pembeli03 data;
6     Node03 next;
7     pesanan03 next2;
8
9     public Node03(Node03 prev, pembeli03 data, Node03 next, pesanan03 next2) {
10        this.prev = prev;
11        this.data = data;
12        this.next = next;
13        this.next2 = next2;
14    }
15 }
16
```

```
1 package Kuis2_02_Alexander_1F;
2
3 public class pesanan03 {
4     public String kodePesanan;
5     public String namaPesanan;
6     public int harga;
7
8     public pesanan03(String kodePesanan, String namaPesanan, int harga) {
9         this.kodePesanan = kodePesanan;
10        this.namaPesanan = namaPesanan;
11        this.harga = harga;
12    }
13 }
14
```

```

1 package Kuis2_K2_Alexander_13;
2
3 public class DoublyLinkedList03 {
4     Node03 head;
5     Node03 tail;
6     int size;
7
8     public DoublyLinkedList03() {
9         head = null;
10        tail = null;
11        size = 0;
12    }
13
14    public boolean isEmpty() {
15        return head == null;
16    }
17
18    public void enqueue(int item1, item2 item3) {
19        if (isEmpty()) {
20            head = new Node03(null, item1, null, item2);
21            tail = head;
22        } else {
23            Node03 newNode = new Node03(tail, item1, null, item2);
24            tail.next = newNode;
25            tail = newNode;
26        }
27        size++;
28    }
29
30    public void dequeue() throws Exception {
31        if (isEmpty()) {
32            throw new Exception("Antrian masih kosong");
33        } else {
34            head = head.next;
35            if (head != null) {
36                head.prev = null;
37            } else {
38                tail = null;
39            }
40            size--;
41        }
42    }
43
44    public void peek() throws Exception {
45        if (isEmpty()) {
46            throw new Exception("Antrian masih kosong");
47        } else {
48            System.out.println("Antrian Terdepan: " + head.data.kodePesanan + ", " + head.data.namaPembeli + ", " + head.data.phoneNumber + ", " + head.next2.namaPesanan + ", " + head.next2.harga);
49        }
50    }
51
52    public void peekRear() throws Exception {
53        if (isEmpty()) {
54            throw new Exception("Antrian masih kosong");
55        } else {
56            System.out.println("Antrian Terakhir: " + tail.data.kodePesanan + ", " + tail.data.namaPembeli + ", " + tail.data.phoneNumber + ", " + tail.next2.namaPesanan + ", " + tail.next2.harga);
57        }
58    }
59
60    public int size() {
61        return size;
62    }
63
64    public void print() {
65        if (!isEmpty()) {
66            Node03 tmp = head;
67            while (tmp != null) {
68                System.out.println("Kode Pesanan: " + tmp.data.kodePesanan + ", Nama Pembeli: " + tmp.data.namaPembeli + ", No Telp: " + tmp.data.phoneNumber + ", Nama Pesanan: " + tmp.next2.namaPesanan + ", Harga: " + tmp.next2.harga);
69                tmp = tmp.next;
70            }
71        } else {
72            System.out.println("Antrian masih kosong");
73        }
74    }
75
76    public void sortByName() {
77        if (isEmpty()) {
78            System.out.println("Antrian masih kosong");
79            return;
80        }
81        Node03 current, index;
82        Node03 tempData;
83        int tempOrder;
84
85        for (current = head; current.next != null; current = current.next) {
86            for (index = current.next; index != null; index = index.next) {
87                if (current.data.namaPembeli.compareToIgnoreCase(index.data.namaPembeli) > 0) {
88                    tempData = current.data;
89                    tempOrder = current.next2;
90                    current.data = index.data;
91                    current.next2 = index.next2;
92                    index.data = tempData;
93                    index.next2 = tempOrder;
94                }
95            }
96        }
97    }
98
99    public void printSortByName() {
100        sortByName();
101        print();
102    }
103
104    public int calculateTotalRevenue() {
105        if (isEmpty()) {
106            return 0;
107        }
108
109        Node03 tmp = head;
110        int totalRevenue = 0;
111        while (tmp != null) {
112            totalRevenue += tmp.next2.harga;
113            tmp = tmp.next;
114        }
115        return totalRevenue;
116    }
117 }
118
119

```

```

1 package Kuis2_02_Alexander_1F;
2
3 import java.util.Scanner;
4
5 public class AntrianMain03 {
6
7     public static void main(String[] args) throws Exception {
8         Scanner sc = new Scanner(System.in);
9
10        // penyimpanan data antrian dan pesanan
11        DoubleLinkedList03 antrian = new DoubleLinkedList03();
12        antrian.Enqueue(new pembeli03("001", "Alex", "0894123456", new pesanan03("001", "Mie Ayam", 2000));
13        antrian.Enqueue(new pembeli03("002", "Dedy", "0895777777", new pesanan03("002", "Bakso", 1000));
14        antrian.Enqueue(new pembeli03("003", "Yefta", "0894111111", new pesanan03("003", "Rujak Soto", 3000));
15        antrian.Enqueue(new pembeli03("004", "Roky", "0894222222", new pesanan03("004", "Soto Ayam", 4000));
16
17        try {
18            while (true) {
19                System.out.println("=====");
20                System.out.println("PENGANTRI pesanan makanan");
21                System.out.println("=====");
22                System.out.println("1. Tambah Data Antrian");
23                System.out.println("2. Daftar Antrian");
24                System.out.println("3. Hapus Data Antrian");
25                System.out.println("4. Laporan urutan pesanan by nama ");
26                System.out.println("5. Hitung total pendapatan");
27                System.out.println("6. Keluar");
28                System.out.println("=====");
29                System.out.print("Pilih Menu (1-6): ");
30                int pilih = sc.nextInt();
31                sc.nextLine();
32                switch (pilih) {
33                    case 1:
34                        System.out.print("Memor Antrian: ");
35                        String nomerAntrian = sc.nextLine();
36                        System.out.print("Nama Antrian: ");
37                        String nama = sc.nextLine();
38                        System.out.print("Memor Telepon: ");
39                        String noTelp = sc.nextLine();
40                        System.out.print("Nama Pesanan: ");
41                        String namaPesanan = sc.nextLine();
42                        System.out.print("Harga: ");
43                        int harga = sc.nextInt();
44                        sc.nextLine();
45                        antrian.Enqueue(new pembeli03(nomerAntrian, nama, noTelp, new pesanan03(nomerAntrian, namaPesanan, harga));
46                        break;
47                    case 2:
48                        antrian.print();
49                        System.out.println("Sisa Antrian: " + antrian.size());
50                        break;
51                    case 3:
52                        antrian.Dequeue();
53                        break;
54                    case 4:
55                        antrian.printSortedByName();
56                        break;
57                    case 5:
58                        int totalRevenue = antrian.calculateTotalRevenue();
59                        System.out.println("Total Pendapatan: " + totalRevenue);
60                        break;
61                    case 6:
62                        System.out.println("Terima Kasih");
63                        System.exit(0);
64                        break;
65                    default:
66                        System.out.println("Pilihan Tidak Tersedia");
67                }
68            }
69        } catch (Exception e) {
70            System.out.println("Terjadi Kesalahan: " + e.getMessage());
71        } finally {
72            sc.close();
73        }
74    }
75 }
76

```

Hasil;

```

=====
PENGANTRI pesanan makanan
=====

1. Tambah Data Antrian
2. Daftar Antrian
3. Hapus Data Antrian
4. Laporan urutan pesanan by nama
5. Hitung total pendapatan
6. Keluar
=====

Pilih Menu (1-6): 1
Nomor Antrian: 5
Nama Antrian: susi susisanto
Nomor Telepon: 99999
Nama Pesanan: Nasi goreng pedas
Harga: 90000
=====

PENGANTRI pesanan makanan
=====

1. Tambah Data Antrian
2. Daftar Antrian
3. Hapus Data Antrian
4. Laporan urutan pesanan by nama
5. Hitung total pendapatan
6. Keluar
=====

Pilih Menu (1-6): 2
Kode Pesanan: 001, Nama Pembeli: Alex, No Telp: 0894123456, Nama Pesanan: Mie Ayam, Harga: 2000
Kode Pesanan: 002, Nama Pembeli: Dedy, No Telp: 0895777777, Nama Pesanan: Bakso, Harga: 1000
Kode Pesanan: 003, Nama Pembeli: Yefta, No Telp: 0894111111, Nama Pesanan: Rujak Soto, Harga: 3000
Kode Pesanan: 004, Nama Pembeli: Roky, No Telp: 0894222222, Nama Pesanan: Soto Ayam, Harga: 4000
Kode Pesanan: 5, Nama Pembeli: susi susisanto, No Telp: 99999, Nama Pesanan: Nasi goreng pedas, Harga: 90000
Sisa Antrian: 5
=====

PENGANTRI pesanan makanan

```

```
=====
PENGANTRI pesanan makanan
=====
```

1. Tambah Data Antrian
2. Daftar Antrian
3. Hapus Data Antrian
4. Laporan urutan pesanan by nama
5. Hitung total pendapatan
6. Keluar

```
=====
Pilih Menu (1-6): 3
=====
```

```
PENGANTRI pesanan makanan
=====
```

1. Tambah Data Antrian
2. Daftar Antrian
3. Hapus Data Antrian
4. Laporan urutan pesanan by nama
5. Hitung total pendapatan
6. Keluar

```
=====
Pilih Menu (1-6): 2
```

```
Kode Pesanan: 002, Nama Pembeli: Dedy, No Telp: 0895777777, Nama Pesanan: Bakso, Harga: 1000
Kode Pesanan: 003, Nama Pembeli: Yefta, No Telp: 0894111111, Nama Pesanan: Rujak Soto, Harga: 3000
Kode Pesanan: 004, Nama Pembeli: Roky, No Telp: 0894222222, Nama Pesanan: Soto Ayam, Harga: 4000
Kode Pesanan: 5, Nama Pembeli: susi susisanto, No Telp: 99999, Nama Pesanan: Nasi goreng pedas, Harga: 90000
Sisa Antrian: 4
=====
```

```
PENGANTRI pesanan makanan
=====
```

1. Tambah Data Antrian
2. Daftar Antrian
3. Hapus Data Antrian
4. Laporan urutan pesanan by nama
5. Hitung total pendapatan

2. Daftar Antrian
3. Hapus Data Antrian
4. Laporan urutan pesanan by nama
5. Hitung total pendapatan
6. Keluar

```
=====
Pilih Menu (1-6): 4
```

```
Kode Pesanan: 002, Nama Pembeli: Dedy, No Telp: 0895777777, Nama Pesanan: Bakso, Harga: 1000
Kode Pesanan: 004, Nama Pembeli: Roky, No Telp: 0894222222, Nama Pesanan: Soto Ayam, Harga: 4000
Kode Pesanan: 5, Nama Pembeli: susi susisanto, No Telp: 99999, Nama Pesanan: Nasi goreng pedas, Harga: 90000
Kode Pesanan: 003, Nama Pembeli: Yefta, No Telp: 0894111111, Nama Pesanan: Rujak Soto, Harga: 3000
=====
```

```
PENGANTRI pesanan makanan
=====
```

1. Tambah Data Antrian
2. Daftar Antrian
3. Hapus Data Antrian
4. Laporan urutan pesanan by nama
5. Hitung total pendapatan
6. Keluar

```
=====
Pilih Menu (1-6): 5
```

```
Total Pendapatan: 98000
=====
```

```
PENGANTRI pesanan makanan
=====
```

1. Tambah Data Antrian
2. Daftar Antrian
3. Hapus Data Antrian
4. Laporan urutan pesanan by nama
5. Hitung total pendapatan
6. Keluar

```
=====
Pilih Menu (1-6): 6
```

```
Terima Kasih
```