

LAPORAN PRAKTIKUM

MATA KULIAH ALGORITMA DAN STRUKTUR DATA

Dosen Pengampu : Triana Fatmawati, S.T, M.T

KUIS 2



Nama : M. Zidna Billah Faza
NIM : 2341760030
Prodi : D-IV Sistem Informasi Bisnis

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2024

PROGRAM COLLECTION

Deskripsi Program

Program dalam bahasa java ini mengimplementasikan java collection untuk membuat sebuah sistem manajemen pada perusahaan laundry menjadi lebih mudah dalam pencatatannya. Program ini memiliki fitur menambahkan pelanggan, menghapus pelanggan, melihat pelanggan, menambahkan transaksi, menghapus transaksi, melihat transaksi, mengurutkan pelanggan berdasarkan nama, serta mencari transaksi dengan nama pelanggan. Java collection yang dipakai adalah LinkedList dan untuk sorting menggunakan Comparator.

Screenshot Program dan Output

● Transaksi_18.java

```
package ProgramCollection;

public class Transaksi_18 {

    public static int idTransaksiCounter = 1;
    int idTransaksi;
    Pelanggan_18 Pelanggan;
    String jenisLayanan;
    double berat;
    double harga;

    public Transaksi_18(Pelanggan_18 Pelanggan, String jenisLayanan, double berat, double harga) {
        this.idTransaksi = idTransaksiCounter++;
        this.Pelanggan = Pelanggan;
        this.jenisLayanan = jenisLayanan;
        this.berat = berat;
        this.harga = harga;
    }

    @Override
    public String toString() {
        return "|" + idTransaksi + "|" + Pelanggan.namaPelanggan + "|" + jenisLayanan + "|" + berat + "|" + harga + "|";
    }
}
```

- Pelanggan_18.java

```
package ProgramCollection;

public class Pelanggan_18 {

    int idPelanggan;
    String namaPelanggan;
    String noTelp;

    public Pelanggan_18(int idPelanggan, String namaPelanggan, String noTelp) {
        this.idPelanggan = idPelanggan;
        this.namaPelanggan = namaPelanggan;
        this.noTelp = noTelp;
    }

    @Override

    public String toString() {
        return "|" + idPelanggan + "\t|" + namaPelanggan + "\t\t|" + noTelp + "\t\t|";
    }
}
```

- Import java collection

```
import java.util.Scanner;
import java.util.LinkedList;
import java.util.Collections;
import java.util.Comparator;
```

- Inisialisasi LinkedList pada pelanggan dan Transaksi

```
public static LinkedList<Pelanggan_18> listPelanggan = new LinkedList<>();
public static LinkedList<Transaksi_18> listTransaksi = new LinkedList<>();
public static Scanner input18 = new Scanner(System.in);
```

- Isi data pada listpelanggan

```
listPelanggan.add(new Pelanggan_18(idPelanggan:1, namaPelanggan:"Ahmad", noTelp:"081234567890"));
listPelanggan.add(new Pelanggan_18(idPelanggan:2, namaPelanggan:"Budi", noTelp:"081234567891"));
listPelanggan.add(new Pelanggan_18(idPelanggan:3, namaPelanggan:"Sandi", noTelp:"081234567892"));
listPelanggan.add(new Pelanggan_18(idPelanggan:4, namaPelanggan:"Dewi", noTelp:"081234567893"));
listPelanggan.add(new Pelanggan_18(idPelanggan:5, namaPelanggan:"Eka", noTelp:"081234567894"));
```

- Pilihan menu

```
header();
System.out.println(x:"          LAUNDRY JAYA MANDIRI");
header();
System.out.println(x:"1. Tambah Pelanggan");
System.out.println(x:"2. Hapus Pelanggan");
System.out.println(x:"3. Lihat Pelanggan");
System.out.println(x:"4. Tambah Transaksi");
System.out.println(x:"5. Hapus Transaksi");
System.out.println(x:"6. Lihat Transaksi");
System.out.println(x:"7. Urutkan Pelanggan Berdasarkan Nama (A-Z)");
System.out.println(x:"8. Cari Transaksi Berdasarkan Nama Pelanggan");
System.out.println(x:"9. Keluar");
header();
System.out.print(s:"Masukkan pilihan      : ");
pilihan = input18.nextInt();
```

```
=====
                        LAUNDRY JAYA MANDIRI
=====
1. Tambah Pelanggan
2. Hapus Pelanggan
3. Lihat Pelanggan
4. Tambah Transaksi
5. Hapus Transaksi
6. Lihat Transaksi
7. Urutkan Pelanggan Berdasarkan Nama (A-Z)
8. Cari Transaksi Berdasarkan Nama Pelanggan
9. Keluar
=====
```

- Tambah pelanggan

```
public static void tambahPelanggan() {
    header();
    System.out.println(x:"          Tambah Pelanggan");
    header();
    System.out.print(s:"Masukkan ID Pelanggan      : ");
    int id = input18.nextInt();
    input18.nextLine();
    System.out.print(s:"Masukkan Nama Pelanggan      : ");
    String nama = input18.nextLine();
    System.out.print(s:"Masukkan No Telp Pelanggan    : ");
    String noTelp = input18.nextLine();
    listPelanggan.add(new Pelanggan_18(id, nama, noTelp));
    System.out.println(x:"Pelanggan berhasil ditambahkan!");
}
```

```
=====
Masukkan pilihan      : 1
=====
          Tambah Pelanggan
=====
Masukkan ID Pelanggan      : 6
Masukkan Nama Pelanggan    : Sarah
Masukkan No Telp Pelanggan : 081234567895
Pelanggan berhasil ditambahkan!
```

- Hapus pelanggan

```
public static void hapusPelanggan() {
    header();
    System.out.println(x:"          Hapus Pelanggan");
    header();
    lihatPelanggan();
    System.out.print(s:"Masukkan ID Pelanggan yang akan dihapus : ");
    int id = input18.nextInt();
    listPelanggan.removeIf(p -> p.idPelanggan == id);
    System.out.println(x:"Pelanggan berhasil dihapus!");
}
```

```
=====
Masukkan pilihan      : 2
=====
          Hapus Pelanggan
=====
| ID | Nama      | No Telp      |
|----|-----|-----|
| 1  | Ahmad    | 081234567890 |
| 2  | Budi     | 081234567891 |
| 3  | Sandi    | 081234567892 |
| 4  | Dewi     | 081234567893 |
| 5  | Eka      | 081234567894 |
| 6  | Sarah    | 081234567895 |
|----|-----|-----|
Masukkan ID Pelanggan yang akan dihapus : 4
Pelanggan berhasil dihapus!
```

- Lihat data pelanggan

```
public static void lihatPelanggan() {
    System.out.println(x:"-----");
    System.out.println(x:"| ID\t| Nama\t\t| No Telp\t\t|");
    System.out.println(x:"-----");
    for (Pelanggan_18 pelanggan : listPelanggan) {
        System.out.println(pelanggan);
    }
    System.out.println(x:"-----");
}
```

| | | | |
|----------------------|-------|--------------|--|
| ===== | | | |
| Masukkan pilihan : 3 | | | |
| ID | Nama | No Telp | |
| ----- | | | |
| 1 | Ahmad | 081234567890 | |
| 2 | Budi | 081234567891 | |
| 3 | Sandi | 081234567892 | |
| 5 | Eka | 081234567894 | |
| 6 | Sarah | 081234567895 | |
| ----- | | | |

- Tambah transaksi

```
public static void tambahTransaksi() {
    header();
    System.out.println(x:"          Tambah Transaksi");
    header();
    lihatPelanggan();
    System.out.print(s:"Masukkan ID Pelanggan          : ");
    int idPelanggan = input18.nextInt();
    Pelanggan_18 pelanggan = findPelangganById(idPelanggan);
    if (pelanggan == null) {
        System.out.println(x:"Pelanggan tidak ditemukan!");
        return;
    }

    String jenisLayanan = pilihLayanan();
    System.out.print(s:"Masukkan berat atau jumlah pcs      : ");
    double jumlah = input18.nextDouble();
    double harga = hitungHarga(jenisLayanan, jumlah);

    listTransaksi.add(new Transaksi_18(pelanggan, jenisLayanan, jumlah, harga));
    System.out.println(x:"Transaksi berhasil ditambahkan!");
}
```

```
=====
Masukkan pilihan          : 4
=====
          Tambah Transaksi
=====
| ID | Nama          | No Telp          |
|----|-----|-----|
| 1  | Ahmad         | 081234567890    |
| 2  | Budi          | 081234567891    |
| 3  | Sandi         | 081234567892    |
| 5  | Eka           | 081234567894    |
| 6  | Sarah         | 081234567895    |
|----|-----|-----|
Masukkan ID Pelanggan          : 1
=====
          Jenis Layanan
=====
1. Cuci Kering Regular (6000/kg)
2. Cuci Kering Ekspres (9000/kg)
3. Cuci Setrika Regular (8000/kg)
4. Cuci Setrika Ekspres (12000/kg)
5. Cuci Sepatu (35000/pcs)
6. Cuci Boneka (10000/pcs)
=====
Pilih jenis layanan          : 3
Masukkan berat atau jumlah pcs : 3.4
Transaksi berhasil ditambahkan!
```

- Hapus transaksi

```
public static void hapusTransaksi() {
    header();
    System.out.println(x:"          Hapus Transaksi");
    header();
    lihatTransaksi();
    System.out.print(s:"Masukkan ID Transaksi yang akan dihapus   : ");
    int id = input18.nextInt();
    listTransaksi.removeIf(t -> t.idTransaksi == id);
    System.out.println(x:"Transaksi berhasil dihapus!");
}
```

```
=====
Masukkan pilihan      : 5
=====
                        Hapus Transaksi
=====
```

| ID | Nama Pelanggan | Jenis Layanan | Berat/pcs | Harga |
|----|----------------|----------------------|-----------|---------|
| 1 | Ahmad | Cuci Setrika Regular | 3.4 | 27200.0 |
| 2 | Sandi | Cuci Kering Regular | 2.8 | 16800.0 |

```
=====
Masukkan ID Transaksi yang akan dihapus   : 1
Transaksi berhasil dihapus!
```

- Lihat data transaksi

```
public static void lihatTransaksi() {
    System.out.println(x:"-----");
    System.out.println(x:"| ID\t| Nama Pelanggan\t| Jenis Layanan\t| Berat/pcs\t| Harga\t|");
    System.out.println(x:"-----");
    for (Transaksi_18 transaksi : listTransaksi) {
        System.out.println(transaksi);
    }
    System.out.println(x:"-----");
}
```

```
=====
Masukkan pilihan      : 6
=====
```

| ID | Nama Pelanggan | Jenis Layanan | Berat/pcs | Harga |
|----|----------------|---------------------|-----------|---------|
| 2 | Sandi | Cuci Kering Regular | 2.8 | 16800.0 |

- Mengurutkan data pelanggan berdasarkan nama dengan urutan A - Z

```
public static void urutkanPelanggan() {
    header();
    System.out.println(x:"      Urutkan Pelanggan Berdasarkan Nama (A-Z)");
    header();
    Collections.sort(listPelanggan, Comparator.comparing(p -> p.namaPelanggan));
    lihatPelanggan();
}
```

```
=====
Masukkan pilihan      : 7
=====
      Urutkan Pelanggan Berdasarkan Nama (A-Z)
=====
```

| ID | Nama | No Telp |
|----|-------|--------------|
| 1 | Ahmad | 081234567890 |
| 2 | Budi | 081234567891 |
| 5 | Eka | 081234567894 |
| 3 | Sandi | 081234567892 |
| 6 | Sarah | 081234567895 |

```
=====
```

- Mencari transaksi berdasarkan nama pelanggan

```
public static void cariTransaksi() {
    header();
    System.out.println(x:"      Cari Transaksi Berdasarkan Nama Pelanggan");
    header();
    System.out.print(s:"Masukkan Nama Pelanggan  : ");
    input18.nextLine();
    String namaPelanggan = input18.nextLine();
    System.out.println(x:"-----");
    System.out.println(x:"| ID\t| Nama Pelanggan\t| Jenis Layanan\t| Berat/pcs\t| Harga\t|");
    System.out.println(x:"-----");
    for (Transaksi_18 transaksi : listTransaksi) {
        if (transaksi.Pelanggan.namaPelanggan.equalsIgnoreCase(namaPelanggan)) {
            System.out.println(transaksi);
        }
    }
    System.out.println(x:"-----");
}
```

```
=====
Masukkan pilihan      : 8
=====
      Cari Transaksi Berdasarkan Nama Pelanggan
=====
Masukkan Nama Pelanggan  : Sandi
```

| ID | Nama Pelanggan | Jenis Layanan | Berat/pcs | Harga |
|----|----------------|---------------------|-----------|---------|
| 2 | Sandi | Cuci Kering Regular | 2.8 | 16800.0 |

```
=====
```

PROGRAM NON COLLECTION

Deskripsi Program

Program dalam bahasa java ini mengimplementasikan struktur data LinkedList untuk membuat sebuah sistem manajemen pada perusahaan laundry menjadi lebih mudah dalam pencatatannya. Program ini memiliki fitur menambahkan pelanggan, menghapus pelanggan, melihat pelanggan, menambahkan transaksi, menghapus transaksi, melihat transaksi, mengurutkan pelanggan berdasarkan nama, serta mencari transaksi dengan nama pelanggan. Pada program ini menggunakan sorting dengan metode insertion sort dan penyimpanan data dengan struktur data LinkedList.

Screenshot Program dan Output

- NodePelanggan_18.java

```
public class NodePelanggan_18 {
    Pelanggan_18 pelanggan;
    NodePelanggan_18 next;

    public NodePelanggan_18(Pelanggan_18 pelanggan, NodePelanggan_18 next) {
        this.pelanggan = pelanggan;
        this.next = next;
    }
}
```

- NodeTransaksi_18.java

```
public class NodeTransaksi_18 {
    Transaksi_18 transaksi;
    NodeTransaksi_18 next;

    public NodeTransaksi_18(Transaksi_18 transaksi, NodeTransaksi_18 next) {
        this.transaksi = transaksi;
        this.next = next;
    }
}
```

- Pelanggan_18.java

```
public class Pelanggan_18 {
    int idPelanggan;
    String namaPelanggan;
    String noTelp;

    public Pelanggan_18(int id, String nama, String noTelp) {
        this.idPelanggan = id;
        this.namaPelanggan = nama;
        this.noTelp = noTelp;
    }

    @Override
    public String toString() {
        return "|" + idPelanggan + "\t|" + namaPelanggan + "\t\t|" + noTelp + "\t\t|";
    }
}
```

- Transaksi_18.java

```
public class Transaksi_18 {
    static int counter = 1;
    int idTransaksi;
    Pelanggan_18 pelanggan;
    String jenisLayanan;
    double jumlah;
    double harga;

    public Transaksi_18(Pelanggan_18 pelanggan, String jenisLayanan, double jumlah, double harga) {
        this.idTransaksi = counter++;
        this.pelanggan = pelanggan;
        this.jenisLayanan = jenisLayanan;
        this.jumlah = jumlah;
        this.harga = harga;
    }

    @Override
    public String toString() {
        return "|" + idTransaksi + "|" + pelanggan.namaPelanggan + "|" + jenisLayanan + "|" + jumlah
            + "|" + harga + "|";
    }
}
```

● LinkedListPelanggan_18.java

```
1 package ProgramNonCollection;
2
3 public class LinkedListPelanggan_18 {
4     NodePelanggan_18 head;
5
6     public boolean isEmpty() {
7         return (head == null);
8     }
9
10    public void print() {
11        if (!isEmpty()) {
12            NodePelanggan_18 currentNode = head;
13            while (currentNode != null) {
14                System.out.println(currentNode.pelanggan);
15                currentNode = currentNode.next;
16            }
17        } else {
18            System.out.println("Linked List Kosong");
19        }
20    }
21
22    public void addFirst(Pelanggan_18 pelanggan) {
23        NodePelanggan_18 newNode = new NodePelanggan_18(pelanggan, null);
24        if (isEmpty()) {
25            head = newNode;
26        } else {
27            newNode.next = head;
28            head = newNode;
29        }
30    }
31
32    public void addLast(Pelanggan_18 pelanggan) {
33        NodePelanggan_18 newNode = new NodePelanggan_18(pelanggan, null);
34        if (isEmpty()) {
35            head = newNode;
36        } else {
37            NodePelanggan_18 currentNode = head;
38            while (currentNode.next != null) {
39                currentNode = currentNode.next;
40            }
41            currentNode.next = newNode;
42        }
43    }
44
45    public void removeFirst() {
46        if (!isEmpty()) {
47            head = head.next;
48        } else {
49            System.out.println("Linked List Kosong");
50        }
51    }
52
53    public void removeLast() {
54        if (isEmpty()) {
55            System.out.println("Linked List Kosong");
56        } else if (head.next == null) {
57            head = null;
58        } else {
59            NodePelanggan_18 currentNode = head;
60            while (currentNode.next.next != null) {
61                currentNode = currentNode.next;
62            }
63            currentNode.next = null;
64        }
65    }
66
67    public void remove(int key) {
68        if (isEmpty()) {
69            System.out.println("Linked List kosong");
70        } else if (head.pelanggan.idPelanggan == key) {
71            removeFirst();
72        } else {
73            NodePelanggan_18 currentNode = head;
74            while (currentNode.next != null) {
75                if (currentNode.next.pelanggan.idPelanggan == key) {
76                    currentNode.next = currentNode.next.next;
77                    break;
78                }
79                currentNode = currentNode.next;
80            }
81        }
82    }
83
84    public Pelanggan_18 findById(int id) {
85        NodePelanggan_18 currentNode = head;
86        while (currentNode != null) {
87            if (currentNode.pelanggan.idPelanggan == id) {
88                return currentNode.pelanggan;
89            }
90            currentNode = currentNode.next;
91        }
92        return null;
93    }
94
95    public void sortByName() {
96        if (!isEmpty() && head.next != null) {
97            NodePelanggan_18 sortedHead = null;
98            NodePelanggan_18 currentNode = head;
99            while (currentNode != null) {
100                NodePelanggan_18 nextNode = currentNode.next;
101                sortedHead = sortedInsert(sortedHead, currentNode);
102                currentNode = nextNode;
103            }
104            head = sortedHead;
105        }
106    }
107
108    private NodePelanggan_18 sortedInsert(NodePelanggan_18 sortedHead, NodePelanggan_18 newNode) {
109        if (sortedHead == null || sortedHead.pelanggan.namaPelanggan.compareTo(newNode.pelanggan.namaPelanggan) >= 0) {
110            newNode.next = sortedHead;
111            return newNode;
112        } else {
113            NodePelanggan_18 current = sortedHead;
114            while (current.next != null
115                && current.next.pelanggan.namaPelanggan.compareTo(newNode.pelanggan.namaPelanggan) < 0) {
116                current = current.next;
117            }
118            newNode.next = current.next;
119            current.next = newNode;
120            return sortedHead;
121        }
122    }
123 }
```

● LinkedListTransaksi_18.java

```
1 package ProgramNonCollection;
2
3 public class LinkedListTransaksi_18 {
4     NodeTransaksi_18 head;
5
6     public boolean isEmpty() {
7         return (head == null);
8     }
9
10    public void print() {
11        if (!isEmpty()) {
12            NodeTransaksi_18 currentNode = head;
13            while (currentNode != null) {
14                System.out.println(currentNode.transaksi);
15                currentNode = currentNode.next;
16            }
17        } else {
18            System.out.println("Linked List Kosong");
19        }
20    }
21
22    public void addFirst(Transaksi_18 transaksi) {
23        NodeTransaksi_18 newNode = new NodeTransaksi_18(transaksi, null);
24        if (isEmpty()) {
25            head = newNode;
26        } else {
27            newNode.next = head;
28            head = newNode;
29        }
30    }
31
32    public void addLast(Transaksi_18 transaksi) {
33        NodeTransaksi_18 newNode = new NodeTransaksi_18(transaksi, null);
34        if (isEmpty()) {
35            head = newNode;
36        } else {
37            NodeTransaksi_18 currentNode = head;
38            while (currentNode.next != null) {
39                currentNode = currentNode.next;
40            }
41            currentNode.next = newNode;
42        }
43    }
44
45    public void removeFirst() {
46        if (isEmpty()) {
47            head = head.next;
48        } else {
49            System.out.println("Linked List Kosong");
50        }
51    }
52
53    public void removeLast() {
54        if (isEmpty()) {
55            System.out.println("Linked List Kosong");
56        } else if (head.next == null) {
57            head = null;
58        } else {
59            NodeTransaksi_18 currentNode = head;
60            while (currentNode.next.next != null) {
61                currentNode = currentNode.next;
62            }
63            currentNode.next = null;
64        }
65    }
66
67    public void remove(int key) {
68        if (isEmpty()) {
69            System.out.println("Linked List kosong");
70        } else if (head.transaksi.idTransaksi == key) {
71            removeFirst();
72        } else {
73            NodeTransaksi_18 currentNode = head;
74            while (currentNode.next != null) {
75                if (currentNode.next.transaksi.idTransaksi == key) {
76                    currentNode.next = currentNode.next.next;
77                    break;
78                }
79                currentNode = currentNode.next;
80            }
81        }
82    }
83
84    public Transaksi_18 findById(int id) {
85        NodeTransaksi_18 currentNode = head;
86        while (currentNode != null) {
87            if (currentNode.transaksi.idTransaksi == id) {
88                return currentNode.transaksi;
89            }
90            currentNode = currentNode.next;
91        }
92        return null;
93    }
94
95    public void sortByName() {
96        if (!isEmpty() && head.next != null) {
97            NodeTransaksi_18 sortedHead = null;
98            NodeTransaksi_18 currentNode = head;
99            while (currentNode != null) {
100                NodeTransaksi_18 nextNode = currentNode.next;
101                sortedHead = sortedInsert(sortedHead, currentNode);
102                currentNode = nextNode;
103            }
104            head = sortedHead;
105        }
106    }
107
108    private NodeTransaksi_18 sortedInsert(NodeTransaksi_18 sortedHead, NodeTransaksi_18 newNode) {
109        if (sortedHead == null || sortedHead.transaksi.pelanggan.namaPelanggan.compareTo(newNode.transaksi.pelanggan.namaPelanggan) >= 0) {
110            newNode.next = sortedHead;
111            return newNode;
112        } else {
113            NodeTransaksi_18 current = sortedHead;
114            while (current.next != null && current.next.transaksi.pelanggan.namaPelanggan.compareTo(newNode.transaksi.pelanggan.namaPelanggan) < 0) {
115                current = current.next;
116            }
117            newNode.next = current.next;
118            current.next = newNode;
119            return sortedHead;
120        }
121    }
122 }
```


- Import java Scanner

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

- Inisialisasi LinkedList

```
public static LinkedListPelanggan_18 listPelanggan = new LinkedListPelanggan_18();
public static LinkedListTransaksi_18 listTransaksi = new LinkedListTransaksi_18();
```

- Menambahkan data pelanggan

```
listPelanggan.addLast(new Pelanggan_18(id:1, nama:"Rahmat", noTelp:"081234567890"));
listPelanggan.addLast(new Pelanggan_18(id:2, nama:"Eko", noTelp:"081234567891"));
listPelanggan.addLast(new Pelanggan_18(id:3, nama:"Imam", noTelp:"081234567892"));
listPelanggan.addLast(new Pelanggan_18(id:4, nama:"Aura", noTelp:"081234567893"));
listPelanggan.addLast(new Pelanggan_18(id:5, nama:"Bella", noTelp:"081234567894"));
```

- Tampilan Menu

```
header();
System.out.println(x:"          LAUNDRY JAYA MANDIRI");
header();
System.out.println(x:"1. Tambah Pelanggan");
System.out.println(x:"2. Hapus Pelanggan");
System.out.println(x:"3. Lihat Pelanggan");
System.out.println(x:"4. Tambah Transaksi");
System.out.println(x:"5. Hapus Transaksi");
System.out.println(x:"6. Lihat Transaksi");
System.out.println(x:"7. Urutkan Pelanggan Berdasarkan Nama (A-Z)");
System.out.println(x:"8. Cari Transaksi Berdasarkan Nama Pelanggan");
System.out.println(x:"9. Keluar");
header();
System.out.print(s:"Masukkan pilihan      : ");
pilihan = input18.nextInt();
```

```
=====
                        LAUNDRY JAYA MANDIRI
=====
1. Tambah Pelanggan
2. Hapus Pelanggan
3. Lihat Pelanggan
4. Tambah Transaksi
5. Hapus Transaksi
6. Lihat Transaksi
7. Urutkan Pelanggan Berdasarkan Nama (A-Z)
8. Cari Transaksi Berdasarkan Nama Pelanggan
9. Keluar
=====
```

- Tambah Pelanggan

```
public static void tambahPelanggan() {
    header();
    System.out.println(x:"          Tambah Pelanggan");
    header();
    System.out.print(s:"Masukkan ID Pelanggan      : ");
    int id = input18.nextInt();
    input18.nextLine();
    System.out.print(s:"Masukkan Nama Pelanggan      : ");
    String nama = input18.nextLine();
    System.out.print(s:"Masukkan No Telp Pelanggan      : ");
    String noTelp = input18.nextLine();
    listPelanggan.addLast(new Pelanggan_18(id, nama, noTelp));
    System.out.println(x:"Pelanggan berhasil ditambahkan!");
}
```

```
=====
Masukkan pilihan      : 1
=====
          Tambah Pelanggan
=====
Masukkan ID Pelanggan      : 6
Masukkan Nama Pelanggan      : Putri
Masukkan No Telp Pelanggan      : 081234567895
Pelanggan berhasil ditambahkan!
```

- Hapus Pelanggan

```
public static void hapusPelanggan() {
    header();
    System.out.println(x:"          Hapus Pelanggan");
    header();
    lihatPelanggan();
    System.out.print(s:"Masukkan ID Pelanggan yang akan dihapus      : ");
    int id = input18.nextInt();
    listPelanggan.remove(id);
    System.out.println(x:"Pelanggan berhasil dihapus!");
}
```

```
=====
Masukkan pilihan      : 2
=====
          Hapus Pelanggan
=====
| ID | Nama      | No Telp      |
|----|-----|-----|
| 1  | Rahma     | 081234567890 |
| 2  | Eko       | 081234567891 |
| 3  | Imam     | 081234567892 |
| 4  | Aura      | 081234567893 |
| 5  | Bella     | 081234567894 |
| 6  | Putri     | 081234567895 |
|----|-----|-----|
Masukkan ID Pelanggan yang akan dihapus      : 2
Pelanggan berhasil dihapus!
```

- Lihat Pelanggan

```
public static void lihatPelanggan() {  
    System.out.println(x:"-----");  
    System.out.println(x:"| ID\t| Nama\t\t\t| No Telp\t\t\t|");  
    System.out.println(x:"-----");  
    listPelanggan.print();  
    System.out.println(x:"-----");  
}
```

| | | | |
|----------------------|-------|--------------|--|
| ===== | | | |
| Masukkan pilihan : 3 | | | |
| ----- | | | |
| ID | Nama | No Telp | |
| ----- | | | |
| 1 | Rahma | 081234567890 | |
| 3 | Imam | 081234567892 | |
| 4 | Aura | 081234567893 | |
| 5 | Bella | 081234567894 | |
| 6 | Putri | 081234567895 | |
| ----- | | | |

- Tambah Transaksi

```
public static void tambahTransaksi() {
    header();
    System.out.println(x:"          Tambah Transaksi");
    header();
    lihatPelanggan();
    System.out.print(s:"Masukkan ID Pelanggan          : ");
    int idPelanggan = input18.nextInt();
    Pelanggan_18 pelanggan = listPelanggan.findById(idPelanggan);
    if (pelanggan == null) {
        System.out.println(x:"Pelanggan tidak ditemukan!");
        return;
    }

    String jenisLayanan = pilihLayanan();
    System.out.print(s:"Masukkan berat atau jumlah pcs      : ");
    double jumlah = input18.nextDouble();
    double harga = hitungHarga(jenisLayanan, jumlah);

    listTransaksi.addLast(new Transaksi_18(pelanggan, jenisLayanan, jumlah, harga));
    System.out.println(x:"Transaksi berhasil ditambahkan!");
}
```

```
=====
Masukkan pilihan          : 4
=====
                          Tambah Transaksi
=====
-----
| ID | Nama          | No Telp          |
-----
| 1  | Rahma         | 081234567890    |
| 3  | Imam         | 081234567892    |
| 4  | Aura         | 081234567893    |
| 5  | Bella        | 081234567894    |
| 6  | Putri        | 081234567895    |
-----
Masukkan ID Pelanggan          : 3
=====
                          Jenis Layanan
=====
1. Cuci Kering Regular (6000/kg)
2. Cuci Kering Ekspres (9000/kg)
3. Cuci Setrika Regular (8000/kg)
4. Cuci Setrika Ekspres (12000/kg)
5. Cuci Sepatu (35000/pcs)
6. Cuci Boneka (10000/pcs)
=====
Pilih jenis layanan          : 2
Masukkan berat atau jumlah pcs : 4.6
Transaksi berhasil ditambahkan!
```


- Hapus Transaksi

```
public static void hapusTransaksi() {
    header();
    System.out.println(x:"          Hapus Transaksi");
    header();
    lihatTransaksi();
    System.out.print(s:"Masukkan ID Transaksi yang akan dihapus   : ");
    int id = input18.nextInt();
    listTransaksi.remove(id);
    System.out.println(x:"Transaksi berhasil dihapus!");
}
```

```
=====
Masukkan pilihan      : 5
=====
                        Hapus Transaksi
=====
```

| ID | Nama Pelanggan | Jenis Layanan | Berat/pcs | Harga |
|----|----------------|---------------------|-----------|---------|
| 1 | Rahma | Cuci Boneka | 2.0 | 20000.0 |
| 2 | Imam | Cuci Kering Ekspres | 4.6 | 41400.0 |

```
=====
Masukkan ID Transaksi yang akan dihapus   : 1
Transaksi berhasil dihapus!
```

- Lihat Transaksi

```
public static void lihatTransaksi() {
    System.out.println(x:"-----");
    System.out.println(x:"| ID\t| Nama Pelanggan\t| Jenis Layanan\t\t| Berat/pcs\t| Harga\t\t|");
    System.out.println(x:"-----");
    listTransaksi.print();
    System.out.println(x:"-----");
}
```

```
=====
Masukkan pilihan      : 6
=====
```

| ID | Nama Pelanggan | Jenis Layanan | Berat/pcs | Harga |
|----|----------------|---------------------|-----------|---------|
| 2 | Imam | Cuci Kering Ekspres | 4.6 | 41400.0 |

```
=====
```


- Urutkan berdasarkan nama pelanggan dengan A - Z

```
public static void urutkanPelanggan() {
    header();
    System.out.println(x:"    Urutkan Pelanggan Berdasarkan Nama (A-Z)");
    header();
    listPelanggan.sortByName();
    lihatPelanggan();
}
```

```
=====
Masukkan pilihan      : 7
=====
    Urutkan Pelanggan Berdasarkan Nama (A-Z)
=====
```

| ID | Nama | No Telp |
|----|-------|--------------|
| 4 | Aura | 081234567893 |
| 5 | Bella | 081234567894 |
| 3 | Imam | 081234567892 |
| 6 | Putri | 081234567895 |
| 1 | Rahma | 081234567890 |

- Cari transaksi berdasarkan nama pelanggan

```
public static void cariTransaksi() {
    header();
    System.out.println(x:"    Cari Transaksi Berdasarkan Nama Pelanggan");
    header();
    System.out.print(s:"Masukkan Nama Pelanggan  : ");
    input18.nextLine();
    String namaPelanggan = input18.nextLine();
    System.out.println(x:"-----");
    System.out.println(x:"| ID\t| Nama Pelanggan\t| Jenis Layanan\t| Berat/pcs\t| Harga\t\t|");
    System.out.println(x:"-----");
    NodeTransaksi_18 currentNode = listTransaksi.head;
    while (currentNode != null) {
        if (currentNode.transaksi.pelanggan.namaPelanggan.equalsIgnoreCase(namaPelanggan)) {
            System.out.println(" " + currentNode.transaksi.pelanggan.idPelanggan + "\t| "
                + currentNode.transaksi.pelanggan.namaPelanggan + "\t| "
                + currentNode.transaksi.jenisLayanan + "\t| " + currentNode.transaksi.jumlah + "\t| "
                + currentNode.transaksi.harga + "\t|");
        }
        currentNode = currentNode.next;
    }
    System.out.println(x:"-----");
}
```

```
=====
Masukkan pilihan      : 8
=====
    Cari Transaksi Berdasarkan Nama Pelanggan
=====
Masukkan Nama Pelanggan  : Imam
```

| ID | Nama Pelanggan | Jenis Layanan | Berat/pcs | Harga |
|----|----------------|---------------------|-----------|---------|
| 2 | Imam | Cuci Kering Ekspres | 4.6 | 41400.0 |

Repository :

https://github.com/zidnafaz/Praktikum-Algoritma-Struktur-Data/tree/main/KUIS_2