

POSSTEST 6

NAMA: MUHAMAD ALFARIZI EDYTIA

NIM : 2309106105

KELAS: C1 23

```
D: > Downloads > J LaporanList.java > LaporanList > hapusData()
1  import java.util.ArrayList;
2  import java.util.Iterator;
3  import java.util.Scanner;
4
5  abstract class Laporan {
6      private final String id;
7      private String namaItem;
8      private double harga;
9
10     public Laporan(String id, String namaItem, double harga) {
11         this.id = id;
12         this.namaItem = namaItem;
13         this.harga = harga;
14     }
15
16     public String getId() {
17         return id;
18     }
19
20     public String getNamaItem() {
21         return namaItem;
22     }
23
24     public void setNamaItem(String namaItem) {
25         this.namaItem = namaItem;
26     }
27
28     public double getHarga() {
29         return harga;
30     }
31 }
```

```

5  abstract class Laporan {
28      public double getHarga() {
30      }
31
32      public void setHarga(double harga) {
33          this.harga = harga;
34      }
35
36      public abstract String getJenis();
37
38      public final void info() {
39          System.out.println(x:"Data laporan finansial berhasil diproses.");
40      }
41
42      @Override
43      public String toString() {
44          return "ID: " + id +
45              " | Jenis: " + getJenis() +
46              " | Nama Item: " + namaItem +
47              " | Harga: Rp " + harga;
48      }
49  }
50
51  interface CetakLaporan {
52      void cetakDetail();
53      void cetakRingkas();
54  }
55
56  class Util {

```

Downloads > LaporanList.java > LaporanList > hapusData()

```
6 class Util {
7     public static double totalHarga = 0;
8
9     public static boolean validasiHarga(double harga) {
10         return harga >= 0;
11     }
12
13     public static void updateTotal(double harga) {
14         totalHarga += harga;
15     }
16 }
17
18 class LaporanPenjualan extends Laporan implements CetakLaporan {
19     public LaporanPenjualan(String id, String namaItem, double harga) {
20         super(id, namaItem, harga);
21     }
22
23     @Override
24     public String getJenis() {
25         return "Penjualan";
26     }
27
28     @Override
29     public void cetakDetail() {
30         System.out.println("[DETAIL] " + toString());
31     }
32
33     @Override
34     public void cetakRingkas() {
```

```

68 class LaporanPenjualan extends Laporan implements CetakLaporan {
84     public void cetakRingkas() {
85         System.out.println("[RINGKAS] Penjualan - " + getNamaItem() + ": Rp " + getHarga());
86     }
87 }
88
89 final class LaporanPengeluaran extends Laporan implements CetakLaporan {
90     public LaporanPengeluaran(String id, String namaItem, double harga) {
91         super(id, namaItem, harga);
92     }
93
94     @Override
95     public String getJenis() {
96         return "Pengeluaran";
97     }
98
99     @Override
100    public void cetakDetail() {
101        System.out.println("[DETAIL] " + toString());
102    }
103
104    @Override
105    public void cetakRingkas() {
106        System.out.println("[RINGKAS] Pengeluaran - " + getNamaItem() + ": Rp " + getHarga());
107    }
108 }
109
110 public class LaporanList {
111     private static ArrayList<Laporan> laporanList = new ArrayList<>();

```

```

114     public static void main(String[] args) {
138         break;
139     default:
140         System.out.println(x:"Pilihan tidak valid! Coba lagi.");
141     }
142     } catch (Exception e) {
143         System.out.println(x:"Input harus berupa angka! Coba lagi.");
144     }
145 }
146 }
147
148 private static void tampilkanMenu() {
149     System.out.println(x:"\n=== Menu Sistem Laporan Finansial Cafe Syafar ===");
150     System.out.println(x:"1. Tambah Data");
151     System.out.println(x:"2. Tampilkan Data");
152     System.out.println(x:"3. Ubah Data");
153     System.out.println(x:"4. Hapus Data");
154     System.out.println(x:"5. Tampilkan Total Harga");
155     System.out.println(x:"6. Keluar");
156     System.out.print(s:"Pilih menu (1-6): ");
157 }
158
159 private static void tambahData() {
160     System.out.println(x:"\n=== Tambah Data ===");
161     try {
162         System.out.print(s:"Pilih jenis data (1. Penjualan, 2. Pengeluaran): ");
163         int jenis = Integer.parseInt(scanner.nextLine());
164         if (jenis != 1 && jenis != 2) {

```

```

0 public class LaporanList {
1     private static void tambahData() {
2
3         throw new IllegalArgumentException(s:"Hanya boleh memilih 1 atau 2.");
4     }
5
6     System.out.print(s:"Masukkan ID: ");
7     String id = scanner.nextLine();
8     System.out.print(s:"Masukkan Nama Item: ");
9     String namaItem = scanner.nextLine();
10    System.out.print(s:"Masukkan Harga: Rp ");
11    double harga = Double.parseDouble(scanner.nextLine());
12
13    if (!Util.validasiHarga(harga)) {
14        throw new IllegalArgumentException(s:"Harga tidak boleh negatif.");
15    }
16
17    Laporan laporanBaru = (jenis == 1)
18        ? new LaporanPenjualan(id, namaItem, harga)
19        : new LaporanPengeluaran(id, namaItem, harga);
20
21    laporanList.add(laporanBaru);
22    Util.updateTotal(harga);
23    laporanBaru.info();
24    System.out.println(x:"Data berhasil ditambahkan.");
25
26 } catch (Exception e) {
27     System.out.println("Terjadi kesalahan: " + e.getMessage());
28 }
29

```

```

10 public class LaporanList {
11
12
13     private static void tampilkanData() {
14         System.out.println(x:"\n=== Tampilkan Data ===");
15         if (laporanList.isEmpty()) {
16             System.out.println(x:"Tidak ada data untuk ditampilkan.");
17         } else {
18             for (Laporan laporan : laporanList) {
19                 if (laporan instanceof CetakLaporan) {
20                     ((CetakLaporan) laporan).cetakDetail();
21                 }
22             }
23         }
24     }
25
26     private static void ubahData() {
27         System.out.println(x:"\n=== Ubah Data ===");
28         System.out.print(s:"Masukkan ID Laporan yang ingin diubah: ");
29         String id = scanner.nextLine();
30
31         boolean ditemukan = false;
32         for (Laporan laporan : laporanList) {
33             if (laporan.getId().equals(id)) {
34                 ditemukan = true;
35                 try {
36                     System.out.print(s:"Masukkan Nama Item Baru: ");
37                     String namaItemBaru = scanner.nextLine();
38                     System.out.print(s:"Masukkan Harga Baru: Rp ");
39                     double hargaBaru = Double.parseDouble(scanner.nextLine());

```

```

        if (!Util.validasiHarga(hargaBaru)) {
            throw new IllegalArgumentException(s:"Harga tidak boleh negatif.");
        }

        Util.totalHarga -= laporan.getHarga();
        laporan.setNamaItem(namaItemBaru);
        laporan.setHarga(hargaBaru);
        Util.updateTotal(hargaBaru);

        System.out.println(x:"Data berhasil diubah.");
    } catch (Exception e) {
        System.out.println("Gagal mengubah data: " + e.getMessage());
    }
    break;
}

}

if (!ditemukan) {
    System.out.println("Data dengan ID " + id + " tidak ditemukan.");
}
}

private static void hapusData() {
    System.out.println(x:"\n=== Hapus Data ===");
    System.out.print(s:"Masukkan ID Laporan yang ingin dihapus: ");
    String id = scanner.nextLine();

```

```

private static void ubahData() {
    boolean ditemukan = false;
    for (Laporan laporan : laporanList) {
        if (laporan.getId().equals(id)) {
            ditemukan = true;
            try {
                System.out.print(s:"Masukkan Nama Item Baru: ");
                String namaItemBaru = scanner.nextLine();
                System.out.print(s:"Masukkan Harga Baru: Rp ");
                double hargaBaru = Double.parseDouble(scanner.nextLine());

                if (!Util.validasiHarga(hargaBaru)) {
                    throw new IllegalArgumentException(s:"Harga tidak boleh negatif.");
                }

                Util.totalHarga -= laporan.getHarga();
                laporan.setNamaItem(namaItemBaru);
                laporan.setHarga(hargaBaru);
                Util.updateTotal(hargaBaru);

                System.out.println(x:"Data berhasil diubah.");
            } catch (Exception e) {
                System.out.println("Gagal mengubah data: " + e.getMessage());
            }
            break;
        }
    }
}

```



```

10 public class LaporanList {
06     private static void ubahData() {
39         System.out.println("Data dengan ID " + id + " tidak ditemukan.");
40     }
41 }
42
43 private static void hapusData() {
44     System.out.println(x:"\n=== Hapus Data ===");
45     System.out.print(s:"Masukkan ID Laporan yang ingin dihapus: ");
46     String id = scanner.nextLine();
47
48     Iterator<Laporan> iterator = laporanList.iterator();
49     boolean ditemukan = false;
50
51     while (iterator.hasNext()) {
52         Laporan laporan = iterator.next();
53         if (laporan.getId().equals(id)) {
54             Util.totalHarga -= laporan.getHarga();
55             iterator.remove();
56             ditemukan = true;
57             System.out.println(x:"Data berhasil dihapus.");
58             break;
59         }
60     }
61
62     if (!ditemukan) {
63         System.out.println("Data dengan ID " + id + " tidak ditemukan.");
64     }
65 }
66 }

```