

EXPLORER

PT3

> .vscode

> bin

> lib

> src

Accessory.class

Accessory.java

Clothing.class

Clothing.java

Main.class

Main.java

ThriftItem.class

ThriftItem.java

README.md

ThriftItem.java

src > ThriftItem.java > ThriftItem > getUkuran()

```
1 public abstract class ThriftItem {
2     protected String jenis;
3     protected String ukuran;
4     protected String kategori;
5
6     public ThriftItem(String jenis, String ukuran, String kategori) {
7         this.jenis = jenis;
8         this.ukuran = ukuran;
9         this.kategori = kategori;
10    }
11
12    public String getJenis() {
13        return jenis;
14    }
15
16    public String getUkuran() {
17        return ukuran;
18    }
19
20    public String getKategori() {
21        return kategori;
22    }
23
24    // Method yang akan di-override
25    public abstract void displayDetails();
26 }
27
```

EXPLORER

PT3

> .vscode

> bin

> lib

> src

Accessory.class

Accessory.java

Clothing.class

Clothing.java

Main.class

Main.java

ThriftItem.class

ThriftItem.java

README.md

Clothing.java

src > Clothing.java > ...

```
1 public class Clothing extends ThriftItem {
2     private String material;
3
4     public Clothing(String jenis, String ukuran, String kategori, String material) {
5         super(jenis, ukuran, kategori);
6         this.material = material;
7     }
8
9     @Override
10    public void displayDetails() {
11        System.out.printf(format:"%-20s | %-15s | %-20s | Material: %s\n", jenis, ukuran, kategori, material);
12    }
13 }
14
```

EXPLORER

PT3

- .vscode
- bin
- lib
- src
 - Accessory.class
 - Accessory.java
 - Clothing.class
 - Clothing.java
 - Main.class
 - Main.java
 - ThriftItem.class
 - ThriftItem.java
- README.md

src > J Accessory.java > ...

```
1 public class Accessory extends ThriftItem {
2     private String color;
3
4     public Accessory(String jenis, String ukuran, String kategori, String color) {
5         super(jenis, ukuran, kategori);
6         this.color = color;
7     }
8
9     @Override
10    public void displayDetails() {
11        System.out.printf(format:"%-20s | %-15s | %-20s | Warna: %s%n", jenis, ukuran, kategori, color);
12    }
13 }
14
```

EXPLORER

PT3

- .vscode
- bin
- lib
- src
 - Accessory.class
 - Accessory.java
 - Clothing.class
 - Clothing.java
 - Main.class
 - Main.java
 - ThriftItem.class
 - ThriftItem.java
- README.md

src > J Main.java > Main

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Main {
5     private static InputStreamReader p = new InputStreamReader(System.in);
6     private static BufferedReader input = new BufferedReader(p);
7     private static ArrayList<ThriftItem> DataThrift = new ArrayList<>();
8
9     Run | Debug
10    public static void main(String[] args) throws Exception {
11        while (true) {
12            System.out.println(x:"\nMenu Utama");
13            System.out.println(x:"1. Tambah Data");
14            System.out.println(x:"2. Tampilkan Data");
15            System.out.println(x:"3. Update Data");
16            System.out.println(x:"4. Hapus Data");
17            System.out.println(x:"5. Keluar");
18            System.out.print(s:"Masukkan pilihan >>> ");
19            int pilihan = Integer.parseInt(input.readLine());
20
21            switch (pilihan) {
22                case 1:
23                    tambahData();
24                    break;
25                case 2:
26                    displayData();
27                    break;
28                case 3:
29                    updateData();
30                    break;
31                case 4:
32                    deleteData();
33                    break;
34                case 5:
35                    System.out.println(x:"Keluar program...");
36                    System.exit(status:0);
37            }
38        }
39    }
40}
```

EXPLORERPT3> .vscode> bin> lib> srcJ Accessory.classJ Accessory.javaJ Clothing.classJ Clothing.javaJ Main.classJ Main.javaJ ThriftItem.classJ ThriftItem.java① README.md> TIMELINE> JAVA PROJECTS

J ThriftItem.javaJ Clothing.javaJ Accessory.javaJ Main.java XJ Clothing.class

src > J Main.java > Main

```
4 public class Main {
9     public static void main(String[] args) throws Exception {
35         System.exit(status:0);
36         break;
37     default:
38         System.out.println(x:"Pilihan tidak ada. Silakan coba lagi.");
39         break;
40     }
41 }
42 }
43
44 static void tambahData() throws IOException {
45     String lanjut;
46     do {
47         System.out.println(x:"\nTambah Data");
48         System.out.println(x:"Pilih jenis item:");
49         System.out.println(x:"1. Pakaian");
50         System.out.println(x:"2. Aksesoris");
51         System.out.print(s:"Pilihan Anda >>> ");
52         int jenisItem = Integer.parseInt(input.readLine());
53
54         String jenis = "", ukuran = "", kategori = "", material = "", color = "";
55
56         while (jenis.trim().isEmpty()) {
57             System.out.print(s:"Masukkan jenis >>> ");
58             jenis = input.readLine();
59             if (jenis.trim().isEmpty()) {
60                 System.out.println(x:"Jenis tidak boleh kosong.");
61             }
62         }
63
64         while (ukuran.trim().isEmpty()) {
65             System.out.print(s:"Masukkan ukuran >>> ");
66             ukuran = input.readLine();
67             if (ukuran.trim().isEmpty()) {
68                 System.out.println(x:"Ukuran tidak boleh kosong.");
```

EXPLORERPT3> .vscode> bin> lib> srcJ ThriftItem.classJ ThriftItem.java① README.md> TIMELINE> JAVA PROJECTS

J ThriftItem.javaJ Clothing.javaJ Accessory.javaJ Main.java XJ Clothing.class

src > J Main.java > Main
4 public class Main {
44 static void tambahData() throws IOException {
68 System.out.println(x:"Ukuran tidak boleh kosong.");
69 }
70 }
71
72 while (kategori.trim().isEmpty()) {
73 System.out.print(s:"Masukkan kategori >>> ");
74 kategori = input.readLine();
75 if (kategori.trim().isEmpty()) {
76 System.out.println(x:"Kategori tidak boleh kosong.");
77 }
78 }
79
80 ThriftItem item = null;
81 switch (jenisItem) {
82 case 1:
83 System.out.print(s:"Masukkan material >>> ");
84 material = input.readLine();
85 item = new Clothing(jenis, ukuran, kategori, material);
86 break;
87 case 2:
88 System.out.print(s:"Masukkan warna >>> ");
89 color = input.readLine();
90 item = new Accessory(jenis, ukuran, kategori, color);
91 break;
92 default:
93 System.out.println(x:"Pilihan tidak valid.");
94 break;
95 }
96
97 if (item != null) {
98 DataThrift.add(item);
99 System.out.println(x:"Data telah ditambahkan!");
100 }
101

EXPLORERPT3> .vscode> bin> lib> srcJ Accessory.classJ Accessory.javaJ Clothing.classJ Clothing.javaJ Main.classJ Main.javaJ ThriftItem.classJ ThriftItem.java① README.md> TIMELINE> JAVA PROJECTS

J ThriftItem.javaJ Clothing.javaJ Accessory.javaJ Main.java XJ Clothing.class

src > J Main.java > Main
4 public class Main {
44 static void tambahData() throws IOException {
99 System.out.println(x:"Data telah ditambahkan!");
100 }
101
102 System.out.print(s:"Tambah data lagi? (y/n) >>> ");
103 lanjut = input.readLine();
104 } while (lanjut.equalsIgnoreCase(anotherString:"y"));
105 }
106
107 static void displayData() {
108 // Kode asli untuk menampilkan semua data
109 }
110
111 // Overloading method displayData untuk menampilkan berdasarkan kategori
112 static void displayData(String kategori) {
113 if (DataThrift.isEmpty()) {
114 System.out.println(x:"Tidak ada data untuk ditampilkan.");
115 return;
116 }
117
118 System.out.printf(format:"%-3s | %-20s | %-15s | %-20s\n", ...args:"No", "Jenis", "Ukuran", "Kategori");
119 System.out.println(x:"-----");
120
121 int nomor = 1;
122 for (ThriftItem trf : DataThrift) {
123 if (trf.getKategori().equalsIgnoreCase(kategori)) {
124 System.out.printf(format:"%-3d | %-20s | %-15s | %-20s\n", nomor++, trf.getJenis(), trf.getUkuran(), trf.getKategori());
125 trf.displayDetails();
126 }
127 }
128 }
129
130 static void updateData() throws IOException {
131 if (DataThrift.isEmpty()) {
132 System.out.println(x:"Tidak ada data untuk diperbarui.");
133 }
134 }
135

EXPLORER

PT3

> .vscode

> bin

> lib

> src

Accessory.class

Accessory.java

Clothing.class

Clothing.java

Main.class

Main.java

ThriftItem.class

ThriftItem.java

README.md

TIMELINE

ThriftItem.java

Clothing.java

Accessory.java

Main.java

Clothing.class

src > Main.java > Main

4 public class Main {

130 static void updateData() throws IOException {

132 System.out.println(x:"Tidak ada data untuk diperbarui.");

133 return;

134 }

135 displayData();

136 System.out.print(s:"Masukkan nomor data yang ingin diupdate: ");

137 int index = Integer.parseInt(input.readLine()) - 1;

138

139 if (index >= 0 && index < DataThrift.size()) {

140 ThriftItem trf = DataThrift.get(index);

141

142 System.out.print(s:"Masukkan jenis baru >>> ");

143 String jenis = input.readLine();

144 System.out.print(s:"Masukkan ukuran baru >>> ");

145 String ukuran = input.readLine();

146 System.out.print(s:"Masukkan kategori baru >>> ");

147 String kategori = input.readLine();

148

149 // Memeriksa tipe dari trf untuk menentukan apakah itu Clothing atau Accessory

150 if (trf instanceof Clothing) {

151 System.out.print(s:"Masukkan material baru >>> ");

152 String material = input.readLine();

153 Clothing updatedItem = new Clothing(jenis, ukuran, kategori, material);

154 DataThrift.set(index, updatedItem);

155 } else if (trf instanceof Accessory) {

156 System.out.print(s:"Masukkan warna baru >>> ");

157 String color = input.readLine();

158 Accessory updatedItem = new Accessory(jenis, ukuran, kategori, color);

159 DataThrift.set(index, updatedItem);

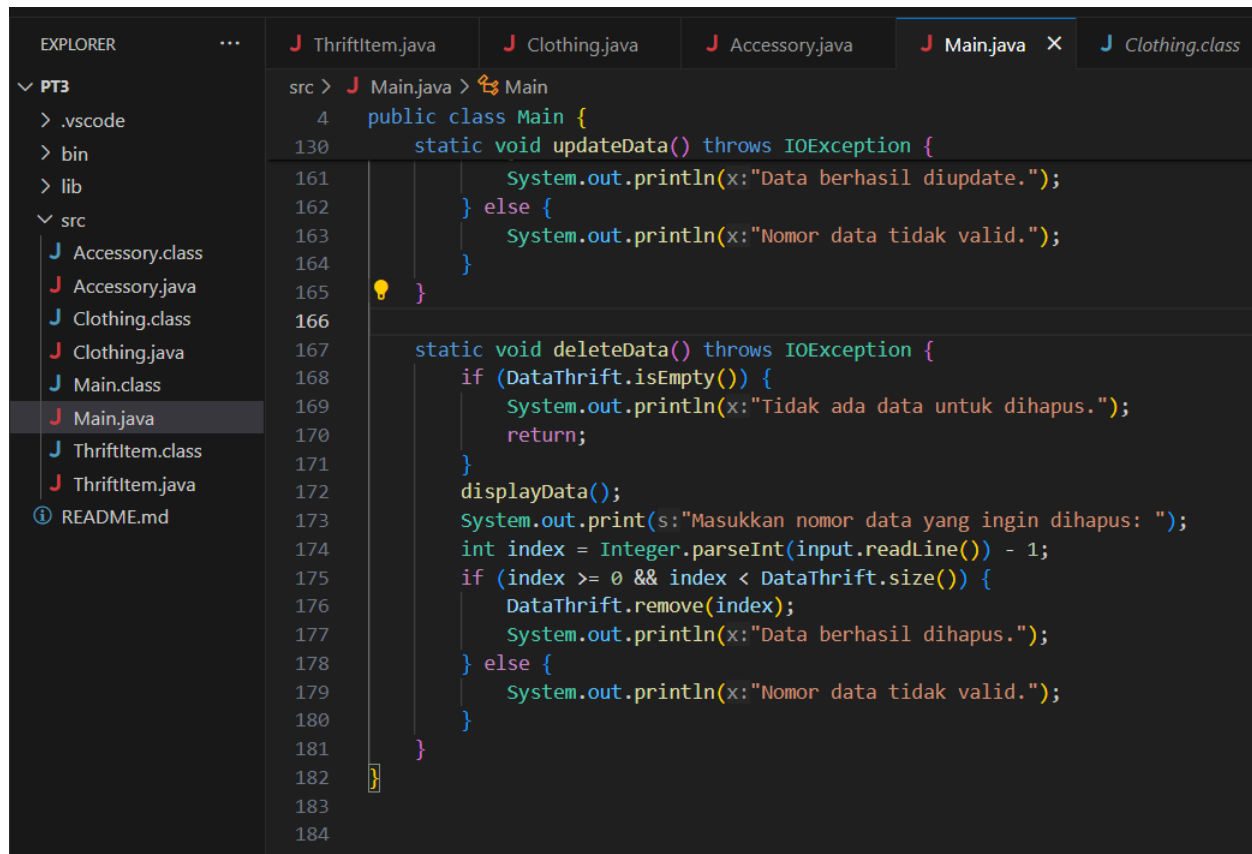
160 }

161 System.out.println(x:"Data berhasil diupdate.");

162 } else {

163 System.out.println(x:"Nomor data tidak valid.");

164 }



The image shows a VS Code editor window with a Java project. The Explorer sidebar on the left shows the project structure: PT3, .vscode, bin, lib, and src. The src folder contains Accessory.class, Accessory.java, Clothing.class, Clothing.java, Main.class, Main.java (selected), ThriftItem.class, and ThriftItem.java. The main editor displays the code for Main.java, which includes two static methods: updateData() and deleteData(). The updateData() method has a bug where it prints "Data berhasil diupdate." for both successful and failed updates. The deleteData() method correctly prints "Tidak ada data untuk dihapus." if the data is empty and "Data berhasil dihapus." if the data is successfully removed. The code is as follows:

```
src > J Main.java > Main
4 public class Main {
130 static void updateData() throws IOException {
161     System.out.println(x:"Data berhasil diupdate.");
162 } else {
163     System.out.println(x:"Nomor data tidak valid.");
164 }
165 }
166
167 static void deleteData() throws IOException {
168     if (DataThrift.isEmpty()) {
169         System.out.println(x:"Tidak ada data untuk dihapus.");
170         return;
171     }
172     displayData();
173     System.out.print(s:"Masukkan nomor data yang ingin dihapus: ");
174     int index = Integer.parseInt(input.readLine()) - 1;
175     if (index >= 0 && index < DataThrift.size()) {
176         DataThrift.remove(index);
177         System.out.println(x:"Data berhasil dihapus.");
178     } else {
179         System.out.println(x:"Nomor data tidak valid.");
180     }
181 }
182 }
183
184
```