

LAPORAN PRAKTIKUM
POSTTEST 6
PEMROGRAMAN BERORIENTASI OBJEK



Disusun oleh:
Tiara Kasma Wati Putri (2309106080)
Kelas (B2 '23)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

```
Mainjava x
Source History
1 import java.util.ArrayList;
2 import java.util.Scanner;
3
4 // Interface Perawatan
5 interface Perawatan {
6     void jadwalServis();
7     void cekKondisi();
8 }
9
10 // Abstract class
11 abstract class AlatBerat implements Perawatan {
12     private int id;
13     private String nama;
14     private int hargaSewa;
15     private static int jumlahAlat = 0; // static variable
16
17     // Constructor
18     public AlatBerat(int id, String nama, int hargaSewa) {
19         this.id = id;
20         this.nama = nama;
21         this.hargaSewa = hargaSewa;
22         jumlahAlat++; // setiap kali objek dibuat, jumlah bertambah
23     }
24
25     // Getter
26     public int getId() {
27         return id;
28     }
29
30     public String getName() {
31         return nama;
32     }
33
34     public int getHargaSewa() {
35         return hargaSewa;
36     }
37
38     // Static method
39     public static int getJumlahAlat() {
40         return jumlahAlat;
41     }
42
43     // Method dari interface Perawatan
44     public void jadwalServis() {
45         System.out.println("Jadwal servis setiap 30 hari.");
46     }
47
48     public void cekKondisi() {
49         System.out.println("Kondisi dicek sebelum dan sesudah penyewaan.");
50     }
51
52     // Method abstrak
53     public abstract void tampilkanInfo();
54 }
55
56 // Kelas turunan Excavator
57 class Excavator extends AlatBerat {
58     private int kapasitasBucket;
59
60     public Excavator(int id, String nama, int hargaSewa, int kapasitasBucket) {
61         super(id, nama, hargaSewa);
62         this.kapasitasBucket = kapasitasBucket;
63     }
64
65     public int getKapasitasBucket() {
66         return kapasitasBucket;
67     }
68
69     @Override
70     public void tampilkanInfo() {
71         System.out.println("ID: " + getId());
72         System.out.println("Nama: " + getName());
73         System.out.println("Harga Sewa: Rp " + getHargaSewa());
74     }
75 }
```

```

74         System.out.println("Kapasitas Bucket: " + kapasitasBucket + " liter");
75         jadwalServis();
76         cekKondisi();
77         System.out.println();
78     }
79 }
80
81 // Kelas turunan Bulldozer
82 class Bulldozer extends AlatBerat {
83     private int kekuatanDorong;
84
85     public Bulldozer(int id, String nama, int hargaSewa, int kekuatanDorong) {
86         super(id, nama, hargaSewa);
87         this.kekuatanDorong = kekuatanDorong;
88     }
89
90     public int getKekuatanDorong() {
91         return kekuatanDorong;
92     }
93
94     @Override
95     public void tampilkanInfo() {
96         System.out.println("ID: " + getId());
97         System.out.println("Nama: " + getName());
98         System.out.println("Harga Sewa: Rp " + getHargaSewa());
99         System.out.println("Kekuatan Dorong: " + kekuatanDorong + " ton");
100        jadwalServis();
101        cekKondisi();
102        System.out.println();
103    }
104 }
105
106 // Kelas final
107 final class ValidasiInput {
108     public static boolean validasiAngka(String input) {
109         return input.matches("\\d+");
110     }
111 }
112
113 // Main class
114 public class Main {
115     private static ArrayList<AlatBerat> daftarAlat = new ArrayList<>();
116     private static Scanner scanner = new Scanner(System.in);
117     private static int idTerakhir = 0;
118
119     public static void main(String[] args) {
120         int pilihan = 0;
121
122         while (pilihan != 4) {
123             System.out.println("===== Sistem Manajemen Alat Berat =====");
124             System.out.println("1. Tambah Alat Berat");
125             System.out.println("2. Lihat Daftar Alat Berat");
126             System.out.println("3. Lihat Total Alat Berat");
127             System.out.println("4. Keluar");
128             System.out.print("Pilih menu: ");
129             String input = scanner.nextLine();
130
131             // Validasi input harus angka
132             if (!ValidasiInput.validasiAngka(input)) {
133                 System.out.println("Input harus berupa angka!");
134                 continue;
135             }
136
137             try {
138                 pilihan = Integer.parseInt(input);
139             } catch (NumberFormatException e) {
140                 System.out.println("Terjadi kesalahan input! " + e.getMessage());
141                 continue;
142             }
143
144             switch (pilihan) {
145                 case 1:
146                     tambahAlatBerat();
147                     break;
148                 case 2:
149                     lihatAlatBerat();
150                     break;
151                 case 3:
152                     System.out.println("Total Alat Berat: " + AlatBerat.getJumlahAlat());
153                     System.out.println();
154                     break;
155                 case 4:
156                     System.out.println("Terima kasih telah menggunakan sistem ini.");
157                     break;
158                 default:
159                     System.out.println("Pilihan tidak valid!");
160                     break;
161             }
162         }
163     }
164
165     private static void tambahAlatBerat() {
166         System.out.println("\nPilih Jenis Alat Berat:");
167         System.out.println("1. Excavator");
168         System.out.println("2. Bulldozer");
169         System.out.print("Pilihan: ");
170         String input = scanner.nextLine();
171
172         if (!ValidasiInput.validasiAngka(input)) {
173             System.out.println("Input harus berupa angka!");
174             return;
175         }
176
177         int jenis;
178         try {
179             jenis = Integer.parseInt(input);
180         } catch (NumberFormatException e) {
181             System.out.println("Terjadi kesalahan input! " + e.getMessage());
182             return;

```

```

183 |
184 |
185 |     System.out.print("Nama Alat: ");
186 |     String nama = scanner.nextLine();
187 |
188 |     System.out.print("Harga Sewa per hari: ");
189 |     int hargaSewa;
190 |     try {
191 |         hargaSewa = Integer.parseInt(scanner.nextLine());
192 |     } catch (NumberFormatException e) {
193 |         System.out.println("Input harga harus berupa angka!");
194 |         return;
195 |     }
196 |
197 |     idTerakhir++;
198 |
199 |     if (jenis == 1) {
200 |         System.out.print("Kapasitas Bucket (liter): ");
201 |         int kapasitas;
202 |         try {
203 |             kapasitas = Integer.parseInt(scanner.nextLine());
204 |         } catch (NumberFormatException e) {
205 |             System.out.println("Input kapasitas harus berupa angka!");
206 |             return;
207 |         }
208 |         daftarAlat.add(new Excavator(idTerakhir, nama, hargaSewa, kapasitas));
209 |     } else if (jenis == 2) {
210 |         System.out.print("Kekuatan Dorong (ton): ");
211 |         int kekuatan;
212 |         try {
213 |             kekuatan = Integer.parseInt(scanner.nextLine());
214 |         } catch (NumberFormatException e) {
215 |             System.out.println("Input kekuatan harus berupa angka!");
216 |             return;
217 |         }
218 |         daftarAlat.add(new Bulldozer(idTerakhir, nama, hargaSewa, kekuatan));
219 |     } else {
220 |         System.out.println("Jenis alat berat tidak valid!");
221 |     }
222 |
223 |     System.out.println("Alat berat berhasil ditambahkan!\n");
224 | }
225 |
226 | private static void LihatAlatBerat() {
227 |     if (daftarAlat.isEmpty()) {
228 |         System.out.println("\nBelum ada data alat berat.\n");
229 |         return;
230 |     }
231 |
232 |     System.out.println("\n==== Daftar Alat Berat =====");
233 |     for (AlatBerat alat : daftarAlat) {
234 |         alat.tampilkanInfo();
235 |     }
236 |     System.out.println("Total Alat Berat: " + AlatBerat.getJumlahAlat());
237 |     System.out.println();
238 | }
239 |
240 |

```