

```

1  import java.util.*;
2
3  // Interface
4  interface Sewa {
5      void sewaUnit();           // method 1
6      void kembalikanUnit();    // method 2
7  }
8
9  // Abstract Class Unit + Interface
10 abstract class Unit implements Sewa {
11     protected final int id; // final attribute
12     private String nama;
13     private double harga;
14     private boolean tersedia;
15
16     public Unit(int id, String nama, double harga, boolean tersedia) {
17         this.id = id;
18         this.nama = nama;
19         this.harga = harga;
20         this.tersedia = tersedia;
21     }
22
23     public final int getId() {
24         return id;
25     }
26
27     public String getName() {
28         return nama;
29     }
30
31     public void setName(String nama) {
32         this.nama = nama;
33     }
34
35     public double getHarga() {
36         return harga;
37     }
38
39     public void setHarga(double harga) {
40         this.harga = harga;
41     }
42
43     public boolean isTersedia() {
44         return tersedia;
45     }
46
47     public void setTersedia(boolean tersedia) {
48         this.tersedia = tersedia;
49     }
50
51     public abstract String getInfo();
52
53     public String getInfo(boolean lengkap) {
54         if (lengkap) {
55             return "ID: " + id + ", Nama: " + nama + ", Harga: Rp " + harga +
56                 ", Status: " + (tersedia ? "Ready" : "Rented") + ", Info Tambahan: " + getInfo();
57         } else {
58             return nama + " (Rp " + harga + ")";
59         }
60     }
61
62     // Implementasi Interface
63     @Override
64     public void sewaUnit() {
65         if (tersedia) {
66             tersedia = false;
67             System.out.println("Unit berhasil disewa.");
68         } else {
69             System.out.println("Unit sedang tidak tersedia.");
70         }
71     }
72
73     @Override
74     public void kembalikanUnit() {
75         if (!tersedia) {
76             tersedia = true;
77             System.out.println("Unit telah dikembalikan.");
78         } else {
79             System.out.println("Unit sudah tersedia.");
80         }
81     }
82 }
83
84 // Excavator
85 class Excavator extends Unit {
86     private double kapasitas;
87
88     public Excavator(int id, String nama, double harga, boolean tersedia, double kapasitas) {
89         super(id, nama, harga, tersedia);
90         this.kapasitas = kapasitas;
91     }
92
93     public double getKapasitas() {
94         return kapasitas;
95     }
96
97     public void setKapasitas(double kapasitas) {
98         this.kapasitas = kapasitas;
99     }
100
101     @Override
102     public String getInfo() {
103         return "Kapasitas Bucket: " + kapasitas + " m3";
104     }
105 }

```

```

1  // Bulldozer
2  class Bulldozer extends Unit {
3      private String tipeBlade;
4
5      public Bulldozer(int id, String nama, double harga, boolean tersedia, String tipeBlade) {
6          super(id, nama, harga, tersedia);
7          this.tipeBlade = tipeBlade;
8      }
9
10     public String getTipeBlade() {
11         return tipeBlade;
12     }
13
14     public void setTipeBlade(String tipeBlade) {
15         this.tipeBlade = tipeBlade;
16     }
17
18     @Override
19     public String getInfo() {
20         return "Tipe Blade: " + tipeBlade;
21     }
22 }
23
24 // Final Class Main Program
25 public final class POSTTEST {
26     private static Scanner scanner = new Scanner(System.in);
27     private static List<Unit> daftarUnit = new ArrayList<>();
28     private static int unitId = 1;
29
30     // Static attribute dan method
31     public static final String namaPerusahaan = "CV. Bintang Jaya";
32
33     public static void tampilkanNamaPerusahaan() {
34         System.out.println("\n===== Penyewaan Alat Berat | " + namaPerusahaan + " | =====\n");
35     }
36
37     public static void main(String[] args) {
38         while (true) {
39             try {
40                 tampilkanNamaPerusahaan();
41                 System.out.println("===== Admin Menu =====");
42                 System.out.println("1. Kelola Unit");
43                 System.out.println("2. Keluar");
44                 System.out.print("Pilih menu: ");
45                 int pilihan = scanner.nextInt();
46                 scanner.nextLine();
47
48                 switch (pilihan) {
49                     case 1:
50                         MUnit();
51                         break;
52                     case 2:
53                         System.exit(0);
54                     default:
55                         System.out.println("Pilihan tidak valid!");
56                 }
57             } catch (InputMismatchException e) {
58                 System.out.println("Input tidak valid! Harap masukkan angka.");
59                 scanner.nextLine();
60             }
61         }
62     }
63
64     private static void MUnit() {
65         while (true) {
66             try {
67                 System.out.println("\n=== Pengelolaan Unit Sewa ===");
68                 System.out.println("1. Tambah Unit");
69                 System.out.println("2. Lihat Unit");
70                 System.out.println("3. Perbarui Unit");
71                 System.out.println("4. Hapus Unit");
72                 System.out.println("5. Sewa Unit");
73                 System.out.println("6. Kembalikan Unit");
74                 System.out.println("7. Kembali");
75                 System.out.print("Pilih: ");
76                 int pilihan = scanner.nextInt();
77                 scanner.nextLine();
78

```

```

1 private static void Main() {
2     while (true) {
3         try {
4             System.out.println("\n=== Pengelolaan Unit Sewa ===");
5             System.out.println("1. Tambah Unit");
6             System.out.println("2. Lihat Unit");
7             System.out.println("3. Perbarui Unit");
8             System.out.println("4. Hapus Unit");
9             System.out.println("5. Sewa Unit");
10            System.out.println("6. Kembalikan Unit");
11            System.out.println("7. Kembali");
12            System.out.print("Pilih: ");
13            int pilihan = scanner.nextInt();
14            scanner.nextLine();
15
16            if (pilihan == 1) {
17                System.out.println("Pilih tipe unit:");
18                System.out.println("1. Excavator");
19                System.out.println("2. Bulldozer");
20                int tipe = scanner.nextInt();
21                scanner.nextLine();
22
23                System.out.print("Nama Unit: ");
24                String nama = scanner.nextLine();
25                System.out.print("Harga /Day: ");
26                double harga = scanner.nextDouble();
27                scanner.nextLine();
28
29                if (tipe == 1) {
30                    System.out.print("Kapasitas Bucket (m3): ");
31                    double kapasitas = scanner.nextDouble();
32                    scanner.nextLine();
33                    daftarUnit.add(new Excavator(unitId++, nama, harga, true, kapasitas));
34                } else if (tipe == 2) {
35                    System.out.print("Tipe Blade: ");
36                    String blade = scanner.nextLine();
37                    daftarUnit.add(new Bulldozer(unitId++, nama, harga, true, blade));
38                } else {
39                    System.out.println("Tipe tidak valid!");
40                }
41            }
42
43            } else if (pilihan == 2) {
44                if (daftarUnit.isEmpty()) {
45                    System.out.println("Belum ada unit yang ditambahkan.");
46                } else {
47                    for (Unit unit : daftarUnit)
48                        System.out.println(unit.getInfo(true));
49                }
50            }
51
52            } else if (pilihan == 3) {
53                System.out.print("Masukkan ID unit yang akan diperbarui: ");
54                int id = scanner.nextInt();
55                scanner.nextLine();
56                boolean ditemukan = false;
57
58                for (Unit unit : daftarUnit) {
59                    if (unit.getId() == id) {
60                        System.out.print("Nama baru: ");
61                        unit.setNama(scanner.nextLine());
62                        System.out.print("Harga /Day: ");
63                        unit.setHarga(scanner.nextDouble());
64                        scanner.nextLine();
65
66                        if (unit instanceof Excavator) {
67                            System.out.print("Kapasitas Bucket baru (m3): ");
68                            ((Excavator) unit).setKapasitas(scanner.nextDouble());
69                            scanner.nextLine();
70                        } else if (unit instanceof Bulldozer) {
71                            System.out.print("Tipe Blade baru: ");
72                            ((Bulldozer) unit).setTipeBlade(scanner.nextLine());
73                        }
74
75                        System.out.println("Data unit berhasil diperbarui!");
76                        ditemukan = true;
77                        break;
78                    }
79                }
80
81                if (!ditemukan) System.out.println("Unit dengan ID tersebut tidak ditemukan.");
82            }
83
84            } else if (pilihan == 4) {
85                System.out.print("Masukkan ID unit yang akan dihapus: ");
86                int id = scanner.nextInt();
87                scanner.nextLine();
88
89                boolean dihapus = daftarUnit.removeIf(unit -> unit.getId() == id);
90                if (dihapus) {
91                    System.out.println("Unit berhasil dihapus.");
92                } else {
93                    System.out.println("Unit tidak ditemukan.");
94                }
95            }
96
97            } else if (pilihan == 5) {
98                System.out.print("Masukkan ID unit yang ingin disewa: ");
99                int id = scanner.nextInt();
100               scanner.nextLine();
101
102               boolean ditemukan = false;
103               for (Unit unit : daftarUnit) {
104                   if (unit.getId() == id) {
105                       unit.sewaUnit();
106                       ditemukan = true;
107                       break;
108                   }
109               }
110
111               if (!ditemukan) System.out.println("Unit tidak ditemukan.");
112            }
113
114            } else if (pilihan == 6) {
115                System.out.print("Masukkan ID unit yang ingin dikembalikan: ");
116                int id = scanner.nextInt();
117                scanner.nextLine();
118
119                boolean ditemukan = false;
120                for (Unit unit : daftarUnit) {
121                    if (unit.getId() == id) {
122                        unit.kembalikanUnit();
123                        ditemukan = true;
124                        break;
125                    }
126                }
127
128                if (!ditemukan) System.out.println("Unit tidak ditemukan.");
129            }
130
131            } else if (pilihan == 7) {
132                break;
133            }
134
135            } else {
136                System.out.println("Pilihan tidak valid");
137            }
138
139            } catch (InputMismatchException e) {
140                System.out.println("Input tidak valid! Harap masukkan angka.");
141                scanner.nextLine();
142            }
143        }
144    }
145
146    }
147
148    }
149
150    }
151
152    }
153
154    }
155
156    }
157
158    }
159
160    }
161
162    }
163
164    }
165
166    }
167
168    }
169
170    }
171
172    }
173
174    }
175
176    }
177
178    }
179
180    }
181
182    }
183
184    }
185
186    }
187
188    }
189
190    }
191
192    }
193
194    }
195
196    }
197
198    }
199
200    }
201
202    }
203
204    }
205
206    }
207
208    }
209
210    }
211
212    }
213
214    }
215
216    }
217
218    }
219
220    }
221
222    }
223
224    }
225
226    }
227
228    }
229
230    }
231
232    }
233
234    }
235
236    }
237
238    }
239
240    }
241
242    }
243
244    }
245
246    }
247
248    }
249
250    }
251
252    }
253
254    }
255
256    }
257
258    }
259
260    }
261
262    }
263
264    }
265
266    }
267
268    }
269
270    }
271
272    }
273
274    }
275
276    }
277
278    }
279
280    }
281
282    }
283
284    }
285
286    }
287
288    }
289
290    }
291
292    }
293
294    }
295
296    }
297
298    }
299
300    }
301
302    }
303
304    }
305
306    }
307
308    }
309
310    }
311
312    }
313
314    }
315
316    }
317
318    }
319
320    }
321
322    }
323
324    }
325
326    }
327
328    }
329
330    }
331
332    }
333
334    }
335
336    }
337
338    }
339
340    }
341
342    }
343
344    }
345
346    }
347
348    }
349
350    }
351
352    }
353
354    }
355
356    }
357
358    }
359
360    }
361
362    }
363
364    }
365
366    }
367
368    }
369
370    }
371
372    }
373
374    }
375
376    }
377
378    }
379
380    }
381
382    }
383
384    }
385
386    }
387
388    }
389
390    }
391
392    }
393
394    }
395
396    }
397
398    }
399
400    }
401
402    }
403
404    }
405
406    }
407
408    }
409
410    }
411
412    }
413
414    }
415
416    }
417
418    }
419
420    }
421
422    }
423
424    }
425
426    }
427
428    }
429
430    }
431
432    }
433
434    }
435
436    }
437
438    }
439
440    }
441
442    }
443
444    }
445
446    }
447
448    }
449
450    }
451
452    }
453
454    }
455
456    }
457
458    }
459
460    }
461
462    }
463
464    }
465
466    }
467
468    }
469
470    }
471
472    }
473
474    }
475
476    }
477
478    }
479
480    }
481
482    }
483
484    }
485
486    }
487
488    }
489
490    }
491
492    }
493
494    }
495
496    }
497
498    }
499
500    }
501
502    }
503
504    }
505
506    }
507
508    }
509
510    }
511
512    }
513
514    }
515
516    }
517
518    }
519
520    }
521
522    }
523
524    }
525
526    }
527
528    }
529
530    }
531
532    }
533
534    }
535
536    }
537
538    }
539
540    }
541
542    }
543
544    }
545
546    }
547
548    }
549
550    }
551
552    }
553
554    }
555
556    }
557
558    }
559
560    }
561
562    }
563
564    }
565
566    }
567
568    }
569
570    }
571
572    }
573
574    }
575
576    }
577
578    }
579
580    }
581
582    }
583
584    }
585
586    }
587
588    }
589
590    }
591
592    }
593
594    }
595
596    }
597
598    }
599
600    }
601
602    }
603
604    }
605
606    }
607
608    }
609
610    }
611
612    }
613
614    }
615
616    }
617
618    }
619
620    }
621
622    }
623
624    }
625
626    }
627
628    }
629
630    }
631
632    }
633
634    }
635
636    }
637
638    }
639
640    }
641
642    }
643
644    }
645
646    }
647
648    }
649
650    }
651
652    }
653
654    }
655
656    }
657
658    }
659
660    }
661
662    }
663
664    }
665
666    }
667
668    }
669
670    }
671
672    }
673
674    }
675
676    }
677
678    }
679
680    }
681
682    }
683
684    }
685
686    }
687
688    }
689
690    }
691
692    }
693
694    }
695
696    }
697
698    }
699
700    }
701
702    }
703
704    }
705
706    }
707
708    }
709
710    }
711
712    }
713
714    }
715
716    }
717
718    }
719
720    }
721
722    }
723
724    }
725
726    }
727
728    }
729
730    }
731
732    }
733
734    }
735
736    }
737
738    }
739
740    }
741
742    }
743
744    }
745
746    }
747
748    }
749
750    }
751
752    }
753
754    }
755
756    }
757
758    }
759
760    }
761
762    }
763
764    }
765
766    }
767
768    }
769
770    }
771
772    }
773
774    }
775
776    }
777
778    }
779
780    }
781
782    }
783
784    }
785
786    }
787
788    }
789
790    }
791
792    }
793
794    }
795
796    }
797
798    }
799
800    }
801
802    }
803
804    }
805
806    }
807
808    }
809
810    }
811
812    }
813
814    }
815
816    }
817
818    }
819
820    }
821
822    }
823
824    }
825
826    }
827
828    }
829
830    }
831
832    }
833
834    }
835
836    }
837
838    }
839
840    }
841
842    }
843
844    }
845
846    }
847
848    }
849
850    }
851
852    }
853
854    }
855
856    }
857
858    }
859
860    }
861
862    }
863
864    }
865
866    }
867
868    }
869
870    }
871
872    }
873
874    }
875
876    }
877
878    }
879
880    }
881
882    }
883
884    }
885
886    }
887
888    }
889
890    }
891
892    }
893
894    }
895
896    }
897
898    }
899
900    }
901
902    }
903
904    }
905
906    }
907
908    }
909
910    }
911
912    }
913
914    }
915
916    }
917
918    }
919
920    }
921
922    }
923
924    }
925
926    }
927
928    }
929
930    }
931
932    }
933
934    }
935
936    }
937
938    }
939
940    }
941
942    }
943
944    }
945
946    }
947
948    }
949
950    }
951
952    }
953
954    }
955
956    }
957
958    }
959
960    }
961
962    }
963
964    }
965
966    }
967
968    }
969
970    }
971
972    }
973
974    }
975
976    }
977
978    }
979
980    }
981
982    }
983
984    }
985
986    }
987
988    }
989
990    }
991
992    }
993
994    }
995
996    }
997
998    }
999
1000   }

```