

SS PROGRAM

Main.java

```
1 import java.util.ArrayList;
2 import java.util.Scanner;
3
4 public class Main {
5     static ArrayList<Sepeda> daftarSepeda = new ArrayList<>();
6     static ArrayList<TransaksiPenyewaan> daftarTransaksi = new ArrayList<>();
7     static Scanner input = new Scanner(System.in);
8     static int idSepedaCounter = 1;
9     static int idTransaksiCounter = 1;
10
11     public static void main(String[] args) {
12         while (true) {
13             System.out.println("\n=== SISTEM PENYEWAAN SEPEDA ===");
14             System.out.println("1. Tambah Sepeda");
15             System.out.println("2. Lihat Semua Sepeda");
16             System.out.println("3. Ubah Sepeda");
17             System.out.println("4. Hapus Sepeda");
18             System.out.println("5. Tambah Transaksi Penyewaan");
19             System.out.println("6. Lihat Semua Transaksi");
20             System.out.println("7. Ubah Transaksi");
21             System.out.println("8. Hapus Transaksi");
22             System.out.println("9. Keluar");
23             System.out.print("Pilih menu: ");
24             int pilihan = input.nextInt();
25             input.nextLine();
26
27             switch (pilihan) {
28                 case 1: {
29                     System.out.println("Pilih tipe sepeda:");
30                     System.out.println("1. Sepeda Gunung");
31                     System.out.println("2. Sepeda Lipat");
32                     System.out.print("Pilihan: ");
33                     int tipe = input.nextInt();
34                     input.nextLine();
35                     System.out.print("Masukkan jenis sepeda: ");
36                     String jenis = input.nextLine();
37
38                     if (tipe == 1) {
39                         daftarSepeda.add(new SepedaGunung(idSepedaCounter++, jenis));
40                     } else if (tipe == 2) {
41                         daftarSepeda.add(new SepedaLipat(idSepedaCounter++, jenis));
42                     } else {
43                         System.out.println("Inputan Tidak Valid.");
44                         break;
45                     }
46                     System.out.println("Sepeda berhasil ditambahkan!");
47                     break;
48                 }
49
50                 case 2: {
51                     Sepeda.LihatSemuaSepeda(daftarSepeda);
52                     break;
53                 }
54
55                 case 3: {
56                     System.out.print("Masukkan ID sepeda yang ingin diperbarui: ");
57                     int id = input.nextInt();
58                     input.nextLine();
59                     System.out.print("Masukkan jenis sepeda baru: ");
60                     String jenis = input.nextLine();
61                     System.out.print("Apakah tersedia? (true/false): ");
62                     boolean tersedia = input.nextBoolean();
63                     Sepeda.PerbaruiSepeda(daftarSepeda, id, jenis, tersedia);
64                     break;
65                 }
66
67                 case 4: {
68                     System.out.print("Masukkan ID sepeda yang ingin dihapus: ");
69                     int id = input.nextInt();
70                     Sepeda.HapusSepeda(daftarSepeda, id);
71                     break;
72                 }
73
74                 case 5: {
75                     System.out.print("Masukkan nama penyewa: ");
76                     String penyewa = input.nextLine();
77                     System.out.print("Masukkan ID sepeda yang ingin disewa: ");
78                     int idSepeda = input.nextInt();
79                     TransaksiPenyewaan.TambahTransaksi(daftarTransaksi, daftarSepeda, idTransaksiCounter++, idSepeda, penyewa);
80                     break;
81                 }
82
83                 case 6: {
84                     TransaksiPenyewaan.LihatSemuaTransaksi(daftarTransaksi);
85                     break;
86                 }
87
88                 case 7: {
89                     System.out.print("Masukkan ID transaksi yang ingin diperbarui: ");
90                     int id = input.nextInt();
91                     TransaksiPenyewaan.PerbaruiTransaksi(daftarTransaksi, daftarSepeda, id);
92                     break;
93                 }
94
95                 case 8: {
96                     System.out.print("Masukkan ID transaksi yang ingin dihapus: ");
97                     int id = input.nextInt();
98                     TransaksiPenyewaan.HapusTransaksi(daftarTransaksi, id);
99                     break;
100                }
101
102                case 9: {
103                    System.out.println("Program selesai.");
104                    System.exit(0);
105                    break;
106                }
107
108                default: {
109                    System.out.println("Silakan coba lagi.");
110                    break;
111                }
112            }
113        }
114    }
115 }
116
```

Sepeda.java

```

1  import java.util.ArrayList;
2
3  public abstract class Sepeda {
4      private int id;
5      private String jenis;
6      private boolean tersedia;
7
8      public Sepeda(int id, String jenis, boolean tersedia) {
9          this.id = id;
10         this.jenis = jenis;
11         this.tersedia = tersedia;
12     }
13
14     public int GetId() {
15         return id;
16     }
17
18     public String GetJenis() {
19         return jenis;
20     }
21
22     public boolean IsTersedia() {
23         return tersedia;
24     }
25
26     public void SetId(int id) {
27         this.id = id;
28     }
29
30     public void SetJenis(String jenis) {
31         this.jenis = jenis;
32     }
33
34     public void SetTersedia(boolean tersedia) {
35         this.tersedia = tersedia;
36     }
37
38     public void TampilkanInfo() {
39         System.out.println("ID: " + id + " | Jenis: " + jenis + " | Tersedia: " + (tersedia ? "Ya" : "Tidak") + " | Tipe: " + GetType());
40     }
41
42     // Abstract method yang wajib di-override oleh turunan
43     public abstract String GetType();
44
45     public static void LihatSemuaSepeda(ArrayList<Sepeda> daftarSepeda) {
46         if (daftarSepeda.isEmpty()) {
47             System.out.println("Tidak ada sepeda tersedia.");
48             return;
49         }
50         for (Sepeda s : daftarSepeda) {
51             s.TampilkanInfo();
52         }
53     }
54
55     public static void PerbaruiSepeda(ArrayList<Sepeda> daftarSepeda, int id, String jenisBaru, boolean tersedia) {
56         for (Sepeda s : daftarSepeda) {
57             if (s.GetId() == id) {
58                 s.SetJenis(jenisBaru);
59                 s.SetTersedia(tersedia);
60                 System.out.println("Sepeda berhasil diperbarui!");
61                 return;
62             }
63         }
64         System.out.println("Sepeda tidak ditemukan.");
65     }
66
67     public static void HapusSepeda(ArrayList<Sepeda> daftarSepeda, int id) {
68         daftarSepeda.removeIf(s -> s.GetId() == id);
69         System.out.println("Sepeda berhasil dihapus!");
70     }
71 }
72

```

SepedaGunung.java

```
1 public class SepedaGunung extends Sepeda {
2     public SepedaGunung(int id, String jenis) {
3         super(id, jenis, true);
4     }
5
6     @Override
7     public void TampilkanInfo() {
8         System.out.println("== Sepeda Gunung ==");
9         super.TampilkanInfo();
10    }
11
12    @Override
13    public String GetTipe() {
14        return "Sepeda Gunung";
15    }
16 }
17
```

SepedaLipat.java

```
1 public final class SepedaLipat extends Sepeda {
2     public SepedaLipat(int id, String jenis) {
3         super(id, jenis, true);
4     }
5
6     @Override
7     public void TampilkanInfo() {
8         System.out.println("== Sepeda Lipat ==");
9         super.TampilkanInfo();
10    }
11
12    @Override
13    public String GetTipe() {
14        return "Sepeda Lipat";
15    }
16 }
17
```

TransaksiPenyewaan.java

```
1 import java.util.ArrayList;
2
3 public class TransaksiPenyewaan {
4     private final int idTransaksi;
5     private int idSepeda;
6     private String penyewa;
7     private boolean selesai;
8
9     public TransaksiPenyewaan(int idTransaksi, int idSepeda, String penyewa, boolean selesai) {
10         this.idTransaksi = idTransaksi;
11         this.idSepeda = idSepeda;
12         this.penyewa = penyewa;
13         this.selesai = selesai;
14     }
15
16     public int GetIdTransaksi() { return idTransaksi; }
17     public int GetIdSepeda() { return idSepeda; }
18     public String GetPenyewa() { return penyewa; }
19     public boolean isSelesai() { return selesai; }
20
21     public void SetIdSepeda(int idSepeda) { this.idSepeda = idSepeda; }
22     public void SetPenyewa(String penyewa) { this.penyewa = penyewa; }
23     public void SetSelesai(boolean selesai) { this.selesai = selesai; }
24
25     public final void TampilkanInfo() {
26         System.out.println("ID Transaksi: " + idTransaksi + " | ID Sepeda: " + idSepeda + " | Penyewa: " + penyewa + " | Selesai: " + (selesai ? "Ya" : "Tidak"));
27     }
28
29     public static void TambahTransaksi(ArrayList<TransaksiPenyewaan> daftarTransaksi, ArrayList<Sepeda> daftarSepeda, int idTransaksi, int idSepeda, String penyewa, boolean langsungSelesai) {
30         for (Sepeda s : daftarSepeda) {
31             if (s.getId() == idSepeda && s.isTersedia()) {
32                 daftarTransaksi.add(new TransaksiPenyewaan(idTransaksi, idSepeda, penyewa, langsungSelesai));
33                 s.setTersedia(l langsungSelesai);
34                 System.out.println("Transaksi (langsung selesai: " + langsungSelesai + ") berhasil dibuat!");
35                 return;
36             }
37         }
38         System.out.println("Sepeda tidak tersedia atau tidak ditemukan.");
39     }
40
41     public static void TambahTransaksi(ArrayList<TransaksiPenyewaan> daftarTransaksi, ArrayList<Sepeda> daftarSepeda, int idTransaksi, int idSepeda, String penyewa) {
42         TambahTransaksi(daftarTransaksi, daftarSepeda, idTransaksi, idSepeda, penyewa, false);
43     }
44
45     public static void LihatSemuaTransaksi(ArrayList<TransaksiPenyewaan> daftarTransaksi) {
46         if (daftarTransaksi.isEmpty()) {
47             System.out.println("Tidak ada transaksi penyewaan.");
48             return;
49         }
50         for (TransaksiPenyewaan t : daftarTransaksi) {
51             t.TampilkanInfo();
52         }
53     }
54
55     public static void PerbaruiTransaksi(ArrayList<TransaksiPenyewaan> daftarTransaksi, ArrayList<Sepeda> daftarSepeda, int id) {
56         for (TransaksiPenyewaan t : daftarTransaksi) {
57             if (t.getIdTransaksi() == id) {
58                 t.setSelesai(true);
59                 for (Sepeda s : daftarSepeda) {
60                     if (s.getId() == t.getIdSepeda()) {
61                         s.setTersedia(true);
62                         break;
63                     }
64                 }
65                 System.out.println("Transaksi berhasil diperbarui!");
66                 return;
67             }
68         }
69         System.out.println("Transaksi tidak ditemukan.");
70     }
71
72     public static void HapusTransaksi(ArrayList<TransaksiPenyewaan> daftarTransaksi, int id) {
73         daftarTransaksi.removeIf(t -> t.getIdTransaksi() == id);
74         System.out.println("Transaksi berhasil dihapus!");
75     }
76 }
77
```

