

Nama : Siti Fauziah Wulandari

NIM : 2309106038

Kelas : A2

**Praktikum PBO**

**POSTTEST 5**

## Implementasi Keyword Final

1. Final Class: Di Admin.java pada baris 4

```
ApotekSystem > src > models > Admin.java > Admin
Edit file using CodeParrot (ctrl+h)
1 package models;
2
3 // Final class
4 public final class Admin extends User {
5     private String departemen;
6
7     public Admin(int idAdmin, String nama, String noTelp, String departemen) {
8         super(idAdmin, nama, tipeUser:"Admin", noTelp);
9         this.departemen = departemen;
10    }
11
12    public String getDepartemen() {
13        return departemen;
14    }
15
16    public void setDepartemen(String departemen) {
17        this.departemen = departemen;
18    }
19
20    // Implementasi method abstract dari parent class
21    @Override
22    public void tampilkanInfo() {
23        System.out.println("ID Admin: " + id);
24        System.out.println("Nama: " + nama);
25        System.out.println("Tipe: " + tipeUser);
26        System.out.println("No Telp: " + noTelp);
27        System.out.println("Departemen: " + departemen);
28        System.out.println(x:"-----");
29    }
30 }
```

```
Edit file using CodeParrot (ctrl+n)
1 package models;
2
3 // Abstract class --> parent class
4 public abstract class User {
5     protected int id;
6     protected String nama;
7     // Final attribute
8     protected final String tipeUser;
9     protected String noTelp;
10
11     public User(int id, String nama, String tipeUser, String noTelp) {
12         this.id = id;
13         this.nama = nama;
14         this.tipeUser = tipeUser;
15         this.noTelp = noTelp;
16     }
17
18     // Getter
19     public int getId() {
20         return id;
21     }
22
23     public String getNama() {
24         return nama;
25     }
26
27     public String getTypeUser() {
28         return tipeUser;
29     }
30
31     public String getNoTelp() {
32         return noTelp;
33     }
34
35     // Setter
36     public void setName(String nama) {
37         this.nama = nama;
38     }
39
40     public void setNoTelp(String noTelp) {
41         this.noTelp = noTelp;
42     }
43
44     public abstract void tampilkanInfo();
45
46     public final String getIdentitas() {
47         return "ID: " + id + ", Nama: " + nama + ", Tipe: " + tipeUser;
48     }
49 }
```

2. method final di PelangganService.java pada baris 68-71:

```
1 package services;
2
3 import models.Obat;
4 import models.Pelanggan;
5 import models.ResepDokter;
6 import models.Transaksi;
7
8 import java.time.LocalDate;
9 import java.util.ArrayList;
10 import java.util.Scanner;
11
12 public class PelangganService {
13     // Encapsulation
14     private ArrayList<Obat> daftarObat = new ArrayList<>();
15     private ArrayList<Transaksi> daftarTransaksi = new ArrayList<>();
16     private ArrayList<ResepDokter> daftarResep = new ArrayList<>();
17     private Scanner scanner = new Scanner(System.in);
18     private int idPelanggan = 1;
19     private int idResep = 1;
20
21     public PelangganService(ArrayList<Obat> daftarObat, ArrayList<Transaksi> daftarTransaksi, ArrayList<ResepDokter> daftarResep, Scanner scanner) {
22         this.daftarObat = daftarObat;
23         this.daftarTransaksi = daftarTransaksi;
24         this.daftarResep = daftarResep;
25         this.scanner = scanner;
26     }
27
28     public void menuPelanggan() {
29         while (true) {
30             System.out.println(<: "\n=== MENU PELANGGAN ===");
31             System.out.println(<: "1. Lihat Obat");
32             System.out.println(<: "2. Beli Obat");
33             System.out.println(<: "3. Lihat Riwayat Transaksi");
34             System.out.println(<: "4. Logout");
35             System.out.print(<: "Pilih: ");
36             int pilihan = scanner.nextInt();
37             scanner.nextLine();
38
39             switch (pilihan) {
40                 case 1: lihatObat(); break;
41                 case 2: beliObat(); break;
42                 case 3: lihatRiwayatTransaksi(); break;
43                 case 4: return;
44                 default: System.out.println(<: "Pilihan tidak valid.");
45             }
46         }
47     }
48 }
```

```

49     private void lihatObat() {
50         System.out.println(x:"\n=== DAFTAR OBAT ===");
51         for (Obat obat : daftarObat) {
52             obat.tampilkanObat();
53         }
54     }
55
56     private int inputUsia() {
57         while (true) {
58             try {
59                 System.out.print(s:"Masukkan Usia: ");
60                 return Integer.parseInt(scanner.nextLine());
61             } catch (NumberFormatException e) {
62                 System.out.println(x:"Usia harus berupa angka. Silakan coba lagi.");
63             }
64         }
65     }
66
67     // Method final
68     public final void tambahPelanggan(Pelanggan pelanggan) {
69         System.out.println(x:"Pelanggan baru berhasil ditambahkan!");
70         pelanggan.tampilkanInfo();
71     }
72
73     // protected
74     protected void beliObat() {
75         System.out.println(x:"=== BELI OBAT ===");
76         lihatObat();
77         System.out.print(s:"Masukkan ID Obat yang ingin dibeli: ");
78         int idObat = scanner.nextInt();
79         scanner.nextLine();
80
81         Obat obatDibeli = null;
82         for (Obat obat : daftarObat) {
83             if (obat.getIdObat() == idObat) {
84                 obatDibeli = obat;
85                 break;
86             }
87         }
88
89         if (obatDibeli == null) {
90             System.out.println(x:"Obat tidak ditemukan!");
91             return;
92         }
93     }

```

```

94     System.out.print(s:"Masukkan Nama Anda: ");
95     String namaPelanggan = scanner.nextLine();
96     int usia = inputUsia();
97     System.out.print(s:"Masukkan Alamat: ");
98     String alamat = scanner.nextLine();
99     System.out.print(s:"Masukkan Nomor Telepon: ");
100
101     String noTelp = scanner.nextLine();
102
103     System.out.print(s:"Jumlah yang ingin dibeli: ");
104     int jumlah = scanner.nextInt();
105     scanner.nextLine();
106
107     if (jumlah > obatDibeli.getStok()) {
108         System.out.println(x:"Stok tidak mencukupi.");
109         return;
110     }
111
112     boolean butuhResep = obatDibeli.isResepDokter();
113     if (butuhResep) {
114         System.out.println(x:"\n=== Obat ini memerlukan resep dokter. ===");
115         System.out.print(s:"Nama Dokter: ");
116         String namaDokter = scanner.nextLine();
117         System.out.print(s:"Nama Klinik: ");
118         String namaKlinik = scanner.nextLine();
119         LocalDate tanggalResep = LocalDate.now();
120
121         ResepDokter resep = new ResepDokter(idResep++, idPelanggan, idObat, namaDokter, namaKlinik, tanggalResep, statusVerifikasi:false);
122         daftarResep.add(resep);
123         System.out.println(x:"Resep berhasil diunggah, menunggu verifikasi Admin.");
124     } else {
125         System.out.println(x:"\nObat ini **tidak** memerlukan resep dokter. Langsung ke pembayaran.");
126     }
127
128     System.out.println(x:"\n=== PILIH METODE PEMBAYARAN ===");
129     System.out.println(x:"1. Ambil di Apotek");
130     System.out.println(x:"2. Transfer");
131     System.out.print(s:"Pilih: ");
132     int metode = scanner.nextInt();
133     scanner.nextLine();
134
135     String metodePembayaran = metode == 1 ? "Ambil di Apotek" : "Transfer";
136     double totalHarga = jumlah * obatDibeli.getHarga();
137     String statusTransaksi = butuhResep ? "Menunggu Verifikasi Resep" : "Selesai";
138     String statusPengambilan = metode == 1 ? "Belum Diambil" : "Selesai";

```

```

140     Pelanggan pelanggan = new Pelanggan(idPelanggan, namaPelanggan, usia, alamat, noTelp);
141
142     System.out.println(x:"\n=== DETAIL PELANGGAN ===");
143     pelanggan.tampilkanInfo();
144
145     System.out.println("Identitas: " + pelanggan.getIdentitas());
146
147     Transaksi transaksi = new Transaksi(
148         daftarTransaksi.size() + 1, idPelanggan, idObat, jumlah, totalHarga, LocalDate.now(),
149         statusTransaksi, metodePembayaran, statusPengambilan, alamat, namaPelanggan, usia, noTelp
150     );
151
152     daftarTransaksi.add(transaksi);
153     obatDibeli.setStok(obatDibeli.getStok() - jumlah);
154     System.out.println(x:"\nTransaksi berhasil dibuat!");
155
156     idPelanggan++;
157 }
158
159 // Getter
160 public ArrayList<Transaksi> getDaftarTransaksi() {
161     return daftarTransaksi;
162 }
163
164 // Setter
165 public void setDaftarTransaksi(ArrayList<Transaksi> daftarTransaksi) {
166     this.daftarTransaksi = daftarTransaksi;
167 }
168
169 private void lihatRiwayatTransaksi() {
170     System.out.println(x:"\n=== RIWAYAT TRANSAKSI ===");
171     for (Transaksi transaksi : daftarTransaksi) {
172         transaksi.tampilkanTransaksi();
173     }
174 }
175 }

```

## Implementasi Abstract Class

1. Abstract Class: Di User.java pada baris 4

```
1 package models;
2
3 // Abstract class --> parent class
4 public abstract class User {
5     protected int id;
6     protected String nama;
7     // Final attribute
8     protected final String tipeUser;
9     protected String noTelp;
10
11     public User(int id, String nama, String tipeUser, String noTelp) {
12         this.id = id;
13         this.nama = nama;
14         this.tipeUser = tipeUser;
15         this.noTelp = noTelp;
16     }
17
18     // Getter
19     public int getId() {
20         return id;
21     }
22
23     public String getNama() {
24         return nama;
25     }
26
27     public String getTipeUser() {
28         return tipeUser;
29     }
30
31     public String getNoTelp() {
32         return noTelp;
33     }
34
35     // Setter
36     public void setName(String nama) {
37         this.nama = nama;
38     }
39
40     public void setNoTelp(String noTelp) {
41         this.noTelp = noTelp;
42     }
43
44     public abstract void tampilkanInfo();
45
46     public final String getIdentitas() {
47         return "ID: " + id + ", Nama: " + nama + ", Tipe: " + tipeUser;
48     }
49 }
50
```

## 2. Abstract Method: Di User.java pada baris 44

```
1  package models;
2
3  // Abstract class --> parent class
4  public abstract class User {
5      protected int id;
6      protected String nama;
7      // Final attribute
8      protected final String tipeUser;
9      protected String noTelp;
10
11     public User(int id, String nama, String tipeUser, String noTelp) {
12         this.id = id;
13         this.nama = nama;
14         this.tipeUser = tipeUser;
15         this.noTelp = noTelp;
16     }
17
18     // Getter
19     public int getId() {
20         return id;
21     }
22
23     public String getNama() {
24         return nama;
25     }
26
27     public String getTipeUser() {
28         return tipeUser;
29     }
30
31     public String getNoTelp() {
32         return noTelp;
33     }
34
35     // Setter
36     public void setNama(String nama) {
37         this.nama = nama;
38     }
39
40     public void setNoTelp(String noTelp) {
41         this.noTelp = noTelp;
42     }
43
44     public abstract void tampilkanInfo();
45
46     public final String getIdentitas() {
47         return "ID: " + id + ", Nama: " + nama + ", Tipe: " + tipeUser;
48     }
49 }
50
```

Method abstract ini diimplementasikan di subclass:

1. Di Admin.java pada baris 21-27
2. Di Pelanggan.java pada baris 32-43

```
1 package models;
2
3 // Final class
4 public final class Admin extends User {
5     private String departemen;
6
7     public Admin(int idAdmin, String nama, String noTelp, String departemen) {
8         super(idAdmin, nama, "Admin", noTelp);
9         this.departemen = departemen;
10    }
11
12    public String getDepartemen() {
13        return departemen;
14    }
15
16    public void setDepartemen(String departemen) {
17        this.departemen = departemen;
18    }
19
20    @Override
21    public void tampilkanInfo() {
22        System.out.println("ID Admin: " + id);
23        System.out.println("Nama: " + nama);
24        System.out.println("Tipe: " + tipeUser);
25        System.out.println("No Telp: " + noTelp);
26        System.out.println("Departemen: " + departemen);
27        System.out.println("-----");
28    }
29 }
```

```
1 package models;
2
3 public class Pelanggan extends User {
4     private int usia;
5     private String alamat;
6
7     public Pelanggan(int idPelanggan, String nama, int usia, String alamat, String noTelp) {
8         super(idPelanggan, nama, "Pelanggan", noTelp);
9         this.usia = usia;
10        this.alamat = alamat;
11    }
12
13    // Getter
14    public int getUsia() {
15        return usia;
16    }
17
18    public String getAlamat() {
19        return alamat;
20    }
21
22    // Setter
23    public void setUsia(int usia) {
24        this.usia = usia;
25    }
26
27    public void setAlamat(String alamat) {
28        this.alamat = alamat;
29    }
30
31    @Override
32    public void tampilkanInfo() {
33        System.out.println("ID Pelanggan: " + id);
34        System.out.println("Nama: " + nama);
35        System.out.println("Tipe: " + tipeUser);
36        System.out.println("Usia: " + usia);
37        System.out.println("Alamat: " + alamat);
38        System.out.println("No Telp: " + noTelp);
39        System.out.println("-----");
40    }
41
42    public void tampilkanPelanggan() {
43        tampilkanInfo();
44    }
45 }
```