

Nama: Cellia Auzia Nugraha

NIM: 2309106005

Kelas: A1'23

SS PROGRAM

POSTTEST 6

Interface (Memiliki 2 method didalamnya) dan penggunaan static

1. CetakInfo.java

```
1 package model;  
2  
3 public interface CetakInfo {  
4     void tampilkanInfo();  
5     void cetakLabel();  
6 }  
7
```

2. Bunga.java

```
1 package model;  
2  
3 public class Bunga implements CetakInfo {  
4     private int kode;  
5     private String nama;  
6     private int stock;  
7     private double harga;  
8  
9     public Bunga(int kode, String nama, int stock, double harga){  
10        this.kode = kode;  
11        this.nama = nama;  
12        this.stock = stock;  
13        this.harga = harga;  
14    }  
15  
16    public int getKode(){  
17        return kode;  
18    }  
19  
20    public void setKode(int kode){  
21        this.kode = kode;  
22    }  
23  
24    public String getNama(){  
25        return nama;  
26    }  
27  
28    public void setNama(String nama){  
29        this.nama = nama;  
30    }  
31  
32    public int getStock(){  
33        return stock;  
34    }  
35}
```

```

36     public void setStock(int stock){
37         this.stock = stock;
38     }
39
40     public double getHarga(){
41         return harga;
42     }
43
44     public void setHarga(double harga){
45         this.harga = harga;
46     }
47
48     @Override
49     public void tampilkanInfo(){
50         System.out.println("Kode: " + kode + ", Nama: " + nama + ", Stok: " + stock + ", Harga: Rp" + harga);
51     }
52
53     @Override
54     public void cetakLabel(){
55         System.out.println("=== Label ===\n" + nama + " (Rp" + harga + ")\nStok: " + stock);
56     }
57
58     @Override
59     public String toString(){
60         return nama;
61     }
62 }

```

3. TokoBunga.java

```

1  import java.util.ArrayList;
2  import java.util.List;
3  import java.util.Scanner;
4  import model.Pengguna;
5  import model.Transaksi;
6  import model.Bunga;
7  import model.CetakInfo;
8  import laporan.LaporanPenjualan;
9  import laporan.LaporanStok;
10
11 public class TokoBunga {
12     static List<Bunga> daftarBunga = new ArrayList<>();
13     static List<Transaksi> daftarTransaksi = new ArrayList<>();
14     static List<Pengguna> penggunaList = new ArrayList<>();
15     static Scanner scanner = new Scanner(System.in);
16     static Pengguna penggunaAktif;
17
18     public static void main(String[] args) throws Exception {
19         penggunaList.add(new Pengguna(username:"admin", pass:"admin",role:"admin"));
20         penggunaList.add(new Pengguna(username:"kasir", pass:"kasir", role:"kasir"));
21
22         boolean isRunning = true;
23         while(isRunning){
24             if(penggunaAktif == null){
25                 if(!login()){
26                     System.out.println(x:"Login Gagal!");
27                     continue;
28                 }
29             }
30             int pilih;
31             if (penggunaAktif != null) {
32                 do {
33                     System.out.println(x:"\n=====Sistem Penjualan Dan Stock Bunga=====");
34                     System.out.println(x:"1. Tambah Bunga (Admin)");
35                     System.out.println(x:"2. Perbarui Bunga (Admin)");

```

```

36         System.out.println(x:"3. Hapus Bunga (Admin)");
37         System.out.println(x:"4. Buat Transaksi Penjualan (Kasir)");
38         System.out.println(x:"5. Lihat Laporan Penjualan (Admin)");
39         System.out.println(x:"6. Lihat Laporan Stok Bunga (Admin)");
40         System.out.println(x:"7. Tampilkan Semua Bunga");
41         System.out.println(x:"8. Logout");
42         System.out.println(x:"9. Keluar");
43         System.out.print(s:"Pilihan: ");
44         pilih = scanner.nextInt();
45         scanner.nextLine();
46
47         switch(pilih){
48             case 1:
49                 if (penggunaAktif.getRole().equals(anObject:"admin")) tambahBunga();
50                 else System.out.println(x:"Akses ditolak!");
51                 break;
52             case 2:
53                 if (penggunaAktif.getRole().equals(anObject:"admin")) perbaruiBunga();
54                 else System.out.println(x:"Akses ditolak!");
55                 break;
56             case 3:
57                 if (penggunaAktif.getRole().equals(anObject:"admin")) hapusBunga();
58                 else System.out.println(x:"Akses ditolak!");
59                 break;
60             case 4:
61                 if (penggunaAktif.getRole().equals(anObject:"kasir")) buatTransaksi();
62                 else System.out.println(x:"Akses ditolak!");
63                 break;
64             case 5:
65                 if (penggunaAktif.getRole().equals(anObject:"admin")) lihatLaporanPenjualan();
66                 else System.out.println(x:"Akses ditolak!");
67                 break;
68             case 6:
69                 if (penggunaAktif.getRole().equals(anObject:"admin")) lihatLaporanStok();

```

```

70                 else System.out.println(x:"Akses ditolak!");
71                 break;
72             case 7:
73                 tampilkanSemuaBunga();
74                 break;
75             case 8:
76                 System.out.println(x:"Logout berhasil. Silakan login kembali.");
77                 penggunaAktif = null;
78                 break;
79             case 9:
80                 System.out.println(x:"Keluar dari sistem.");
81                 isRunning = false;
82                 break;
83         }
84     } while (isRunning && penggunaAktif != null);
85 }
86 }
87 }
88
89 public static boolean login() {
90     try {
91         System.out.print(s:"Masukkan Username: ");
92         String username = scanner.nextLine();
93         System.out.print(s:"Masukkan Password: ");
94         String pass = scanner.nextLine();
95
96         for (Pengguna pengguna : penggunaList) {
97             if (pengguna.getUsername().equals(username) && pengguna.getPass().equals(pass)) {
98                 penggunaAktif = pengguna;
99                 System.out.println("Login berhasil sebagai " + pengguna.getRole());
100                 return true;
101             }
102         }
103         System.out.println(x:"Username atau password salah!");

```

```

104     } catch (Exception e) {
105         System.out.println("Terjadi kesalahan saat login: " + e.getMessage());
106     }
107     return false;
108 }
109
110 public static void tambahBunga() {
111     try {
112         System.out.print(s:"Masukkan Kode Bunga: ");
113         int kode = Integer.parseInt(scanner.nextLine());
114         System.out.print(s:"Masukkan Nama Bunga: ");
115         String nama = scanner.nextLine();
116         System.out.print(s:"Masukkan Stok: ");
117         int stock = Integer.parseInt(scanner.nextLine());
118         System.out.print(s:"Masukkan Harga: ");
119         double harga = Double.parseDouble(scanner.nextLine());
120
121         daftarBunga.add(new Bunga(kode, nama, stock, harga));
122         System.out.println(x:"Bunga berhasil ditambahkan!");
123     } catch (NumberFormatException e) {
124         System.out.println(x:"Input tidak valid! Pastikan angka dimasukkan dengan benar.");
125     } catch (Exception e) {
126         System.out.println("Terjadi kesalahan saat menambahkan bunga: " + e.getMessage());
127     }
128 }
129
130 public static void perbaruiBunga() {
131     try {
132         System.out.print(s:"Masukkan kode Bunga yang akan diperbarui: ");
133         int kode = Integer.parseInt(scanner.nextLine());
134
135         for (Bunga bunga : daftarBunga) {
136             if (bunga.getKode() == kode) {
137                 System.out.print(s:"Masukkan Nama Baru: ");
138                 bunga.setNama(scanner.nextLine());

```

```

139                 System.out.print(s:"Masukkan Stok Baru: ");
140                 bunga.setStock(Integer.parseInt(scanner.nextLine()));
141                 System.out.print(s:"Masukkan Harga Baru: ");
142                 bunga.setHarga(Double.parseDouble(scanner.nextLine()));
143                 System.out.println(x:"Data bunga berhasil diperbarui!");
144                 return;
145             }
146         }
147         System.out.println(x:"Bunga tidak ditemukan!");
148     } catch (Exception e) {
149         System.out.println("Terjadi kesalahan saat memperbarui bunga: " + e.getMessage());
150     }
151 }
152
153 public static void hapusBunga() {
154     try {
155         System.out.print(s:"Masukkan Kode Bunga yang akan dihapus: ");
156         int kode = Integer.parseInt(scanner.nextLine());
157         daftarBunga.removeIf(bunga -> bunga.getKode() == kode);
158         System.out.println(x:"Bunga berhasil dihapus!");
159     } catch (Exception e) {
160         System.out.println("Terjadi kesalahan saat menghapus bunga: " + e.getMessage());
161     }
162 }
163
164 public static void buatTransaksi() {
165     try {
166         System.out.print(s:"Masukkan Kode Bunga yang Dibeli: ");
167         int kode = Integer.parseInt(scanner.nextLine());
168         System.out.print(s:"Masukkan Jumlah yang Dibeli: ");
169         int jumlah = Integer.parseInt(scanner.nextLine());
170         System.out.print(s:"Masukkan Nama Pelanggan: ");
171         String pelanggan = scanner.nextLine();

```

```

173         for (Bunga bunga : daftarBunga) {
174             if (bunga.getKode() == kode) {
175                 if (bunga.getStock() >= jumlah) {
176                     bunga.setStock(bunga.getStock() - jumlah);
177                     Transaksi transaksi = new Transaksi(daftarTransaksi.size() + 1, bunga, jumlah, pelanggan, totalHarga:0);
178                     transaksi.setJumlah(jumlah, updateHarga:true);
179                     daftarTransaksi.add(transaksi);
180                     System.out.println("Transaksi berhasil! Total Harga: Rp" + transaksi.getTotalHarga());
181                 } else {
182                     System.out.println(x:"Stok tidak mencukupi!");
183                 }
184                 return;
185             }
186         }
187         System.out.println(x:"Bunga tidak ditemukan!");
188     } catch (Exception e) {
189         System.out.println("Terjadi kesalahan saat membuat transaksi: " + e.getMessage());
190     }
191 }
192
193 public static void lihatLaporanPenjualan(){
194     try {
195         LaporanPenjualan laporan = new LaporanPenjualan(periode:"Maret 2025", daftarTransaksi);
196         laporan.tampilkanLaporan();
197     } catch (Exception e) {
198         System.out.println("Terjadi kesalahan saat menampilkan laporan penjualan: " + e.getMessage());
199     }
200 }
201
202 public static void lihatLaporanStok(){
203     try {
204         LaporanStok laporan = new LaporanStok(periode:"Maret 2025", daftarBunga);
205         laporan.tampilkanLaporan();
206     } catch (Exception e) {

```

```

207         System.out.println("Terjadi kesalahan saat menampilkan laporan stok: " + e.getMessage());
208     }
209 }
210
211 public static void tampilkanSemuaBunga() {
212     try {
213         System.out.println(x:"\n=== Daftar Bunga ===");
214         for (CetakInfo bunga : daftarBunga) {
215             bunga.tampilkanInfo();
216         }
217     } catch (Exception e) {
218         System.out.println("Terjadi kesalahan saat menampilkan bunga: " + e.getMessage());
219     }
220 }
221 }

```