

Nama: Cellia Auzia Nugraha

NIM: 2309106005

Kelas: A1'23

SS PROGRAM

Polymorphism

1. Laporan.java

```
1  package laporan;
2
3  public abstract class Laporan {
4      private String periode;
5
6      public Laporan(String periode){
7          this.periode = periode;
8      }
9
10     public String getPeriode(){
11         return periode;
12     }
13
14     public void setPeriode(String periode){
15         this.periode = periode;
16     }
17
18     public abstract void tampilkanLaporan();
19 }
20
21
```

2. LaporanPenjualan.java

```
1  package laporan;
2
3  import model.Transaksi;
4  import java.util.List;
5
6  public class LaporanPenjualan extends Laporan {
7      private List<Transaksi> listTransaksi;
8
9      public LaporanPenjualan(String periode, List<Transaksi> listTransaksi){
10         super(periode);
11         this.listTransaksi = listTransaksi;
12     }
13
14     @Override
15     public void tampilkanLaporan(){
16         System.out.println("\n=== Laporan Penjualan Periode: " + getPeriode() + " ===");
17         for (Transaksi transaksi : listTransaksi){
18             System.out.println("ID Laporan: " + transaksi.getID() + ", Nama Pelanggan: " + transaksi.getPelanggan() + ", Bunga: " + transaksi.getBunga() + ",
19
20         }
21     }
22 }
```

3. LaporanStok.java

```
1 package laporan;
2
3 import model.Bunga;
4 import java.util.List;
5
6 public class LaporanStok extends Laporan{
7     private List<Bunga> listBunga;
8
9     public LaporanStok(String periode, List<Bunga> listBunga){
10         super(periode);
11         this.listBunga = listBunga;
12     }
13
14     @Override
15     public void tampilkanLaporan(){
16         System.out.println("\n == Laporan Stok Bunga Periode: " + getPeriode() + " ==");
17         for (Bunga bunga : listBunga){
18             System.out.println("Kode: " + bunga.getKode() + ", Nama Bunga: " + bunga.getNama() + ", Stok Tersisa: " + bunga.getStock());
19         }
20     }
21 }
22
```

4. Transaksi.java

```
1 package model;
2
3 public class Transaksi {
4     private int id;
5     private Bunga bunga;
6     private int jumlah;
7     private double totalHarga;
8     private String pelanggan;
9
10     public Transaksi(int id, Bunga bunga, int jumlah, String pelanggan, double totalHarga){
11         this.id = id;
12         this.bunga = bunga;
13         this.jumlah = jumlah;
14         this.totalHarga = bunga.getHarga() * jumlah;
15         this.pelanggan = pelanggan;
16     }
17
18     public int getID(){
19         return id;
20     }
21     public void setId(int id){
22         this.id = id;
23     }
24     public Bunga getBunga(){
25         return bunga;
26     }
27     public void setBunga(Bunga bunga){
28         this.bunga = bunga;
29     }
30     public int getJumlah(){
31         return jumlah;
32     }
33
34     public void setJumlah(int jumlah, boolean updateHarga){
35         this.jumlah = jumlah;
36         if(updateHarga){
37             this.totalHarga = this.bunga.getHarga() * jumlah;
38         }
39     }
40     public double getTotalHarga(){
41         return totalHarga;
42     }
43     public void setTotalHarga(double totalHarga){
44         this.totalHarga = totalHarga;
45     }
46 }
```

```

47     public String getPelanggan(){
48         return pelanggan;
49     }
50
51     public void setPelanggan(String pelanggan){
52         this.pelanggan = pelanggan;
53     }
54 }

```

5. Toko Bunga.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 laporan.LaporanPenjualan;
8  import laporan.LaporanStok;
9
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
31              int pilih;
32
33              if (penggunaAktif != null) {
34                  do {
35                      System.out.println(x:"\n=====Sistem Penjualan Dan Stock Bunga=====");
36                      System.out.println(x:"1. Tambah Bunga (Admin)");
37                      System.out.println(x:"2. Perbarui Bunga (Admin)");
38                      System.out.println(x:"3. Hapus Bunga (Admin)");
39                      System.out.println(x:"4. Buat Transaksi Penjualan (Kasir)");
40                      System.out.println(x:"5. Lihat Laporan Penjualan (Admin)");
41                      System.out.println(x:"6. Lihat Laporan Stok Bunga (Admin)");
42                      System.out.println(x:"7. Logout");
43                      System.out.println(x:"8. Keluar");
44                      System.out.print(s:"Pilihan: ");
45                      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         System.out.println(x:"Logout berhasil. Silakan login kembali.");
74         penggunaAktif = null;
75         break;
76     case 8:
77         System.out.println(x:"Keluar dari sistem.");
78         isRunning = false;
79         break;
80     default:
81         System.out.println(x:"Pilihan tidak valid!");
82     }
83 } while (isRunning && penggunaAktif != null);
84 }
85 }
86 }

```

```

87     public static boolean login() {
88         System.out.print(s:"Masukkan Username: ");
89         String username = scanner.nextLine();
90         System.out.print(s:"Masukkan Password: ");
91         String pass = scanner.nextLine();
92
93         for (Pegguna pengguna : penggunaList) {
94             if (pengguna.getUsername().equals(username) && pengguna.getPass().equals(pass)) {
95                 penggunaAktif = pengguna;
96                 System.out.println("Login berhasil sebagai " + pengguna.getRole());
97                 return true;
98             }
99         }
100         System.out.println(x:"Username atau password salah!");
101         return false;
102     }
103
104     public static void tambahBunga() {
105         System.out.print(s:"Masukkan Kode Bunga: ");
106         int kode = scanner.nextInt();
107         scanner.nextLine();
108         System.out.print(s:"Masukkan Nama Bunga: ");
109         String nama = scanner.nextLine();
110         System.out.print(s:"Masukkan Stok: ");
111         int stock = scanner.nextInt();
112         System.out.print(s:"Masukkan Harga: ");
113         double harga = scanner.nextDouble();
114
115         daftarBunga.add(new Bunga(kode, nama, stock, harga));
116         System.out.println(x:"Bunga berhasil ditambahkan!");
117     }
118
119     public static void perbaruiBunga() {
120         System.out.print(s:"Masukkan kode Bunga yang akan diperbarui: ");
121         int kode = scanner.nextInt();
122         scanner.nextLine();
123         for (Bunga bunga : daftarBunga) {
124             if (bunga.getKode() == kode) {
125                 System.out.print(s:"Masukkan Nama Baru: ");
126                 bunga.setNama(scanner.nextLine());
127                 System.out.print(s:"Masukkan Stok Baru: ");
128                 bunga.setStock(scanner.nextInt());
129                 System.out.print(s:"Masukkan Harga Baru: ");
130                 bunga.setHarga(scanner.nextDouble());
131                 System.out.println(x:"Data bunga berhasil diperbarui!");
132                 return;
133             }
134         }
135         System.out.println(x:"Bunga tidak ditemukan!");
136     }
137
138     public static void hapusBunga() {
139         System.out.print(s:"Masukkan Kode Bunga yang akan dihapus: ");
140         int kode = scanner.nextInt();
141         scanner.nextLine();
142         daftarBunga.removeIf(bunga -> bunga.getKode() == kode);
143         System.out.println(x:"Bunga berhasil dihapus!");
144     }
145
146     public static void buatTransaksi() {
147         System.out.print(s:"Masukkan Kode Bunga yang Dibeli: ");
148         int kode = scanner.nextInt();
149         System.out.print(s:"Masukkan Jumlah yang Dibeli: ");
150         int jumlah = scanner.nextInt();
151         scanner.nextLine();
152         System.out.print(s:"Masukkan Nama Pelanggan: ");
153         String pelanggan = scanner.nextLine();
154
155         for (Bunga bunga : daftarBunga) {
156             if (bunga.getKode() == kode) {
157                 if (bunga.getStock() >= jumlah) {
158                     bunga.setStock(bunga.getStock() - jumlah);
159                     Transaksi transaksi = new Transaksi(daftarTransaksi.size() + 1, bunga, jumlah, pelanggan, totalHarga:0);
160                     transaksi.setJumlah(jumlah, updateHarga:true);
161
162                     daftarTransaksi.add(transaksi);
163                     System.out.println("Transaksi berhasil! Total Harga: Rp" + transaksi.getTotalHarga());
164                 } else {
165                     System.out.println(x:"Stok tidak mencukupi!");
166                 }
167                 return;
168             }
169         }
170         System.out.println(x:"Bunga tidak ditemukan!");
171     }

```

```
173     public static void lihatLaporanPenjualan(){
174         LaporanPenjualan laporan = new LaporanPenjualan(periode:"Maret 2025", daftarTransaksi);
175         laporan.tampilkanLaporan(); //test dulu bulannya sudah ditentukan
176     }
177
178     public static void lihatLaporanStok(){
179         LaporanStok laporan = new LaporanStok(periode:"Maret 2025", daftarBunga);
180         laporan.tampilkanLaporan();
181     }
182 }
```