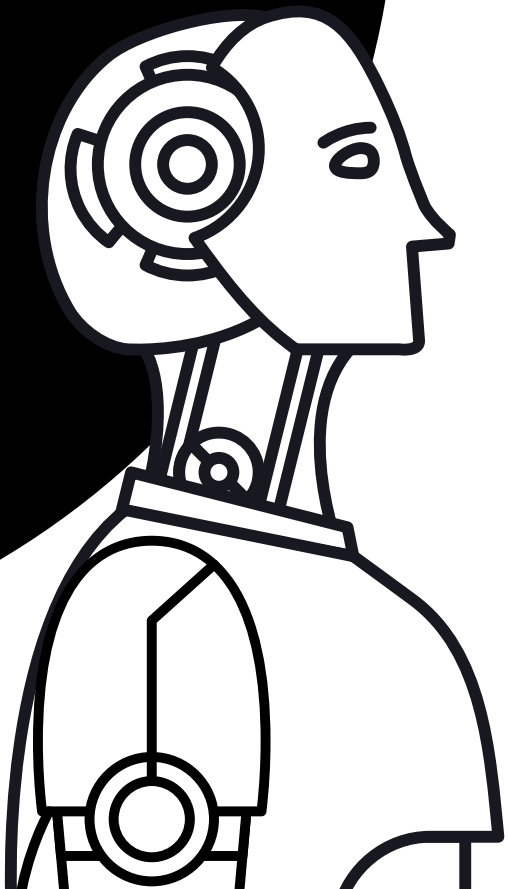


TUGAS BESAR

# TOKO KASIR BERBASIS OOP

PEMROGRAMAN  
BERORIENTASI OBJEK



# Fitur Utama

PAGE 02

## Tambah Barang

Sembako & Elektronik

## Laporan Penjualan

Rekap Transaksi



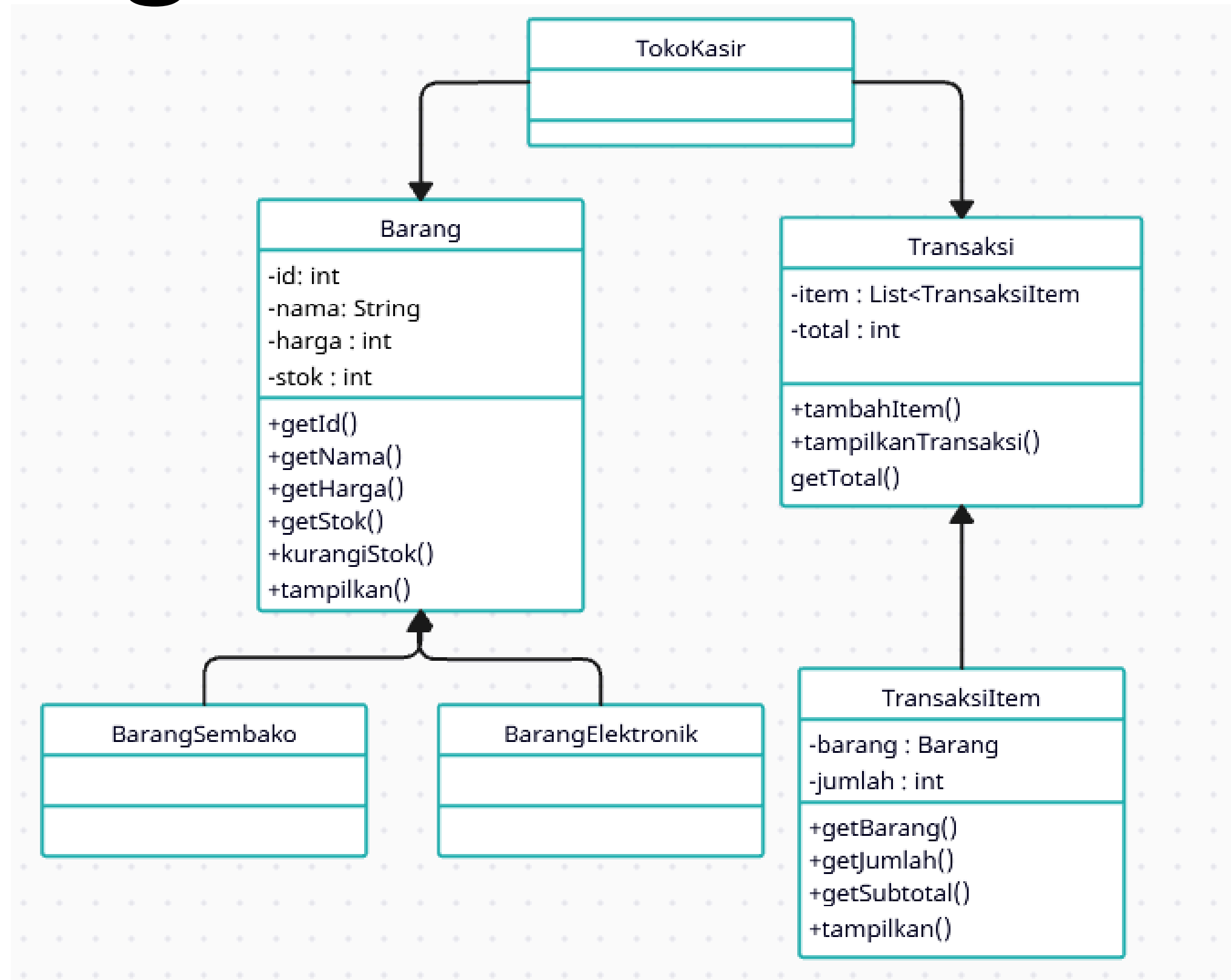
## Lihat Barang

Tampilkan Daftar Lengkap

## Transaksi

Kurangi stok & hitung total

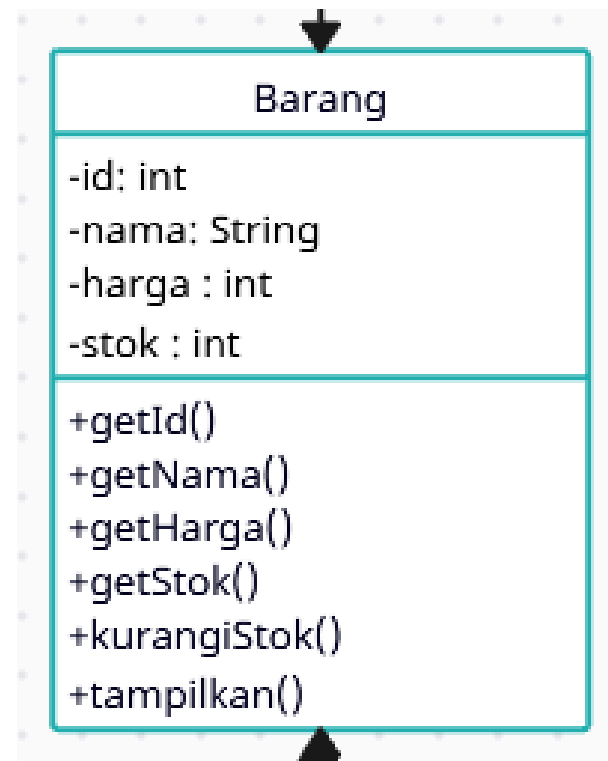
# Class Diagram



# Penjelasan Class

PAGE 04

```
1 public abstract class Barang {
2     private static int counter = 1;
3     private int id;
4     private String nama;
5     private int harga;
6     private int stok;
7
8     public Barang(String nama, int harga, int stok) {
9         this.id = counter++;
10        this.nama = nama;
11        this.harga = harga;
12        this.stok = stok;
13    }
14
15    public int getId() { return id; }
16    public String getNama() { return nama; }
17    public int getHarga() { return harga; }
18    public int getStok() { return stok; }
19
20    public void kurangiStok(int jumlah) {
21        this.stok -= jumlah;
22    }
23
24    // Abstract method → harus diimplementasikan oleh anaknya (abstraksi)
25    public abstract void tampilkan();
26 }
27
```



## Barang

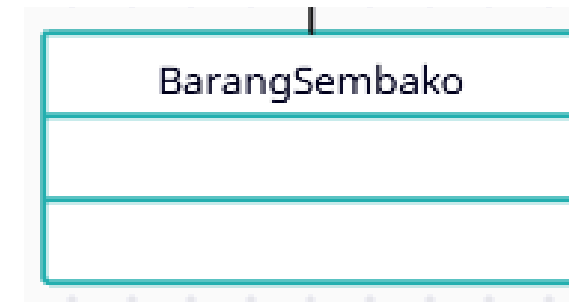
- Menyimpan data umum: id, nama, harga, stok.
- Memiliki method abstrak tampilkan() yang harus diimplementasikan oleh kelas turunannya.
- Menyediakan method untuk mengambil data (getter) dan mengurangi stok.

# Penjelasan Class

PAGE 05



```
1 public class BarangSembako extends Barang {
2     public BarangSembako(String nama, int harga, int stok) {
3         super(nama, harga, stok);
4     }
5
6     @Override
7     public void tampilkan() {
8         System.out.printf("| %-2d | %-12s | %-6d | %-5d | Jenis: Sembako\n",
9             getId(), getNama(), getHarga(), getStok());
10    }
11 }
12
```



## BarangSembako

- Kelas turunan dari Barang untuk barang kebutuhan pokok (sembako).
- Mengimplementasikan method tampilkan() untuk menampilkan informasi barang sembako di daftar barang.

# Penjelasan Class

PAGE 06



```
1 public class BarangElektronik extends Barang {
2     public BarangElektronik(String nama, int harga, int stok) {
3         super(nama, harga, stok);
4     }
5
6     @Override
7     public void tampilkan() {
8         System.out.printf("| %-2d | %-12s | %-6d | %-5d | Jenis: Elektronik\n",
9             getId(), getNama(), getHarga(), getStok());
10    }
11 }
12
```

BarangElektronik

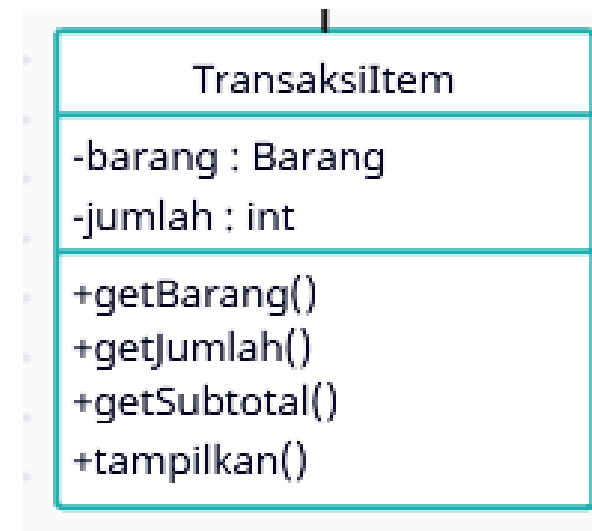
## BarangElektronik

- Kelas turunan dari Barang untuk barang elektronik.
- Mengimplementasikan method tampilkan() untuk menampilkan informasi barang elektronik di daftar barang.

# Penjelasan Class

PAGE 07

```
1 public class TransaksiItem {
2     private Barang barang;
3     private int jumlah;
4
5     public TransaksiItem(Barang barang, int jumlah) {
6         this.barang = barang;
7         this.jumlah = jumlah;
8     }
9
10    public Barang getBarang() { return barang; }
11    public int getJumlah() { return jumlah; }
12    public int getSubtotal() { return barang.getHarga() * jumlah; }
13
14    public void tampilkan() {
15        System.out.println("- " + barang.getNama() + ": " + jumlah + " pcs");
16    }
17 }
18
```



## TransaksiItem

- Kelas yang merepresentasikan satu item dalam transaksi penjualan.
- Menyimpan referensi ke objek Barang dan jumlah yang dibeli.
- Memiliki method untuk menghitung subtotal harga item (harga x jumlah).
- Digunakan dalam kelas Transaksi.

# Penjelasan Class

PAGE 08

```
1 import java.util.ArrayList;
2 import java.util.List;
3
4 public class Transaksi {
5     private List<TransaksiItem> items = new ArrayList<>();
6     private int total = 0;
7
8     public void tambahItem(TransaksiItem item) {
9         items.add(item);
10        total += item.getSubtotal();
11        item.getBarang().kurangiStok(item.getJumlah());
12    }
13
14    public void tampilkanTransaksi() {
15        for (TransaksiItem item : items) {
16            item.tampilkan();
17        }
18        System.out.println("Total: Rp " + total);
19    }
20
21    public int getTotal() {
22        return total;
23    }
24 }
25
```



## Transaksi

- Kelas yang merepresentasikan satu transaksi penjualan.
- Menyimpan daftar item yang dibeli (list of TransaksiItem).
- Menghitung total harga transaksi.
- Memiliki method untuk menambah item dan menampilkan detail transaksi.



# Penjelasan Class

PAGE 09

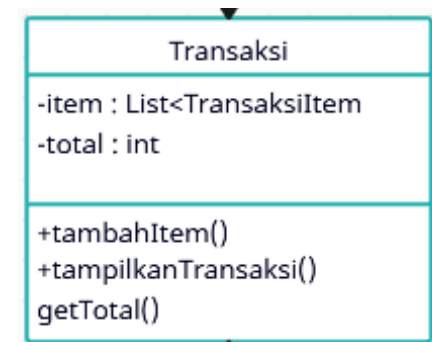
```
1 import java.util.*;
2
3 public class TokoKasir {
4     static Scanner input = new Scanner(System.in);
5     static List<Barang> daftarBarang = new ArrayList<>();
6     static List<Transaksi> laporan = new ArrayList<>();
7
8     public static void main(String[] args) {
9         while (true) {
10             System.out.println("=====");
11             System.out.println("SISTEM KASIR TOKO");
12             System.out.println("=====");
13             System.out.println("1. Tambah Barang");
14             System.out.println("2. Lihat Barang");
15             System.out.println("3. Transaksi Penjualan");
16             System.out.println("4. Laporan Penjualan");
17             System.out.println("5. Keluar");
18             System.out.print("Pilih menu: ");
19             int pilih = input.nextInt();
20
21             switch (pilih) {
22                 case 1: tambahBarang(); break;
23                 case 2: lihatBarang(); break;
24                 case 3: transaksiPenjualan(); break;
25                 case 4: laporanPenjualan(); break;
26                 case 5: System.exit(0);
27             }
28         }
29     }
30
31     static void tambahBarang() {
32         System.out.println("Jenis barang:");
33         System.out.println("1. Sembako");
34         System.out.println("2. Elektronik");
35         System.out.print("Pilih jenis: ");
36         int jenis = input.nextInt();
37     }
```

```
1         input.nextLine(); // consume newline
2         System.out.print("Nama barang: ");
3         String nama = input.nextLine();
4         System.out.print("Harga: ");
5         int harga = input.nextInt();
6         System.out.print("Stok: ");
7         int stok = input.nextInt();
8
9
10        if (jenis == 1) {
11            daftarBarang.add(new BarangSembako(nama, harga, stok));
12        } else if (jenis == 2) {
13            daftarBarang.add(new BarangElektronik(nama, harga, stok)); // tanpa garansi
14        }
15
16        System.out.println("Barang berhasil ditambahkan!");
17    }
18
19    static void lihatBarang() {
20        System.out.println("+---+-----+-----+-----+");
21        System.out.println("| ID | Nama      | Harga | Stok | Keterangan |");
22        System.out.println("+---+-----+-----+-----+");
23        for (Barang b : daftarBarang) {
24            b.tampilkan(); // Polimorfisme: tampilkan sesuai jenisnya
25        }
26        System.out.println("+---+-----+-----+-----+");
27    }
28
29    static void transaksiPenjualan() {
30        System.out.print("Masukkan ID barang: ");
31        int id = input.nextInt();
32        Barang barang = cariBarang(id);
33        if (barang == null) {
34            System.out.println("Barang tidak ditemukan!");
35            return;
36        }
37
38        System.out.print("Masukkan jumlah: ");
39        int jumlah = input.nextInt();
40        int subtotal = barang.getHarga() * jumlah;
41
42        System.out.println("Nama Barang : " + barang.getNama());
43        System.out.println("Harga Satuan : Rp " + barang.getHarga());
44        System.out.println("Subtotal : Rp " + subtotal);
45    }
```

```
1
2        System.out.print("Konfirmasi transaksi? (y/n): ");
3        input.nextLine(); // consume newline
4        String konfirmasi = input.nextLine();
5        if (konfirmasi.equalsIgnoreCase("y")) {
6            Transaksi transaksi = new Transaksi();
7            transaksi.tambahItem(new TransaksiItem(barang, jumlah));
8            laporan.add(transaksi);
9            System.out.println("Stok berhasil dikurangi.");
10           System.out.println("Transaksi berhasil disimpan!");
11       } else {
12           System.out.println("Transaksi dibatalkan.");
13       }
14   }
15
16   static Barang cariBarang(int id) {
17       for (Barang b : daftarBarang) {
18           if (b.getId() == id) return b;
19       }
20       return null;
21   }
22
23   static void laporanPenjualan() {
24       System.out.println("Laporan Transaksi Penjualan");
25       int totalHarian = 0;
26       int no = 1;
27       for (Transaksi t : laporan) {
28           System.out.println("Transaksi " + no + ":");
29           t.tampilkanTransaksi();
30           System.out.println();
31           totalHarian += t.getTotal();
32           no++;
33       }
34       System.out.println("-----");
35       System.out.println("Total pendapatan hari ini: Rp " + totalHarian);
36   }
37 }
38
```

- Kelas utama yang menjalankan aplikasi kasir.
- Menyediakan menu utama (tambah barang, lihat barang, transaksi, laporan).
- Mengelola daftar barang dan laporan transaksi.
- Menjadi penghubung antara user dan seluruh proses bisnis aplikasi.

## TokoKasir



# Penerapan 4 Pilar OOP

PAGE 10

## Enkapsulasi (Encapsulation)

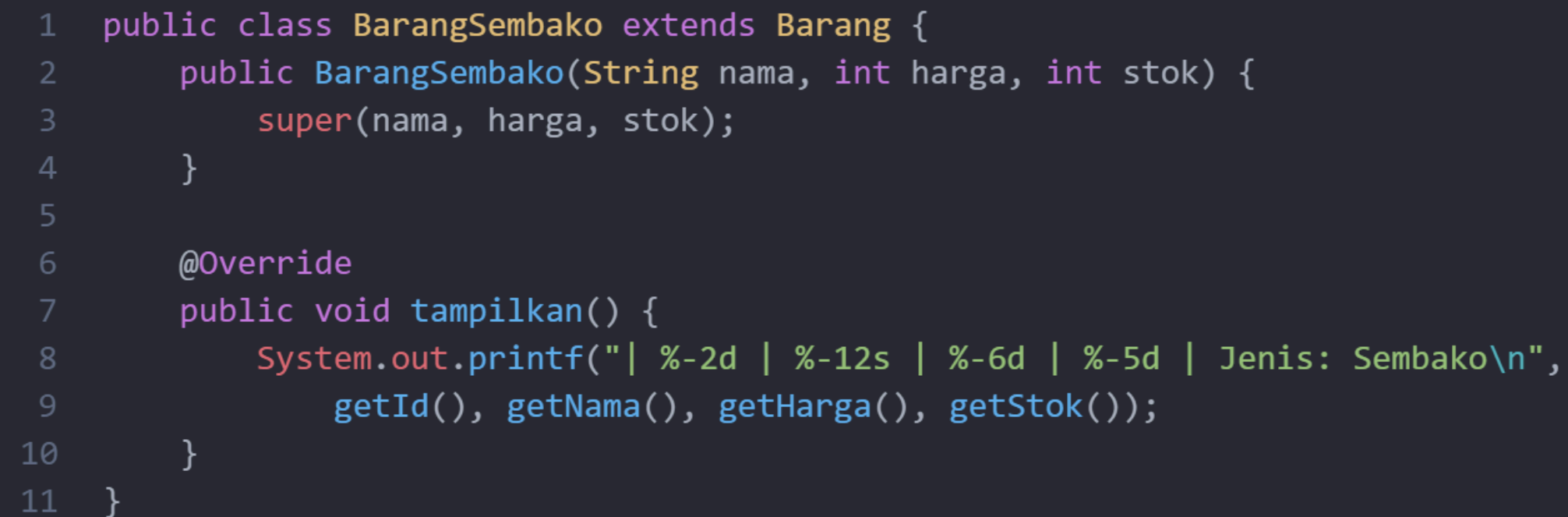
```
1 public abstract class Barang {
2     private static int counter = 1;
3     private int id;
4     private String nama;
5     private int harga;
6     private int stok;
7
8     public Barang(String nama, int harga, int stok) {
9         this.id = counter++;
10        this.nama = nama;
11        this.harga = harga;
12        this.stok = stok;
13    }
14
15    public int getId() { return id; }
16    public String getName() { return nama; }
17    public int getHarga() { return harga; }
18    public int getStok() { return stok; }
19
20    public void kurangiStok(int jumlah) {
21        this.stok -= jumlah;
22    }
23
24    // Abstract method → harus diimplementasikan oleh anaknya (abstraksi)
25    public abstract void tampilkan();
26 }
27
```

- Semua atribut pada kelas dibuat private.
- Akses data hanya melalui method getter/setter.

# Penerapan 4 Pilar OOP

PAGE 11

## Pewarisan (Inheritance)



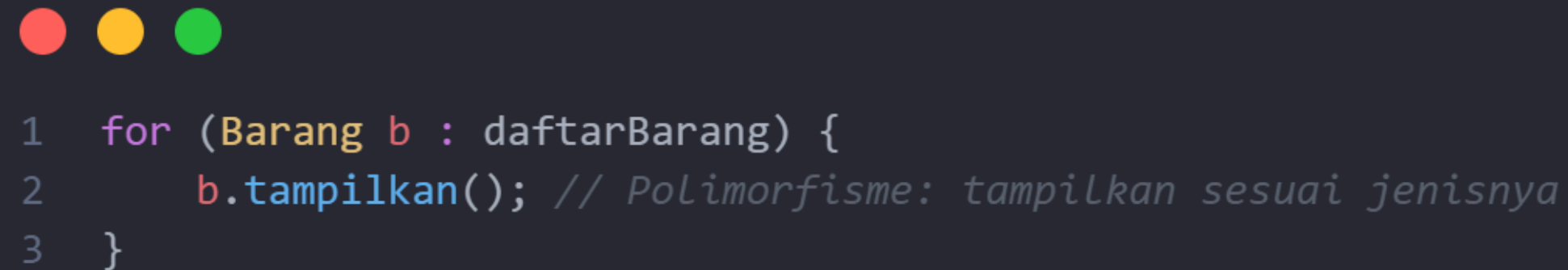
```
1 public class BarangSembako extends Barang {
2     public BarangSembako(String nama, int harga, int stok) {
3         super(nama, harga, stok);
4     }
5
6     @Override
7     public void tampilkan() {
8         System.out.printf("| %-2d | %-12s | %-6d | %-5d | Jenis: Sembako\n",
9             getId(), getNama(), getHarga(), getStok());
10    }
11 }
```

- BarangSembako dan BarangElektronik mewarisi dari Barang.

# Penerapan 4 Pilar OOP

PAGE 12

## Polimorfisme (Polymorphism)



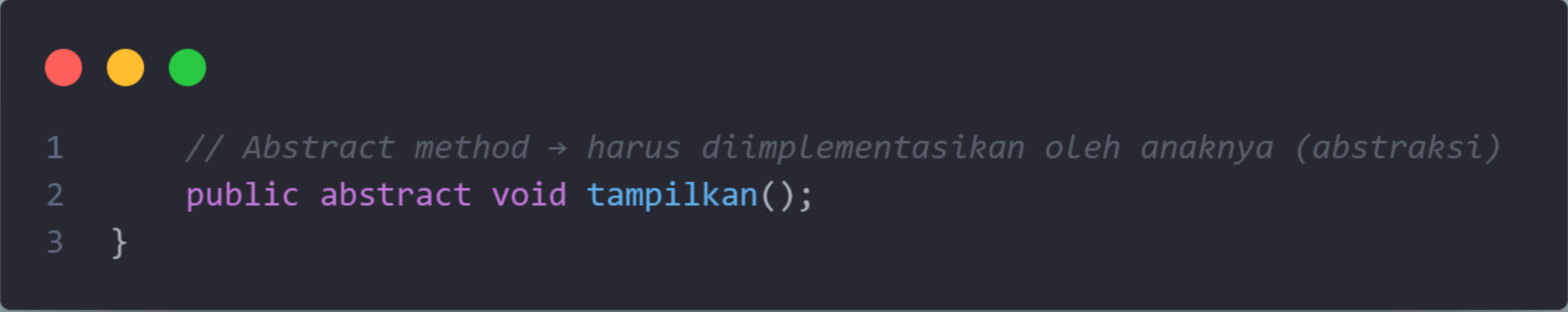
```
1  for (Barang b : daftarBarang) {  
2      b.tampilkan(); // Polimorfisme: tampilkan sesuai jenisnya  
3  }
```

- Method tampilkan() di-override pada setiap turunan Barang.
- Saat menampilkan daftar barang, method tampilkan() yang dipanggil sesuai jenis objeknya.

# Penerapan 4 Pilar OOP

PAGE 13

## Abstraksi (Abstraction)

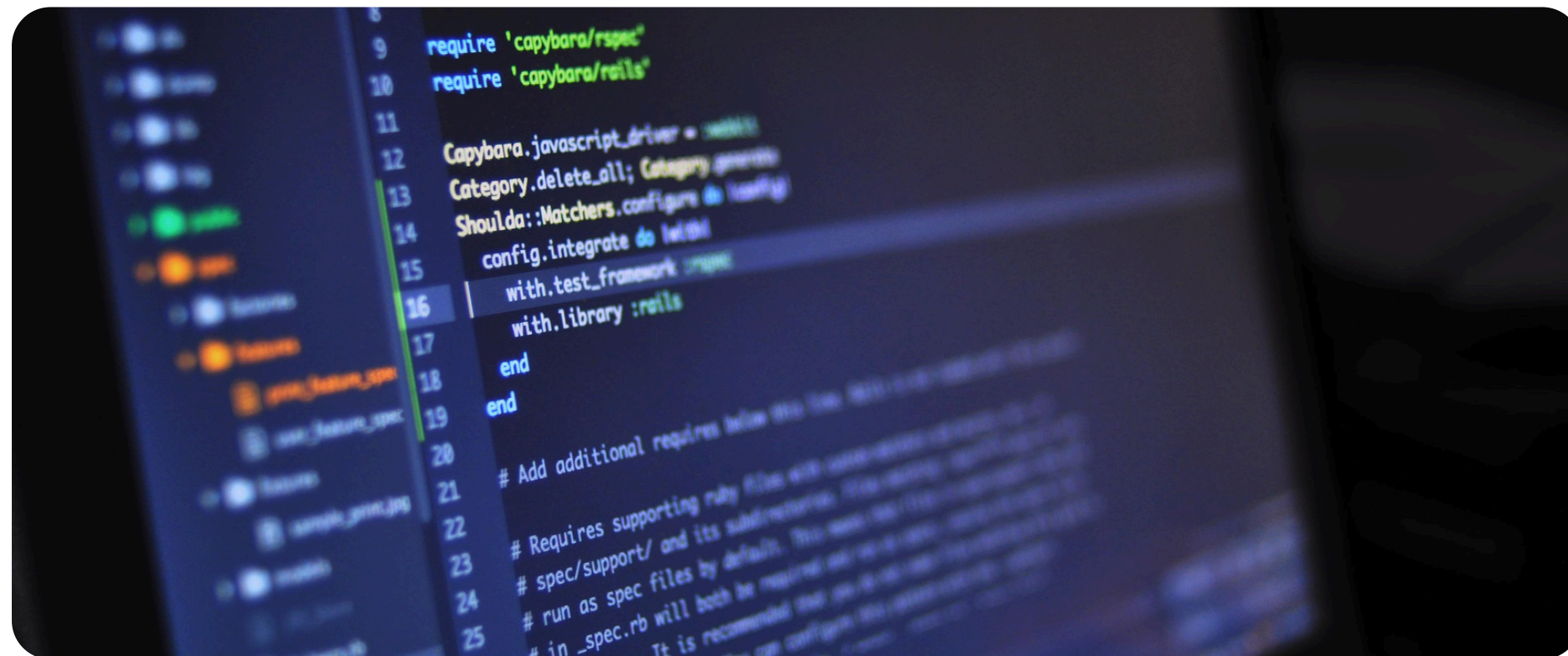


```
1 // Abstract method → harus diimplementasikan oleh anaknya (abstraksi)
2 public abstract void tampilkan();
3 }
```

- Kelas Barang dideklarasikan sebagai abstract dan memiliki method abstrak tampilkan().
- Kelas turunan wajib mengimplementasikan method abstrak tersebut.



# THANK YOU FOR YOUR ATTENTION



Link Github