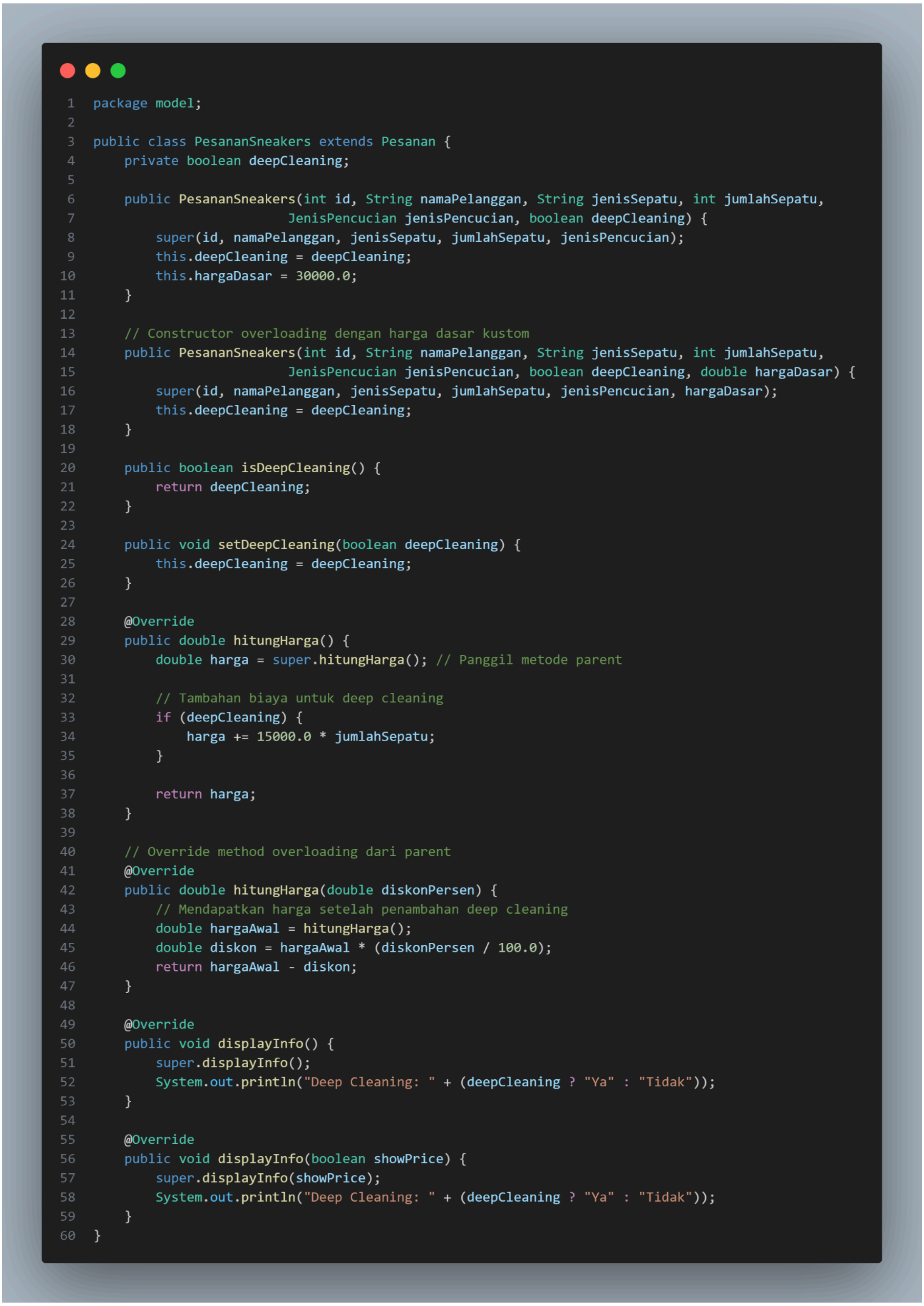
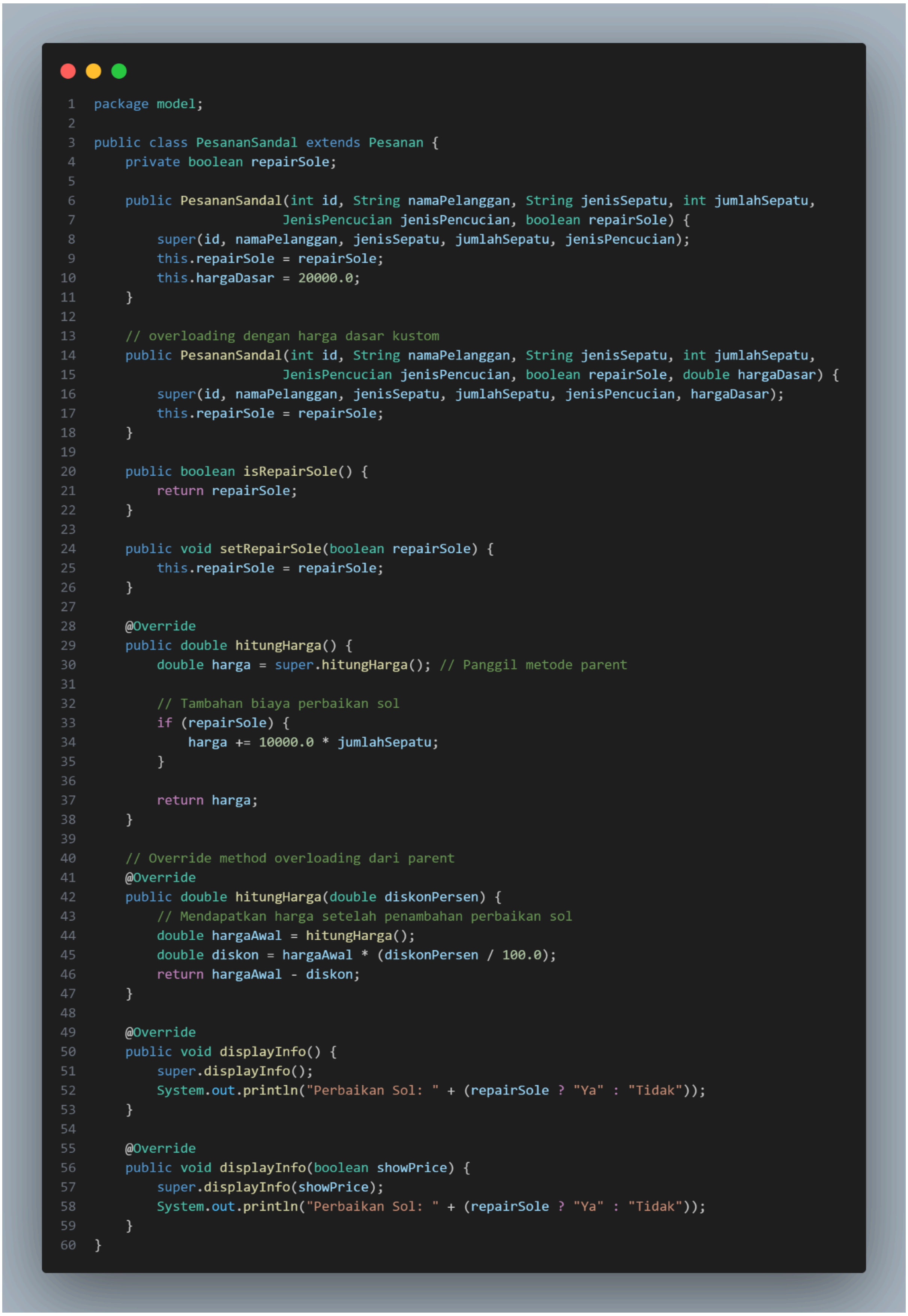
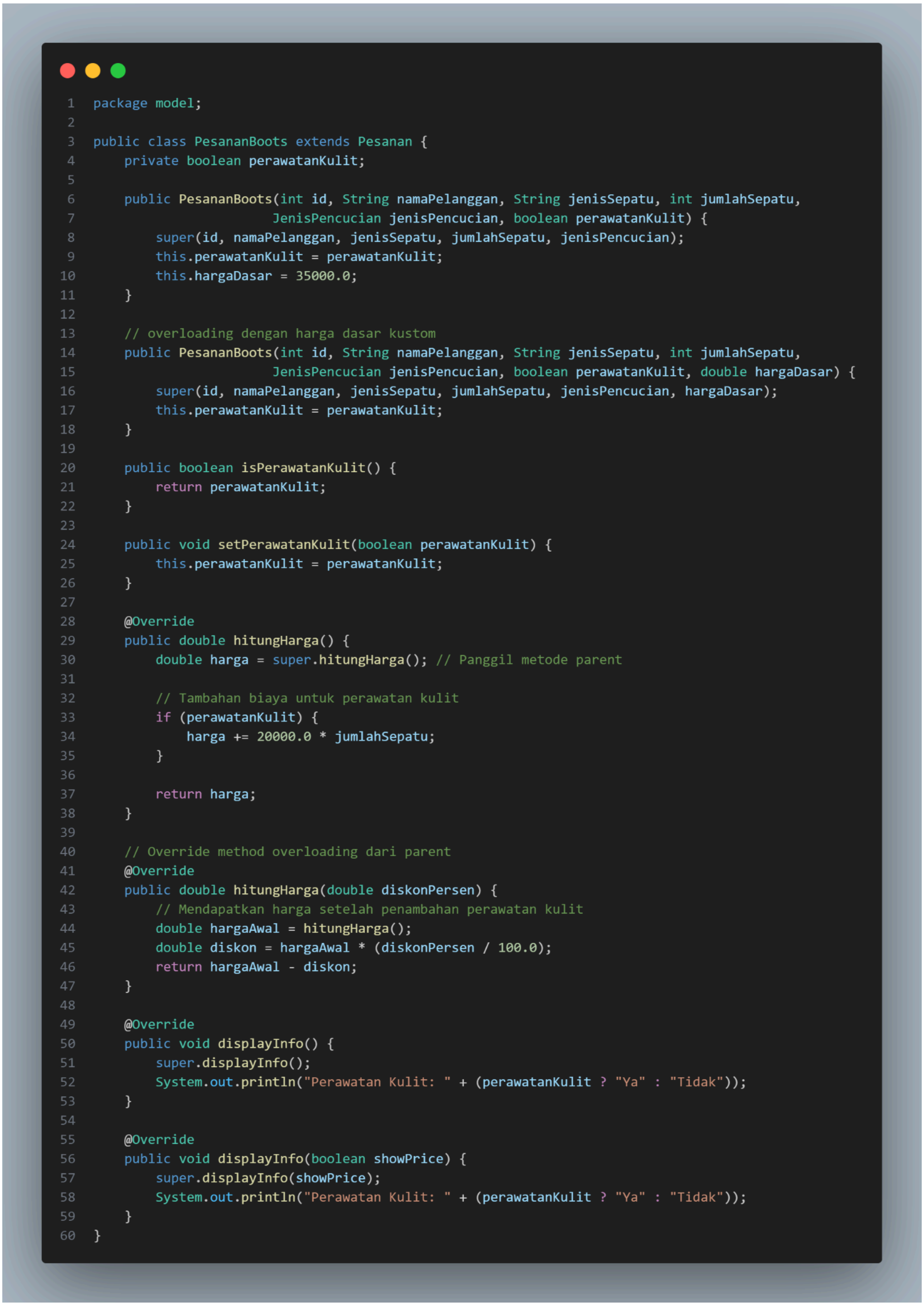
[illegible]



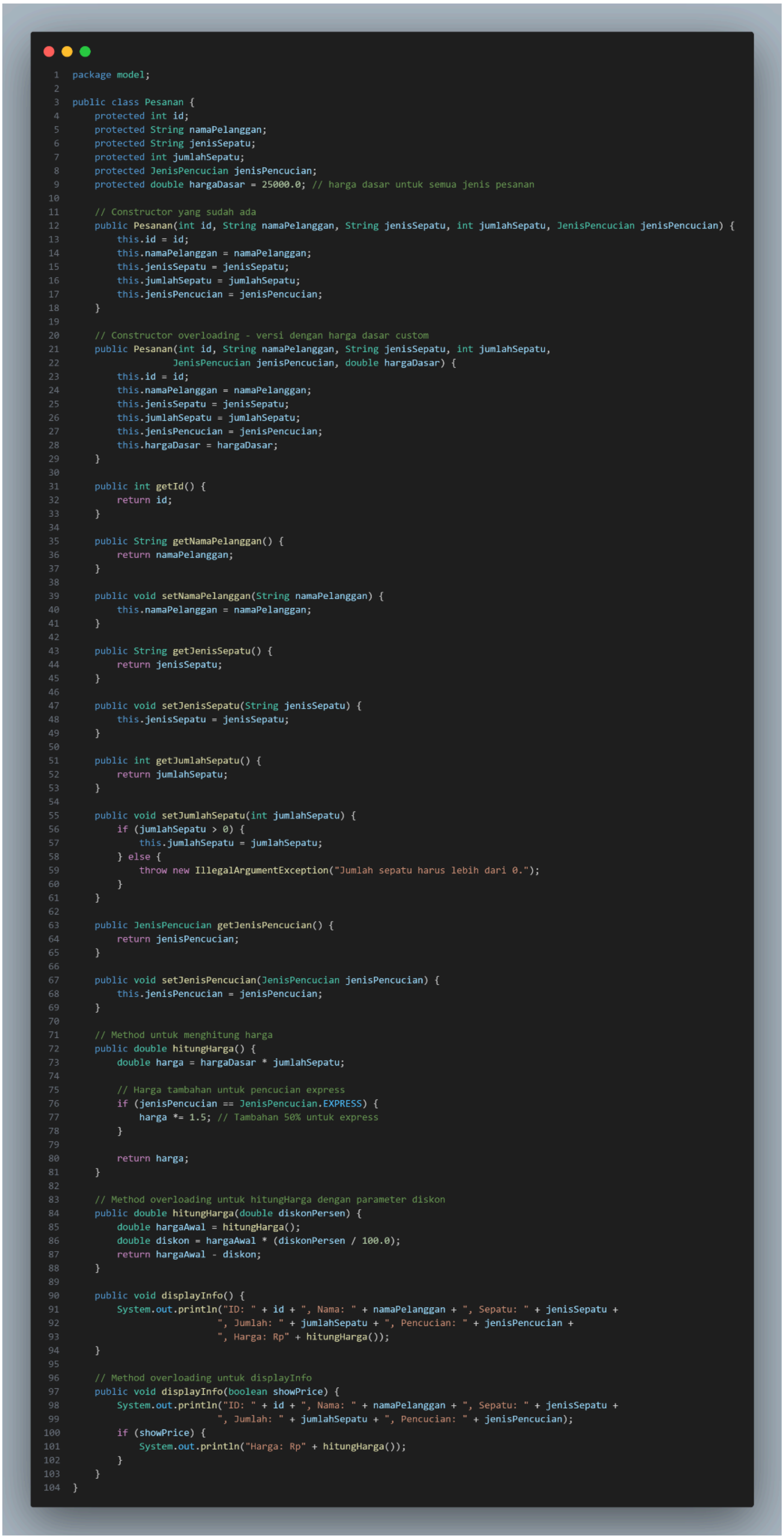
```java
package model;

public class PesananSneakers extends Pesanan {
    private boolean deepCleaning;

    public PesananSneakers(int id, String namaPelanggan, String jenisSepatu, int jumlahSepatu,
                          JenisPencucian jenisPencucian, boolean deepCleaning) {
        super(id, namaPelanggan, jenisSepatu, jumlahSepatu, jenisPencucian);
        this.deepCleaning = deepCleaning;
        this.hargaDasar = 30000.0;
    }

    // Constructor overloading dengan harga dasar kustom
    public PesananSneakers(int id, String namaPelanggan, String jenisSepatu, int jumlahSepatu,
                          JenisPencucian jenisPencucian, boolean deepCleaning, double hargaDasar) {
        super(id, namaPelanggan, jenisSepatu, jumlahSepatu, jenisPencucian, hargaDasar);
        this.deepCleaning = deepCleaning;
    }

    public boolean isDeepCleaning() {
        return deepCleaning;
    }

    public void setDeepCleaning(boolean deepCleaning) {
        this.deepCleaning = deepCleaning;
    }

    @Override
    public double hitungHarga() {
        double harga = super.hitungHarga(); // Panggil metode parent

        // Tambahan biaya untuk deep cleaning
        if (deepCleaning) {
            harga += 15000.0 * jumlahSepatu;
        }

        return harga;
    }

    // Override method overloading dari parent
    @Override
    public double hitungHarga(double diskonPersen) {
        // Mendapatkan harga setelah penambahan deep cleaning
        double hargaAwal = hitungHarga();
        double diskon = hargaAwal * (diskonPersen / 100.0);
        return hargaAwal - diskon;
    }

    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Deep Cleaning: " + (deepCleaning ? "Ya" : "Tidak"));
    }

    @Override
    public void displayInfo(boolean showPrice) {
        super.displayInfo(showPrice);
        System.out.println("Deep Cleaning: " + (deepCleaning ? "Ya" : "Tidak"));
    }
}
```



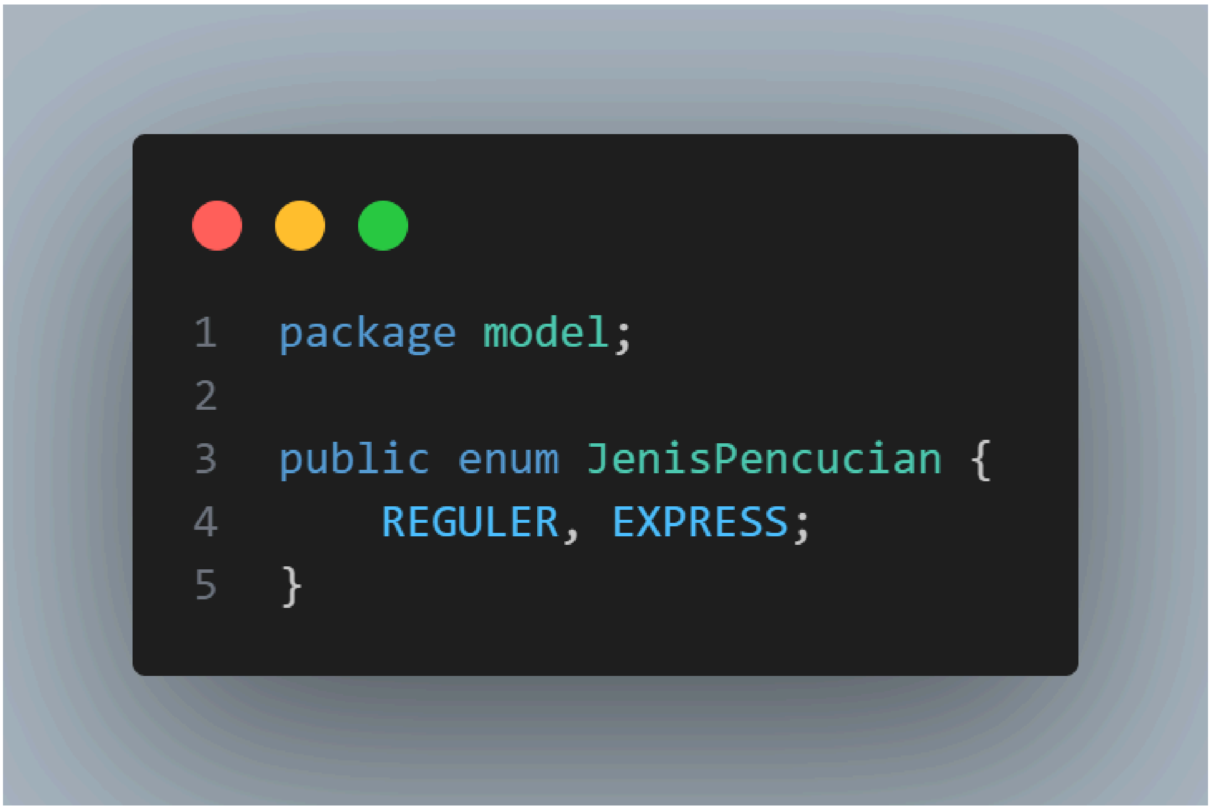
```java
package model;

public class PesananSandal extends Pesanan {
    private boolean repairSole;

    public PesananSandal(int id, String namaPelanggan, String jenisSepatu, int jumlahSepatu,
                        JenisPencucian jenisPencucian, boolean repairSole) {
        super(id, namaPelanggan, jenisSepatu, jumlahSepatu, jenisPencucian);
        this.repairSole = repairSole;
        this.hargaDasar = 20000.0;
    }

    // overloading dengan harga dasar kustom
    public PesananSandal(int id, String namaPelanggan, String jenisSepatu, int jumlahSepatu,
                        JenisPencucian jenisPencucian, boolean repairSole, double hargaDasar) {
        super(id, namaPelanggan, jenisSepatu, jumlahSepatu, jenisPencucian, hargaDasar);
        this.repairSole = repairSole;
    }

    public boolean isRepairSole() {
        return repairSole;
    }

    public void setRepairSole(boolean repairSole) {
        this.repairSole = repairSole;
    }

    @Override
    public double hitungHarga() {
        double harga = super.hitungHarga(); // Panggil metode parent

        // Tambahan biaya perbaikan sol
        if (repairSole) {
            harga += 10000.0 * jumlahSepatu;
        }

        return harga;
    }

    // Override method overloading dari parent
    @Override
    public double hitungHarga(double diskonPersen) {
        // Mendapatkan harga setelah penambahan perbaikan sol
        double hargaAwal = hitungHarga();
        double diskon = hargaAwal * (diskonPersen / 100.0);
        return hargaAwal - diskon;
    }

    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Perbaikan Sol: " + (repairSole ? "Ya" : "Tidak"));
    }

    @Override
    public void displayInfo(boolean showPrice) {
        super.displayInfo(showPrice);
        System.out.println("Perbaikan Sol: " + (repairSole ? "Ya" : "Tidak"));
    }
}
```



```java
package model;

public class PesananBoots extends Pesanan {
    private boolean perawatanKulit;

    public PesananBoots(int id, String namaPelanggan, String jenisSepatu, int jumlahSepatu,
                       JenisPencucian jenisPencucian, boolean perawatanKulit) {
        super(id, namaPelanggan, jenisSepatu, jumlahSepatu, jenisPencucian);
        this.perawatanKulit = perawatanKulit;
        this.hargaDasar = 35000.0;
    }

    // overloading dengan harga dasar kustom
    public PesananBoots(int id, String namaPelanggan, String jenisSepatu, int jumlahSepatu,
                       JenisPencucian jenisPencucian, boolean perawatanKulit, double hargaDasar) {
        super(id, namaPelanggan, jenisSepatu, jumlahSepatu, jenisPencucian, hargaDasar);
        this.perawatanKulit = perawatanKulit;
    }

    public boolean isPerawatanKulit() {
        return perawatanKulit;
    }

    public void setPerawatanKulit(boolean perawatanKulit) {
        this.perawatanKulit = perawatanKulit;
    }

    @Override
    public double hitungHarga() {
        double harga = super.hitungHarga(); // Panggil metode parent

        // Tambahan biaya untuk perawatan kulit
        if (perawatanKulit) {
            harga += 20000.0 * jumlahSepatu;
        }

        return harga;
    }

    // Override method overloading dari parent
    @Override
    public double hitungHarga(double diskonPersen) {
        // Mendapatkan harga setelah penambahan perawatan kulit
        double hargaAwal = hitungHarga();
        double diskon = hargaAwal * (diskonPersen / 100.0);
        return hargaAwal - diskon;
    }

    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Perawatan Kulit: " + (perawatanKulit ? "Ya" : "Tidak"));
    }

    @Override
    public void displayInfo(boolean showPrice) {
        super.displayInfo(showPrice);
        System.out.println("Perawatan Kulit: " + (perawatanKulit ? "Ya" : "Tidak"));
    }
}
```



```java
package model;

public class Pesanan {
    protected int id;
    protected String namaPelanggan;
    protected String jenisSepatu;
    protected int jumlahSepatu;
    protected JenisPencucian jenisPencucian;
    protected double hargaDasar = 25000.0; // harga dasar untuk semua jenis pesanan

    // Constructor yang sudah ada
    public Pesanan(int id, String namaPelanggan, String jenisSepatu, int jumlahSepatu, JenisPencucian jenisPencucian) {
        this.id = id;
        this.namaPelanggan = namaPelanggan;
        this.jenisSepatu = jenisSepatu;
        this.jumlahSepatu = jumlahSepatu;
        this.jenisPencucian = jenisPencucian;
    }

    // Constructor overloading - versi dengan harga dasar custom
    public Pesanan(int id, String namaPelanggan, String jenisSepatu, int jumlahSepatu,
                  JenisPencucian jenisPencucian, double hargaDasar) {
        this.id = id;
        this.namaPelanggan = namaPelanggan;
        this.jenisSepatu = jenisSepatu;
        this.jumlahSepatu = jumlahSepatu;
        this.jenisPencucian = jenisPencucian;
        this.hargaDasar = hargaDasar;
    }

    public int getId() {
        return id;
    }

    public String getNamaPelanggan() {
        return namaPelanggan;
    }

    public void setNamaPelanggan(String namaPelanggan) {
        this.namaPelanggan = namaPelanggan;
    }

    public String getJenisSepatu() {
        return jenisSepatu;
    }

    public void setJenisSepatu(String jenisSepatu) {
        this.jenisSepatu = jenisSepatu;
    }

    public int getJumlahSepatu() {
        return jumlahSepatu;
    }

    public void setJumlahSepatu(int jumlahSepatu) {
        if (jumlahSepatu > 0) {
            this.jumlahSepatu = jumlahSepatu;
        } else {
            throw new IllegalArgumentException("Jumlah sepatu harus lebih dari 0.");
        }
    }

    public JenisPencucian getJenisPencucian() {
        return jenisPencucian;
    }

    public void setJenisPencucian(JenisPencucian jenisPencucian) {
        this.jenisPencucian = jenisPencucian;
    }

    // Method untuk menghitung harga
    public double hitungHarga() {
        double harga = hargaDasar * jumlahSepatu;

        // Harga tambahan untuk pencucian express
        if (jenisPencucian == JenisPencucian.EXPRESS) {
            harga *= 1.5; // Tambahan 50% untuk express
        }

        return harga;
    }

    // Method overloading untuk hitungHarga dengan parameter diskon
    public double hitungHarga(double diskonPersen) {
        double hargaAwal = hitungHarga();
        double diskon = hargaAwal * (diskonPersen / 100.0);
        return hargaAwal - diskon;
    }

    public void displayInfo() {
        System.out.println("ID: " + id + ", Nama: " + namaPelanggan + ", Sepatu: " + jenisSepatu +
                          ", Jumlah: " + jumlahSepatu + ", Pencucian: " + jenisPencucian +
                          ", Harga: Rp" + hitungHarga());
    }

    // Method overloading untuk displayInfo
    public void displayInfo(boolean showPrice) {
        System.out.println("ID: " + id + ", Nama: " + namaPelanggan + ", Sepatu: " + jenisSepatu +
                          ", Jumlah: " + jumlahSepatu + ", Pencucian: " + jenisPencucian);
        if (showPrice) {
            System.out.println("Harga: Rp" + hitungHarga());
        }
    }
}
```



```java
package model;

public enum JenisPencucian {
    REGULER, EXPRESS;
}
```