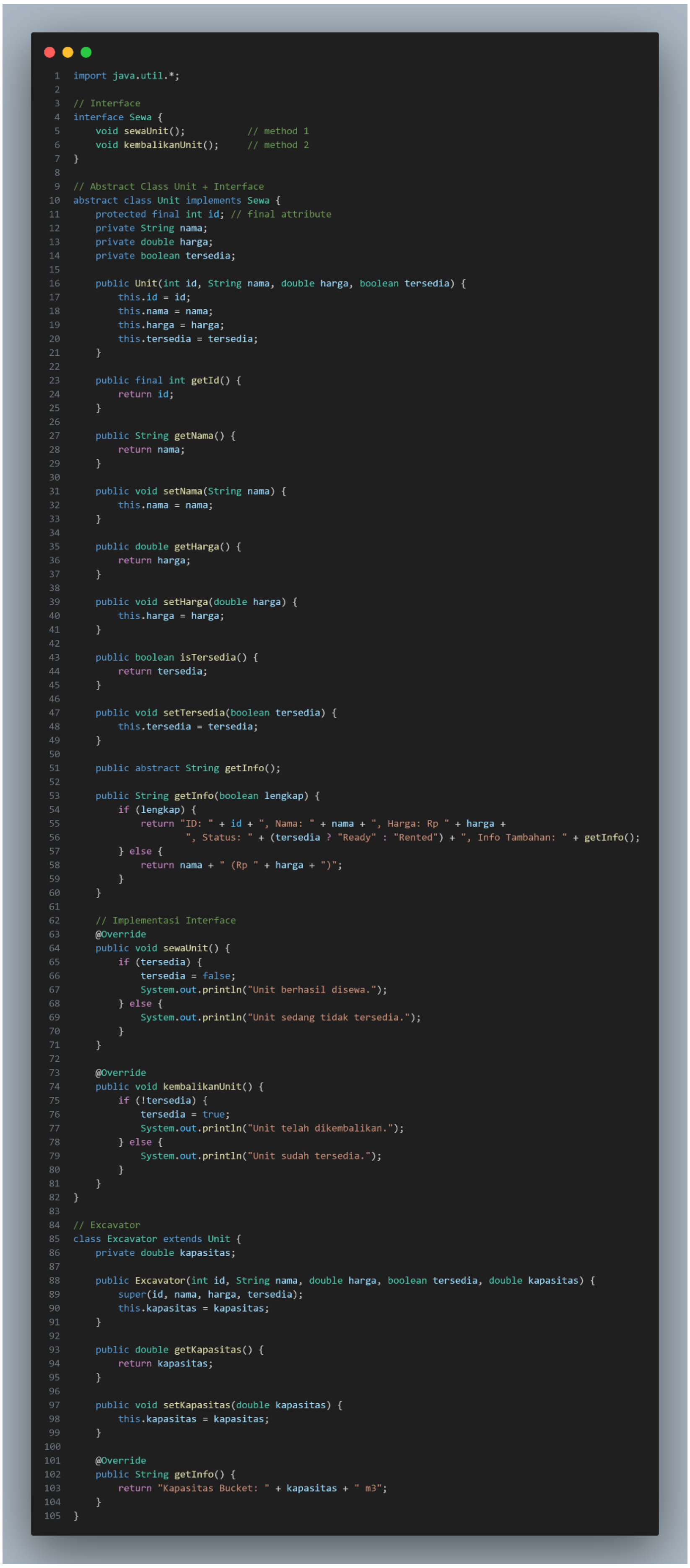
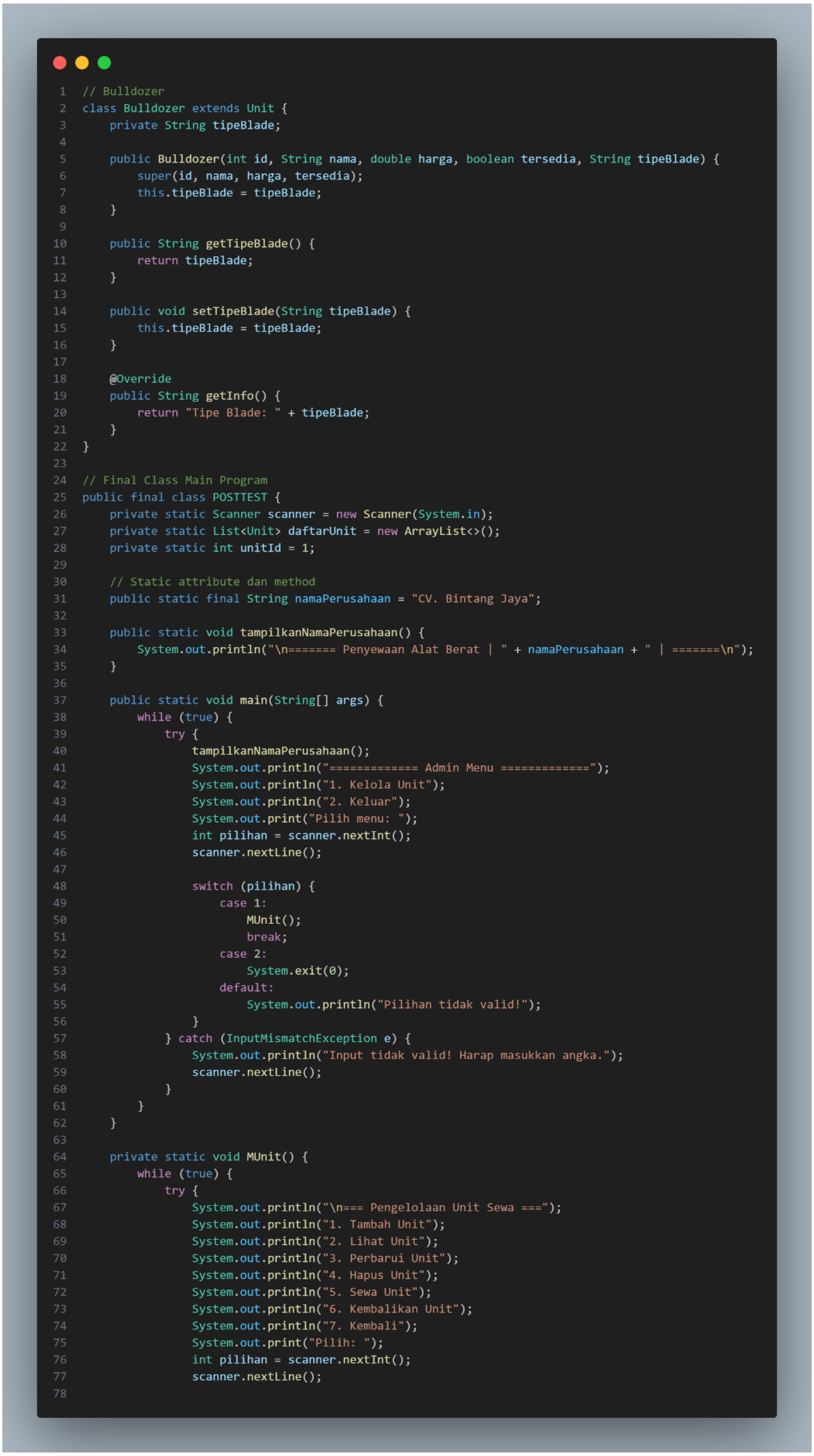
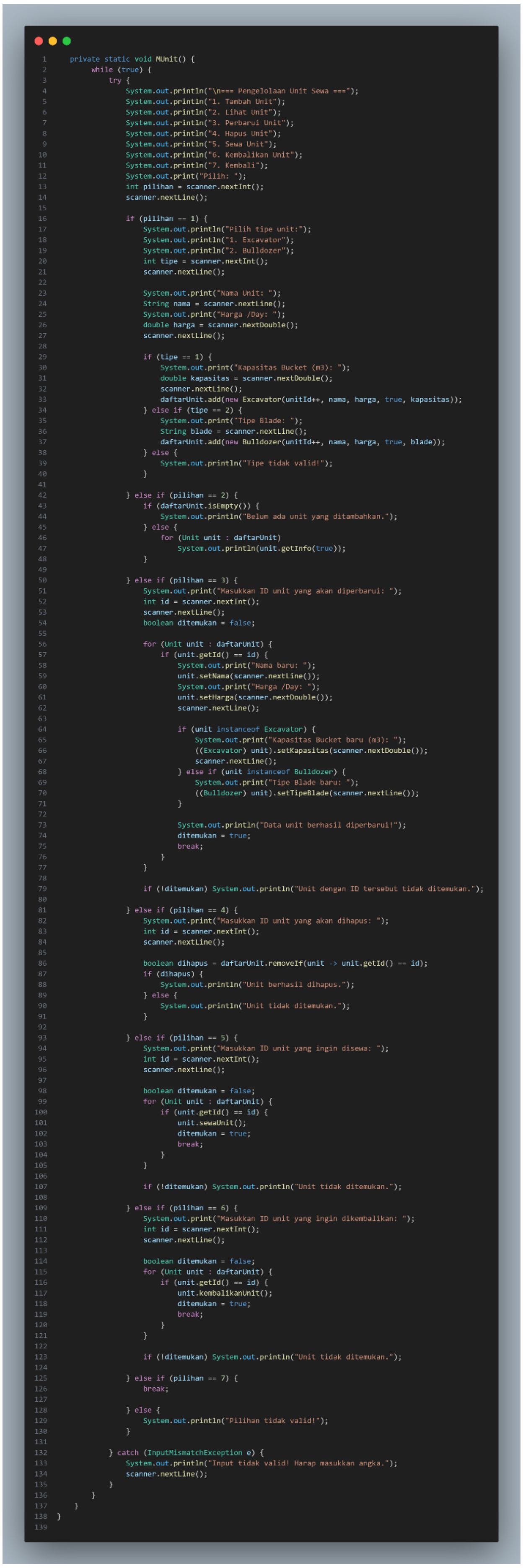


```java
import java.util.*;

// Interface
interface Sewa {
    void sewaUnit();           // method 1
    void kembalikanUnit();     // method 2
}

// Abstract Class Unit + Interface
abstract class Unit implements Sewa {
    protected final int id; // final attribute
    private String nama;
    private double harga;
    private boolean tersedia;

    public Unit(int id, String nama, double harga, boolean tersedia) {
        this.id = id;
        this.nama = nama;
        this.harga = harga;
        this.tersedia = tersedia;
    }

    public final int getId() {
        return id;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public double getHarga() {
        return harga;
    }

    public void setHarga(double harga) {
        this.harga = harga;
    }

    public boolean isTersedia() {
        return tersedia;
    }

    public void setTersedia(boolean tersedia) {
        this.tersedia = tersedia;
    }

    public abstract String getInfo();

    public String getInfo(boolean lengkap) {
        if (lengkap) {
            return "ID: " + id + ", Nama: " + nama + ", Harga: Rp " + harga +
                    ", Status: " + (tersedia ? "Ready" : "Rented") + ", Info Tambahan: " + getInfo();
        } else {
            return nama + " (Rp " + harga + ")";
        }
    }

    // Implementasi Interface
    @Override
    public void sewaUnit() {
        if (tersedia) {
            tersedia = false;
            System.out.println("Unit berhasil disewa.");
        } else {
            System.out.println("Unit sedang tidak tersedia.");
        }
    }

    @Override
    public void kembalikanUnit() {
        if (!tersedia) {
            tersedia = true;
            System.out.println("Unit telah dikembalikan.");
        } else {
            System.out.println("Unit sudah tersedia.");
        }
    }
}

// Excavator
class Excavator extends Unit {
    private double kapasitas;

    public Excavator(int id, String nama, double harga, boolean tersedia, double kapasitas) {
        super(id, nama, harga, tersedia);
        this.kapasitas = kapasitas;
    }

    public double getKapasitas() {
        return kapasitas;
    }

    public void setKapasitas(double kapasitas) {
        this.kapasitas = kapasitas;
    }

    @Override
    public String getInfo() {
        return "Kapasitas Bucket: " + kapasitas + " m3";
    }
}
```



```java
// Bulldozer
class Bulldozer extends Unit {
    private String tipeBlade;

    public Bulldozer(int id, String nama, double harga, boolean tersedia, String tipeBlade) {
        super(id, nama, harga, tersedia);
        this.tipeBlade = tipeBlade;
    }

    public String getTipeBlade() {
        return tipeBlade;
    }

    public void setTipeBlade(String tipeBlade) {
        this.tipeBlade = tipeBlade;
    }

    @Override
    public String getInfo() {
        return "Tipe Blade: " + tipeBlade;
    }
}

// Final Class Main Program
public final class POSTTEST {
    private static Scanner scanner = new Scanner(System.in);
    private static List<Unit> daftarUnit = new ArrayList<>();
    private static int unitId = 1;

    // Static attribute dan method
    public static final String namaPerusahaan = "CV. Bintang Jaya";

    public static void tampilkanNamaPerusahaan() {
        System.out.println("\n======= Penyewaan Alat Berat | " + namaPerusahaan + " | =======\n");
    }

    public static void main(String[] args) {
        while (true) {
            try {
                tampilkanNamaPerusahaan();
                System.out.println("============= Admin Menu =============");
                System.out.println("1. Kelola Unit");
                System.out.println("2. Keluar");
                System.out.print("Pilih menu: ");
                int pilihan = scanner.nextInt();
                scanner.nextLine();

                switch (pilihan) {
                    case 1:
                        MUnit();
                        break;
                    case 2:
                        System.exit(0);
                    default:
                        System.out.println("Pilihan tidak valid!");
                }
            } catch (InputMismatchException e) {
                System.out.println("Input tidak valid! Harap masukkan angka.");
                scanner.nextLine();
            }
        }
    }

    private static void MUnit() {
        while (true) {
            try {
                System.out.println("\n=== Pengelolaan Unit Sewa ===");
                System.out.println("1. Tambah Unit");
                System.out.println("2. Lihat Unit");
                System.out.println("3. Perbarui Unit");
                System.out.println("4. Hapus Unit");
                System.out.println("5. Sewa Unit");
                System.out.println("6. Kembalikan Unit");
                System.out.println("7. Kembali");
                System.out.print("Pilih: ");
                int pilihan = scanner.nextInt();
                scanner.nextLine();

```



```java
private static void MUnit() {
    while (true) {
        try {
            System.out.println("\n=== Pengelolaan Unit Sewa ===");
            System.out.println("1. Tambah Unit");
            System.out.println("2. Lihat Unit");
            System.out.println("3. Perbarui Unit");
            System.out.println("4. Hapus Unit");
            System.out.println("5. Sewa Unit");
            System.out.println("6. Kembalikan Unit");
            System.out.println("7. Kembali");
            System.out.print("Pilih: ");
            int pilihan = scanner.nextInt();
            scanner.nextLine();

            if (pilihan == 1) {
                System.out.println("Pilih tipe unit:");
                System.out.println("1. Excavator");
                System.out.println("2. Bulldozer");
                int tipe = scanner.nextInt();
                scanner.nextLine();

                System.out.print("Nama Unit: ");
                String nama = scanner.nextLine();
                System.out.print("Harga /Day: ");
                double harga = scanner.nextDouble();
                scanner.nextLine();

                if (tipe == 1) {
                    System.out.print("Kapasitas Bucket (m3): ");
                    double kapasitas = scanner.nextDouble();
                    scanner.nextLine();
                    daftarUnit.add(new Excavator(unitId++, nama, harga, true, kapasitas));
                } else if (tipe == 2) {
                    System.out.print("Tipe Blade: ");
                    String blade = scanner.nextLine();
                    daftarUnit.add(new Bulldozer(unitId++, nama, harga, true, blade));
                } else {
                    System.out.println("Tipe tidak valid!");
                }

            } else if (pilihan == 2) {
                if (daftarUnit.isEmpty()) {
                    System.out.println("Belum ada unit yang ditambahkan.");
                } else {
                    for (Unit unit : daftarUnit)
                        System.out.println(unit.getInfo(true));
                }

            } else if (pilihan == 3) {
                System.out.print("Masukkan ID unit yang akan diperbarui: ");
                int id = scanner.nextInt();
                scanner.nextLine();
                boolean ditemukan = false;

                for (Unit unit : daftarUnit) {
                    if (unit.getId() == id) {
                        System.out.print("Nama baru: ");
                        unit.setNama(scanner.nextLine());
                        System.out.print("Harga /Day: ");
                        unit.setHarga(scanner.nextDouble());
                        scanner.nextLine();

                        if (unit instanceof Excavator) {
                            System.out.print("Kapasitas Bucket baru (m3): ");
                            ((Excavator) unit).setKapasitas(scanner.nextDouble());
                            scanner.nextLine();
                        } else if (unit instanceof Bulldozer) {
                            System.out.print("Tipe Blade baru: ");
                            ((Bulldozer) unit).setTipeBlade(scanner.nextLine());
                        }

                        System.out.println("Data unit berhasil diperbarui!");
                        ditemukan = true;
                        break;
                    }
                }

                if (!ditemukan) System.out.println("Unit dengan ID tersebut tidak ditemukan.");

            } else if (pilihan == 4) {
                System.out.print("Masukkan ID unit yang akan dihapus: ");
                int id = scanner.nextInt();
                scanner.nextLine();

                boolean dihapus = daftarUnit.removeIf(unit -> unit.getId() == id);
                if (dihapus) {
                    System.out.println("Unit berhasil dihapus.");
                } else {
                    System.out.println("Unit tidak ditemukan.");
                }

            } else if (pilihan == 5) {
                System.out.print("Masukkan ID unit yang ingin disewa: ");
                int id = scanner.nextInt();
                scanner.nextLine();

                boolean ditemukan = false;
                for (Unit unit : daftarUnit) {
                    if (unit.getId() == id) {
                        unit.sewaUnit();
                        ditemukan = true;
                        break;
                    }
                }

                if (!ditemukan) System.out.println("Unit tidak ditemukan.");

            } else if (pilihan == 6) {
                System.out.print("Masukkan ID unit yang ingin dikembalikan: ");
                int id = scanner.nextInt();
                scanner.nextLine();

                boolean ditemukan = false;
                for (Unit unit : daftarUnit) {
                    if (unit.getId() == id) {
                        unit.kembalikanUnit();
                        ditemukan = true;
                        break;
                    }
                }

                if (!ditemukan) System.out.println("Unit tidak ditemukan.");

            } else if (pilihan == 7) {
                break;

            } else {
                System.out.println("Pilihan tidak valid!");
            }

        } catch (InputMismatchException e) {
            System.out.println("Input tidak valid! Harap masukkan angka.");
            scanner.nextLine();
        }
    }
}
}

```