```java
import java.util.*;

// Abstract Class Unit
abstract class Unit {
    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() { // final method
        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;
    }

    // Abstract method
    public abstract String getInfo();

    // Overloading
    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 + ")";
        }
    }
}

// Child Class 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";
    }
}

// Child Class 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;
    }
}

```

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

    public static void main(String[] args) {
        while (true) {
            System.out.print("\033[23");
            System.out.println("============Admin============");
            System.out.println("\nPenyewaan Alat Berat |CV.Bintang Jaya|");
            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!");
            }
        }
    }

    private static void MUnit() {
        while (true) {
            System.out.println("\nPengelolaan 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. 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();
                    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) {
                for (Unit unit : daftarUnit)
                    System.out.println(unit.getId() + ". " + unit.getNama() + " - Rp " + unit.getHarga() + " /Day - " + (unit.isTersedia() ? "Ready" : "Rented") + " | " + unit.getInfo());

            } else if (pilihan == 3) {
                System.out.print("Masukkan ID unit yang akan diperbarui: ");
                int id = scanner.nextInt();
                scanner.nextLine();
                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) {
                            Excavator ex = (Excavator) unit;
                            System.out.print("Kapasitas Bucket baru (m3): ");
                            ex.setKapasitas(scanner.nextDouble());
                            scanner.nextLine();
                        } else if (unit instanceof Bulldozer) {
                            Bulldozer bd = (Bulldozer) unit;
                            System.out.print("Tipe Blade baru: ");
                            bd.setTipeBlade(scanner.nextLine());
                        }

                        System.out.println("Data unit berhasil diperbarui!");
                        break;
                    }
                }
            } else if (pilihan == 4) {
                System.out.print("Masukkan ID unit: ");
                int id = scanner.nextInt();
                daftarUnit.removeIf(unit -> unit.getId() == id);
            } else break;
        }
    }
}

```