

Nama : Akmal Alvian Pratama

NIM : 2309106021

App.java

```
import model.Donasi;
import java.util.Scanner;
import java.util.ArrayList;

public class App {
    public static void main(String[] args) {
        ArrayList<Donasi> dataDonasi = new ArrayList<>();
        Scanner input = new Scanner(System.in);
        int pilihan;
        while (true) {
            try {
                System.out.println("== Sistem Pengelolaan Donasi Online ==");
                System.out.println("1. Tambah Donasi");
                System.out.println("2. Tampilkan Semua Donasi");
                System.out.println("3. Update Donasi");
                System.out.println("4. Hapus Donasi");
                System.out.println("5. Verifikasi Donasi");
                System.out.println("6. Ubah Kategori Donasi");
                System.out.println("7. Keluar");
                System.out.print("Pilih Menu: ");
                if (!input.hasNextInt()) {
                    throw new Exception("Inputan Harus Angka");
                }
                pilihan = input.nextInt();
                input.nextLine();
                switch (pilihan) {
                    case 1:
                        TambahDonasi(dataDonasi, input);
                        ClearScreen();
                        break;
                    case 2:
                        TampilkanDonasi(dataDonasi);
                        break;
                    case 3:
                        UpdateDonasi(dataDonasi, input);
                        ClearScreen();
                        break;
                }
            } catch (Exception e) {
                System.out.println("Terjadi kesalahan: " + e.getMessage());
            }
        }
    }
}
```

```

        case 4:
            HapusDonasi(dataDonasi, input);
            ClearScreen();
            break;
        case 5:
            VerifikasiDonasi(dataDonasi, input);
            ClearScreen();
            break;
        case 6:
            UbahKategoriDonasi(dataDonasi, input);
            ClearScreen();
            break;
        case 7:
            System.exit(0);
            break;
        default:
            System.out.println("Pilihan tidak tersedia");
            break;
    }
} catch (Exception e) {
    System.out.println("Terjadi Kesalahan: " + e.getMessage());
    input.nextLine();
}
}

public static void ClearScreen() {
System.out.print("\033[H\033[2J");
System.out.flush();
}

public static void TampilkanDonasi(ArrayList<Donasi> dataDonasi) {
    System.out.println("\n== Data Donasi ==");
    if (dataDonasi.isEmpty()) {
        System.out.println("Belum ada donasi yang tercatat.");
        return;
    }
    for (int i = 0; i < dataDonasi.size(); i++) {
        System.out.println("Donasi Ke-" + (i + 1));
        dataDonasi.get(i).TampilkanData();
        System.out.println("=====");
    }
}

```

```
public static void TambahDonasi(ArrayList<Donasi> dataDonasi, Scanner input) {
    try {
        System.out.println("\n==== Tambah Donasi ===");
        System.out.print("Nama Donatur: ");
        String namaDonatur = input.nextLine();

        double jumlahDonasi;
        while (true) {
            System.out.print("Jumlah Donasi (Rp): ");
            if (input.hasNextDouble()) {
                jumlahDonasi = input.nextDouble();
                if (jumlahDonasi <= 0) {
                    System.out.println("Jumlah donasi harus lebih dari 0!");
                    input.nextLine();
                    continue;
                }
                break;
            } else {
                System.out.println("Input harus berupa angka!");
                input.nextLine();
            }
        }

        input.nextLine(); // Consume newline
        System.out.print("Metode Pembayaran: ");
        String metodePembayaran = input.nextLine();

        Donasi donasi = new Donasi(namaDonatur, jumlahDonasi, metodePembayaran);
        dataDonasi.add(donasi);

        // Gimmick efek loading
        System.out.print("Memproses");
        for (int i = 0; i < 3; i++) {
            Thread.sleep(500);
            System.out.print(".");
        }
        System.out.println("\nDonasi Berhasil Ditambahkan!\n");

    } catch (Exception e) {
        System.out.println("Terjadi kesalahan saat menambahkan donasi: " +
e.getMessage());
    }
}
```

```
public static void UpdateDonasi(ArrayList<Donasi> dataDonasi, Scanner input) {
    try {
        System.out.println("\n==== Update Donasi ====");
        System.out.print("Masukkan Nomor Donasi: ");
        if (!input.hasNextInt()) {
            System.out.println("Input harus berupa angka.");
            input.next();
            return;
        }
        int nomor = input.nextInt();
        input.nextLine();
        if (nomor < 1 || nomor > dataDonasi.size()) {
            System.out.println("Data Tidak Ditemukan.");
        } else {
            Donasi donasi = dataDonasi.get(nomor - 1);

            System.out.print("Nama Donatur (" + donasi.getNamaDonatur() + "): ");
            String namaDonatur = input.nextLine();
            if (!namaDonatur.isEmpty()) {
                donasi.setNamaDonatur(namaDonatur);
            }

            System.out.print("Jumlah Donasi (Rp " + donasi.getJumlahDonasi() + "): ");
            String jumlahStr = input.nextLine();
            if (!jumlahStr.isEmpty()) {
                try {
                    double jumlahDonasi = Double.parseDouble(jumlahStr);
                    donasi.setJumlahDonasi(jumlahDonasi);
                } catch (NumberFormatException e) {
                    System.out.println("Jumlah donasi harus berupa angka. Nilai tidak diubah.");
                }
            }
        }

        System.out.print("Metode Pembayaran (" + donasi.getMetodePembayaran() + ")");
        String metodePembayaran = input.nextLine();
        if (!metodePembayaran.isEmpty()) {
            donasi.setMetodePembayaran(metodePembayaran);
        }

        System.out.print("Memproses");
        for (int i = 0; i < 3; i++) {
            Thread.sleep(500);
        }
    }
}
```

```
        System.out.print(".");
    }

    System.out.println("Donasi Berhasil Diupdate");
}
} catch (Exception e) {
    System.out.println("Terjadi kesalahan saat mengupdate donasi: " + e.getMessage());
}
}

public static void HapusDonasi(ArrayList<Donasi> dataDonasi, Scanner input) {
try {
    System.out.println("\n==== Hapus Donasi ====");
    System.out.print("Masukkan Nomor Donasi: ");
    if (!input.hasNextInt()) {
        System.out.println("Input harus berupa angka.");
        input.next();
        return;
    }
    int nomor = input.nextInt();
    input.nextLine();
    if (nomor < 1 || nomor > dataDonasi.size()) {
        System.out.println("Data Tidak Ditemukan.");
    } else {
        dataDonasi.remove(nomor - 1);
        System.out.print("Memproses");
        for (int i = 0; i < 3; i++) {
            Thread.sleep(500);
            System.out.print(".");
        }
        System.out.println("Donasi Berhasil Dihapus");
    }
} catch (Exception e) {
    System.out.println("Terjadi kesalahan saat menghapus donasi: " + e.getMessage());
}
}

public static void VerifikasiDonasi(ArrayList<Donasi> dataDonasi, Scanner input) {
try {
    System.out.println("\n==== Verifikasi Donasi ====");
    System.out.print("Masukkan Nomor Donasi: ");
    if (!input.hasNextInt()) {
        System.out.println("Input harus berupa angka.");
        input.next();
    }
}
```

```

        return;
    }
    int nomor = input.nextInt();
input.nextLine();
    if (nomor < 1 || nomor > dataDonasi.size()) {
        System.out.println("Data Tidak Ditemukan.");
    } else {
        Donasi donasi = dataDonasi.get(nomor - 1);
        // Mengakses properti public langsung
        donasi.statusVerifikasi = true;
        System.out.print("Memproses");
        for (int i = 0; i < 3; i++) {
            Thread.sleep(500);
            System.out.print(".");
        }
        System.out.println("Donasi Berhasil Diverifikasi");
    }
} catch (Exception e) {
    System.out.println("Terjadi kesalahan saat verifikasi donasi: " + e.getMessage());
}
}

public static void UbahKategoriDonasi(ArrayList<Donasi> dataDonasi, Scanner input) {
    try {
        System.out.println("\n==== Ubah Kategori Donasi ====");
        System.out.print("Masukkan Nomor Donasi: ");
        if (!input.hasNextInt()) {
            System.out.println("Input harus berupa angka.");
            input.next();
            return;
        }
        int nomor = input.nextInt();
input.nextLine();
        if (nomor < 1 || nomor > dataDonasi.size()) {
            System.out.println("Data Tidak Ditemukan.");
        } else {
            Donasi donasi = dataDonasi.get(nomor - 1);
            System.out.println("Kategori Saat Ini: " + donasi.getKategoriDonasi());
            System.out.println("Pilih Kategori Baru:");
            System.out.println("1. Umum");
            System.out.println("2. Pendidikan");
            System.out.println("3. Kesehatan");
            System.out.println("4. Bencana Alam");
            System.out.print("Pilihan: ");
        }
    }
}

```

```
if (!input.hasNextInt()) {
    System.out.println("Input harus berupa angka.");
    input.next();
    return;
}

int pilihan = input.nextInt();
input.nextLine();

String kategori = "Umum";
switch (pilihan) {
    case 1:
        kategori = "Umum";
        break;
    case 2:
        kategori = "Pendidikan";
        break;
    case 3:
        kategori = "Kesehatan";
        break;
    case 4:
        kategori = "Bencana Alam";
        break;
    default:
        System.out.println("Pilihan tidak valid, menggunakan kategori Umum");
        break;
}

donasi.setKategoriDonasi(kategori);
System.out.print("Memproses");
for (int i = 0; i < 3; i++) {
    Thread.sleep(500);
    System.out.print(".");
}
System.out.println("Kategori Donasi Berhasil Diubah");
}

} catch (Exception e) {
    System.out.println("Terjadi kesalahan saat mengubah kategori donasi: " +
e.getMessage());
}
}
```

## Donasi.java

```
package model;

// Parent Class
public class Donasi {
    private String namaDonatur;
    private double jumlahDonasi;
    private String metodePembayaran;
    protected String waktuDonasi;
    String kategoriDonasi;
    public boolean statusVerifikasi;

    public Donasi(String namaDonatur, double jumlahDonasi, String metodePembayaran) {
        this.namaDonatur = namaDonatur;
        this.jumlahDonasi = jumlahDonasi;
        this.metodePembayaran = metodePembayaran;
        this.waktuDonasi = java.time.LocalDateTime.now().toString();
        this.kategoriDonasi = "Umum";
        this.statusVerifikasi = false;
    }

    public String getNamaDonatur() { return namaDonatur; }
    public double getJumlahDonasi() { return jumlahDonasi; }
    public String getMetodePembayaran() { return metodePembayaran; }
    protected String getWaktuDonasi() { return waktuDonasi; }
    public String getKategoriDonasi() { return kategoriDonasi; }

    public void setNamaDonatur(String namaDonatur) { this.namaDonatur = namaDonatur; }

// Method asli
    public void setJumlahDonasi(double jumlahDonasi) {
        this.jumlahDonasi = jumlahDonasi;
    }

// ↗ Method Overloading
    public void setJumlahDonasi(String jumlah) {
        try {
            double jumlahParsed = Double.parseDouble(jumlah);
            if (jumlahParsed > 0) {
                this.jumlahDonasi = jumlahParsed;
            } else {
                System.out.println("Jumlah donasi harus lebih dari 0. Nilai tidak diubah.");
            }
        }
    }
}
```

```

        } catch (NumberFormatException e) {
            System.out.println("Format jumlah donasi tidak valid. Nilai tidak diubah.");
        }
    }

    public void setMetodePembayaran(String metodePembayaran) { this.metodePembayaran =
metodePembayaran; }
    public void setKategoriDonasi(String kategoriDonasi) { this.kategoriDonasi =
kategoriDonasi; }

    public void TampilkanData() {
        System.out.println("Nama Donatur: " + namaDonatur);
        System.out.println("Jumlah Donasi: Rp " + jumlahDonasi);
        System.out.println("Metode Pembayaran: " + metodePembayaran);
        System.out.println("Waktu Donasi: " + waktuDonasi);
        System.out.println("Kategori: " + kategoriDonasi);
        System.out.println("Status Verifikasi: " + (statusVerifikasi ? "Terverifikasi" : "Belum
Terverifikasi"));
    }
}

// Subclass Donasi Pendidikan
class DonasiPendidikan extends Donasi {
    private String institusiPenerima;

    public DonasiPendidikan(String namaDonatur, double jumlahDonasi, String
metodePembayaran, String institusiPenerima) {
        super(namaDonatur, jumlahDonasi, metodePembayaran);
        this.kategoriDonasi = "Pendidikan";
        this.institusiPenerima = institusiPenerima;
    }

    public String getInstitusiPenerima() { return institusiPenerima; }
    public void setInstitusiPenerima(String institusiPenerima) { this.institusiPenerima =
institusiPenerima; }

    @Override
    public void TampilkanData() {
        super.TampilkanData();
        System.out.println("Institusi Penerima: " + institusiPenerima);
    }
}

// Subclass Donasi Kesehatan

```

```
class DonasiKesehatan extends Donasi {  
    private String rumahSakitPenerima;  
  
    public DonasiKesehatan(String namaDonatur, double jumlahDonasi, String  
metodePembayaran, String rumahSakitPenerima) {  
        super(namaDonatur, jumlahDonasi, metodePembayaran);  
        this.kategoriDonasi = "Kesehatan";  
        this.rumahSakitPenerima = rumahSakitPenerima;  
    }  
  
    public String getRumahSakitPenerima() { return rumahSakitPenerima; }  
    public void setRumahSakitPenerima(String rumahSakitPenerima) {  
this.rumahSakitPenerima = rumahSakitPenerima; }  
  
    @Override  
    public void TampilkanData() {  
        super.TampilkanData();  
        System.out.println("Rumah Sakit Penerima: " + rumahSakitPenerima);  
    }  
}
```