

KUBUTUHAN FUNGSIONAL

Manajemen Pelanggan

- **Tambah Pelanggan**
 - Pelanggan dapat didaftarkan dengan memasukkan nama, alamat, nomor telepon, dan jumlah galon yang dipesan.
 - Pelanggan bisa memilih **jenis pelanggan** (Reguler/Premium) dengan manfaat berbeda.
- **Lihat Daftar Pelanggan**
 - Sistem menampilkan semua pelanggan yang telah terdaftar.
 - Detail pelanggan mencakup **nama, alamat, nomor telepon, jumlah galon, dan jenis pelanggan**.
- **Update Data Pelanggan**
 - Pelanggan dapat mengubah **nomor telepon** dan **jumlah galon yang dipesan**.
 - Sistem memastikan nomor yang diinput adalah valid sebelum diperbarui.
- **Hapus Pelanggan**
 - Admin dapat menghapus pelanggan dari daftar.
 - Total pelanggan yang terdaftar diperbarui setelah penghapusan dilakukan.
- **Total Pelanggan**
 - Sistem menyimpan jumlah pelanggan secara **statik** untuk referensi cepat.

CLASS DIAGRAM



TAMPILAN DAN PANDUAN PENGGUNA

Aplikasi Antar Galon:

1. Tambah Pelanggan
 2. Lihat Daftar Pelanggan
 3. Update Data Pelanggan
 4. Hapus Pelanggan
 5. Total Pelanggan
 6. Keluar
- Pilih opsi:

Aplikasi ini memungkinkan pengguna untuk mengelola data pelanggan dalam sistem pengantaran galon. Berikut adalah langkah-langkah penggunaan:

1. Membuka Aplikasi

- Jalankan aplikasi dan kamu akan melihat daftar menu utama.

2. Memilih Menu Utama

- Sistem akan menampilkan beberapa opsi: **1 Tambah Pelanggan** → Menambahkan pelanggan baru dengan informasi nama, alamat, nomor telepon, jumlah galon, dan jenis pelanggan (Reguler/Premium). **2 Lihat Daftar Pelanggan** → Menampilkan seluruh pelanggan yang terdaftar beserta detailnya. **3 Update Data Pelanggan** → Memperbarui informasi pelanggan, seperti nomor telepon dan jumlah galon. **4 Hapus Pelanggan** → Menghapus pelanggan dari daftar pelanggan. **5 Total Pelanggan** → Menampilkan jumlah pelanggan yang telah terdaftar. **6 Keluar** → Menutup aplikasi.

3. Memasukkan Data

- Untuk menambahkan pelanggan, masukkan informasi yang diminta setelah memilih **Tambah Pelanggan**.
- Untuk melihat pelanggan, pilih **Lihat Daftar Pelanggan** dan sistem akan menampilkan daftar pelanggan yang ada.

4. Mengupdate dan Menghapus Pelanggan

- Pilih **Update Data Pelanggan** untuk memperbarui nomor telepon atau jumlah galon pelanggan.
- Pilih **Hapus Pelanggan** untuk menghapus pelanggan dari daftar.

5. Menutup Aplikasi

- Jika ingin keluar dari aplikasi, pilih **Keluar**.

Aplikasi ini sangat praktis untuk manajemen pelanggan dalam sistem pengantaran galon. Jika ingin fitur tambahan seperti penyimpanan data ke **database**

```
Pilih opsi: 1
Masukkan Nama: admin
Masukkan Alamat: jl.rk
Masukkan Nomor Telepon: 12345678
Masukkan Jumlah Galon: 1
Pilih jenis pelanggan:
1. Reguler
2. Premium
Pilih opsi: 1
Pelanggan berhasil ditambahkan!
```

Panduan Menambah Pelanggan

1 Memilih Opsi Tambah Pelanggan

- Saat berada di menu utama, masukkan angka 1 lalu tekan **Enter**.

2 Menginput Data Pelanggan

- Masukkan **Nama** pelanggan → Ketik nama, lalu tekan **Enter**.
- Masukkan **Alamat** pelanggan → Ketik alamat, lalu tekan **Enter**.
- Masukkan **Nomor Telepon** pelanggan → Ketik nomor telepon, lalu tekan **Enter**.
- Masukkan **Jumlah Galon** yang dipesan → Ketik angka jumlah galon, lalu tekan **Enter**.

3 Memilih Jenis Pelanggan

- Sistem akan meminta kamu memilih jenis pelanggan:
 - **Ketik 1** untuk **Reguler** (pelanggan biasa).
 - **Ketik 2** untuk **Premium** (pelanggan dengan layanan khusus).
- Masukkan angka sesuai pilihan, lalu tekan **Enter**.

4 Konfirmasi Penambahan Pelanggan

- Setelah semua data dimasukkan, sistem akan menampilkan pesan "**Pelanggan berhasil ditambahkan!**"
- Pelanggan sekarang masuk ke daftar pelanggan dalam sistem.

★ Tips Tambahan:

- Pastikan memasukkan **nomor telepon** dengan benar agar pelanggan bisa dihubungi.

- Jika salah memasukkan data, gunakan fitur **Update Data Pelanggan** di menu utama untuk memperbaikinya.

```
Pilih opsi: 3

Pelanggan #1
Jenis: Pelanggan Reguler
Nama: admin
Alamat: jl.rk
Nomor Telepon: 12345678
Jumlah Galon: 1
Masukkan nomor pelanggan yang ingin di-update: 1
Masukkan Nomor Telepon Baru: 299
Masukkan Jumlah Galon Baru: 3
Data pelanggan berhasil diperbarui!
```

Panduan Mengupdate Data Pelanggan

1 Memilih Opsi Update Pelanggan

- Dari menu utama, masukkan angka 3 lalu tekan **Enter**.

2 Melihat Daftar Pelanggan

- Sistem akan menampilkan daftar pelanggan yang telah terdaftar beserta nomor urutnya.

3 Memilih Pelanggan yang Akan Diperbarui

- Masukkan nomor pelanggan yang ingin diperbarui sesuai dengan daftar yang ditampilkan, lalu tekan **Enter**.
- Jika nomor yang dimasukkan tidak valid, sistem akan memberikan peringatan.

4 Memasukkan Data Baru

- Masukkan **Nomor Telepon Baru** → Ketik nomor baru, lalu tekan **Enter**.
- Masukkan **Jumlah Galon Baru** → Ketik jumlah galon terbaru, lalu tekan **Enter**.

5 Konfirmasi Perubahan Data

- Sistem akan memperbarui data pelanggan dan menampilkan pesan "**Data pelanggan berhasil diperbarui!**".

TAMPILAN CODE

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

// Interface yang menentukan metode pelanggan
interface PelangganInterface {
    void tampilkanData();
}

// Abstract class untuk pelanggan
abstract class PelangganBase implements PelangganInterface {
    final String nama; // Final agar tidak bisa diubah
    final String alamat;
    int nomorTelepon;
    int jumlahGalon;
    static int totalPelanggan = 0; // Static untuk menyimpan jumlah pelanggan

    public PelangganBase(String nama, String alamat, int nomorTelepon, int
jumlahGalon) {
        this.nama = nama;
        this.alamat = alamat;
        this.nomorTelepon = nomorTelepon;
        this.jumlahGalon = jumlahGalon;
        totalPelanggan++;
    }

    // Metode abstrak yang harus diimplementasikan oleh subclass
    abstract void jenisPelanggan();

    @Override
    public void tampilkanData() {
        System.out.println("Nama: " + nama);
        System.out.println("Alamat: " + alamat);
        System.out.println("Nomor Telepon: " + nomorTelepon);
        System.out.println("Jumlah Galon: " + jumlahGalon);
    }
}

// Subclass yang menggunakan Polimorfisme
class PelangganReguler extends PelangganBase {
    public PelangganReguler(String nama, String alamat, int nomorTelepon, int
jumlahGalon) {
        super(nama, alamat, nomorTelepon, jumlahGalon);
    }
}
```

```

        @Override
        void jenisPelanggan() {
            System.out.println("Jenis: Pelanggan Reguler");
        }
    }

class PelangganPremium extends PelangganBase {
    public PelangganPremium(String nama, String alamat, int nomorTelepon, int
jumlahGalon) {
        super(nama, alamat, nomorTelepon, jumlahGalon);
    }

    @Override
    void jenisPelanggan() {
        System.out.println("Jenis: Pelanggan Premium");
    }
}

public class AntarGalonApp2 {
    static ArrayList<PelangganBase> daftarPelanggan = new ArrayList<>();
    static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        while (true) {
            System.out.println("\nAplikasi Antar Galon:");
            System.out.println("1. Tambah Pelanggan");
            System.out.println("2. Lihat Daftar Pelanggan");
            System.out.println("3. Update Data Pelanggan");
            System.out.println("4. Hapus Pelanggan");
            System.out.println("5. Total Pelanggan");
            System.out.println("6. Keluar");

            int pilihan = getValidIntegerInput("Pilih opsi: ");

            switch (pilihan) {
                case 1 -> tambahPelanggan();
                case 2 -> lihatPelanggan();
                case 3 -> updatePelanggan();
                case 4 -> hapusPelanggan();
                case 5 -> System.out.println("Total pelanggan saat ini: " +
PelangganBase.totalPelanggan);
                case 6 -> {
                    System.out.println("Keluar dari aplikasi.");
                    return;
                }
            }
        }
    }
}

```

```

        }
        default -> System.out.println("Pilihan tidak valid! Silakan coba
lagi.");
    }
}

static void tambahPelanggan() {
    System.out.print("Masukkan Nama: ");
    String nama = scanner.nextLine();
    System.out.print("Masukkan Alamat: ");
    String alamat = scanner.nextLine();

    int nomorTelepon = getValidIntegerInput("Masukkan Nomor Telepon: ");
    int jumlahGalon = getValidIntegerInput("Masukkan Jumlah Galon: ");

    int jenis;
    while (true) {
        System.out.println("Pilih jenis pelanggan:");
        System.out.println("1. Reguler");
        System.out.println("2. Premium");
        jenis = getValidIntegerInput("Pilih opsi: ");

        if (jenis == 1 || jenis == 2) {
            break; // Keluar dari loop jika input valid
        } else {
            System.out.println("Masukkan angka yang valid (1 atau 2).");
        }
    }

    PelangganBase pelanggan;

    if (jenis == 1) {
        pelanggan = new PelangganReguler(nama, alamat, nomorTelepon,
jumlahGalon);
    } else {
        pelanggan = new PelangganPremium(nama, alamat, nomorTelepon,
jumlahGalon);
    }

    daftarPelanggan.add(pelanggan);
    System.out.println("Pelanggan berhasil ditambahkan!");
}

```



```

static void lihatPelanggan() {
    if (daftarPelanggan.isEmpty()) {
        System.out.println("Belum ada pelanggan.");
    } else {
        for (int i = 0; i < daftarPelanggan.size(); i++) {
            System.out.println("\nPelanggan #" + (i + 1));
            daftarPelanggan.get(i).jenisPelanggan();
            daftarPelanggan.get(i).tampilkanData();
        }
    }
}

static void updatePelanggan() {
    lihatPelanggan();
    int indeks = getValidIntegerInput("Masukkan nomor pelanggan yang ingin
di-update: ");

    if (indeks < 1 || indeks > daftarPelanggan.size()) {
        System.out.println("Nomor pelanggan tidak valid!");
        return;
    }

    PelangganBase pelanggan = daftarPelanggan.get(indeks - 1);
    System.out.print("Masukkan Nomor Telepon Baru: ");
    pelanggan.nomorTelepon = getValidIntegerInput("");
    System.out.print("Masukkan Jumlah Galon Baru: ");
    pelanggan.jumlahGalon = getValidIntegerInput("");

    System.out.println("Data pelanggan berhasil diperbarui!");
}

static void hapusPelanggan() {
    lihatPelanggan();
    int indeks = getValidIntegerInput("Masukkan nomor pelanggan yang ingin
dihapus: ");

    if (indeks < 1 || indeks > daftarPelanggan.size()) {
        System.out.println("Nomor pelanggan tidak valid!");
        return;
    }

    daftarPelanggan.remove(indeks - 1);
    PelangganBase.totalPelanggan--; // Mengurangi jumlah pelanggan
    System.out.println("Pelanggan berhasil dihapus!");
}

```

```
static int getValidIntegerInput(String prompt) {  
    int number;  
    while (true) {  
        System.out.print(prompt);  
        if (scanner.hasNextInt()) {  
            number = scanner.nextInt();  
            scanner.nextLine();  
            return number;  
        } else {  
            System.out.println("Masukkan angka yang valid! Coba lagi.");  
            scanner.nextLine();  
        }  
    }  
}
```

