

Nama : Raihanfitri adi kalipaksi
Nim : 2309106096

Screen Shoot program

Main.java

```

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;
import model.Barang;
import model.Catatan_pembayaran;
import model.Catatan_pengiriman;
import model.Customer;
import model.Driver;

public class Main {

    public static final int MAX_BARANG = 100;
    public static final int MAX_PENGIRIMAN = 100;
    public static final int MAX PEMBAYARAN = 100;
    public static final int MAX_CUSTOMERS = 100;
    public static final int MAX_DRIVERS = 100;
    static Customer[] customers = new Customer[MAX_CUSTOMERS];
    static Driver[] drivers = new Driver[MAX_DRIVERS];
    static Barang[] daftarBarang = new Barang[MAX_BARANG];
    static Catatan_pengiriman[] daftarPengiriman = new Catatan_pengiriman[MAX_PENGIRIMAN];
    static Catatan_pembayaran[] daftarPembayaran = new Catatan_pembayaran[MAX PEMBAYARAN];
    static int customerCount = 0;
    static int driverCount = 0;
    static int barangCount = 0;
    static int pengirimanCount = 0;
    static int pembayaranCount = 0;
    static int session_user = -1;

```

Run | Debug | Run main | Debug main

```

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    while (true) {
        System.out.println("-----");
        System.out.println("| Selamat Datang Di SIGESIT |");
        System.out.println("| 1. Login |");
        System.out.println("| 2. Register |");
        System.out.println("| 3. Exit |");
        System.out.println("-----");
        System.out.print("Pilih Menu : ");

        String pilihan = scanner.nextLine();
        switch (pilihan) {
            case "1" -> login();
            case "2" -> {
                System.out.println("-----");
                System.out.println("| Register Menu |");
                System.out.println("| 1. Register Customer |");
                System.out.println("| 2. Register Driver |");
                System.out.println("| 3. Kembali |");
                System.out.println("-----");
                System.out.print("Pilih Menu : ");
                String pilihan2 = scanner.nextLine();
                switch (pilihan2) {
                    case "1" -> registerCustomer();
                    case "2" -> registerDriver();
                    case "3" -> System.out.println("Kembali ke Menu Utama");
                    default -> System.out.println("Pilihan tidak ada");
                }
            }
            case "3" -> {
                System.out.println("Sampai Jumpa!!!");
                return;
            }
            default -> System.out.println("Pilihan tidak ada");
        }
    }
}

```

```

public static void tambahCatatanPengiriman(int id_barang, int id_driver) {
    if (driverCount == 0) {
        System.out.println("Catatan pengiriman tidak dapat dibuat karena belum ada driver yang terdaftar.");
        return;
    }

    if (pengirimanCount >= MAX_PENGIRIMAN) {
        System.out.println("Maaf, kapasitas catatan pengiriman penuh!");
        return;
    }

    Catatan_pengiriman newPengiriman = new Catatan_pengiriman();
    newPengiriman.setId_pengiriman(pengirimanCount + 1);
    newPengiriman.setId_barang(id_barang);
    newPengiriman.setId_driver(id_driver);
    newPengiriman.setHarga(0);
    newPengiriman.setStatus_pengiriman("Menunggu Driver");

    SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
    String tanggalSekarang = formatter.format(new Date());
    newPengiriman.setTanggal_pengiriman(tanggalSekarang);

    daftarPengiriman[pengirimanCount] = newPengiriman;
    pengirimanCount++;

    System.out.println("Catatan pengiriman berhasil ditambahkan!");
    System.out.println("ID Pengiriman: " + newPengiriman.getId_pengiriman());
    System.out.println("ID Barang: " + newPengiriman.getId_barang());
    System.out.println("ID Driver: " + newPengiriman.getId_driver());
    System.out.println("Status: " + newPengiriman.getStatus_pengiriman());
    System.out.println("Tanggal Pengiriman: " + newPengiriman.getTanggal_pengiriman());
}

```

```

public static void kirimBarang() {
    Scanner scanner = new Scanner(System.in);

    if (driverCount == 0) {
        System.out.println("Pengiriman barang tidak dapat dilakukan karena belum ada driver yang terdaftar.");
        return;
    }

    if (barangCount >= MAX_BARANG) {
        System.out.println("Maaf, kapasitas barang penuh!");
        return;
    }

    System.out.println("=====");
    System.out.println("|          Kirim Barang          |");
    System.out.println("=====");

    System.out.print("Masukkan Nama Barang: ");
    String namaBarang = scanner.nextLine();

    System.out.print("Masukkan Alamat Asal: ");
    String alamatAsal = scanner.nextLine();

    System.out.print("Masukkan Alamat Tujuan: ");
    String alamatTujuan = scanner.nextLine();

    Barang newBarang = new Barang();
    newBarang.setId_barang(barangCount + 1);
    newBarang.setNama(namaBarang);
    newBarang.setAlamat_asal(alamatAsal);
    newBarang.setAlamat_tujuan(alamatTujuan);
    newBarang.setId_customer(session_user);

    daftarBarang[barangCount] = newBarang;
    barangCount++;

    System.out.println("Barang berhasil dikirim!");
    System.out.println("ID Barang: " + newBarang.getId_barang());
    System.out.println("Nama Barang: " + newBarang.getNama());
    System.out.println("Alamat Asal: " + newBarang.getAlamat_asal());
    System.out.println("Alamat Tujuan: " + newBarang.getAlamat_tujuan());

    tambahCatatanPengiriman(newBarang.getId_barang(), 0);
}

```

```

public static void menuCustomer() {
    Scanner scanner = new Scanner(System.in);

    while (true) {
        System.out.println("=====");
        System.out.println("|          Menu Customer          |");
        System.out.println("| 1. Kirim Barang                |");
        System.out.println("| 2. Lihat Kiriman Ongoing       |");
        System.out.println("| 3. Lihat Kiriman Selesai       |");
        System.out.println("| 4. Logout                      |");
        System.out.println("=====");
        System.out.print("Pilih Menu : ");
        String pilihan = scanner.nextLine();

        switch (pilihan) {
            case "1" -> kirimBarang();
            case "2" -> lihatKirimanOngoing();
            case "3" -> lihatKirimanSelesai();
            case "4" -> {
                System.out.println("Logout berhasil!");
                return;
            }
            default -> System.out.println("Pilihan tidak ada!");
        }
    }
}

```

```

public static void lihatKirimanOngoing() {
    Scanner scanner = new Scanner(System.in);

    System.out.println("=====");
    System.out.println("|      Pengiriman Ongoing      |");
    System.out.println("=====");

    boolean adaPengiriman = false;

    for (int i = 0; i < pengirimanCount; i++) {
        Catatan_pengiriman pengiriman = daftarPengiriman[i];
        if (!pengiriman.getStatus_pengiriman().equals("selesai")) {
            adaPengiriman = true;
            System.out.println("ID Pengiriman: " + pengiriman.getId_pengiriman());
            System.out.println("ID Barang: " + pengiriman.getId_barang());
            System.out.println("Status: " + pengiriman.getStatus_pengiriman());
            System.out.println("Tanggal Pengiriman: " + pengiriman.getTanggal_pengiriman());
            System.out.println("-----");
        }
    }

    if (!adaPengiriman) {
        System.out.println("Tidak ada pengiriman yang ongoing.");
        return;
    }

    System.out.println("1. Batalkan Pesanan");
    System.out.println("2. Kembali");
    System.out.print("Pilih Menu : ");
    String pilihan = scanner.nextLine();

    switch (pilihan) {
        case "1" -> batalkanPesanan();
        case "2" -> System.out.println("Kembali ke menu utama.");
        default -> System.out.println("Pilihan tidak ada!");
    }
}

```

```

public static void lihatKirimanSelesai() {
    System.out.println("=====");
    System.out.println("|      Pengiriman Selesai      |");
    System.out.println("=====");

    boolean adaPengiriman = false;

    for (int i = 0; i < pengirimanCount; i++) {
        Catatan_pengiriman pengiriman = daftarPengiriman[i];
        if (pengiriman.getStatus_pengiriman().equals("selesai")) {
            adaPengiriman = true;
            System.out.println("ID Pengiriman: " + pengiriman.getId_pengiriman());
            System.out.println("ID Barang: " + pengiriman.getId_barang());
            System.out.println("Status: " + pengiriman.getStatus_pengiriman());
            System.out.println("Tanggal Pengiriman: " + pengiriman.getTanggal_pengiriman());
            System.out.println("-----");
        }
    }

    if (!adaPengiriman) {
        System.out.println("Tidak ada pengiriman yang selesai.");
    }
}

public static void batalkanPesanan() {
    Scanner scanner = new Scanner(System.in);

    System.out.println("=====");
    System.out.println("|      Batalkan Pesanan      |");
    System.out.println("=====");

    System.out.print("Masukkan ID Pengiriman yang ingin dibatalkan: ");
    int id_pengiriman = scanner.nextInt();
    scanner.nextLine();

    for (int i = 0; i < pengirimanCount; i++) {
        Catatan_pengiriman pengiriman = daftarPengiriman[i];
        if (pengiriman.getId_pengiriman() == id_pengiriman && pengiriman.getStatus_pengiriman().equals("Menunggu Driver")) {
            for (int j = i; j < pengirimanCount - 1; j++) {
                daftarPengiriman[j] = daftarPengiriman[j + 1];
            }
            pengirimanCount--;
            System.out.println("Pesanan dengan ID " + id_pengiriman + " berhasil dibatalkan!");
            return;
        }
    }

    System.out.println("ID Pengiriman tidak ditemukan atau tidak bisa dibatalkan.");
}

```

```

public static void lakukanPembayaran() {
    Scanner scanner = new Scanner(System.in);

    System.out.println("=====");
    System.out.println("|        Lakukan Pembayaran        |");
    System.out.println("=====");

    System.out.print("Masukkan ID Pengiriman yang ingin dibayar: ");
    int id_pengiriman = scanner.nextInt();
    scanner.nextLine();
    System.out.println("Pilih Metode Pembayaran:");
    System.out.println("1. Cash");
    System.out.println("2. Transfer");
    System.out.print("Pilih Metode : ");
    String metode = scanner.nextLine();

    String metodePembayaran = metode.equals("1") ? "Cash" : "Transfer";
    for (int i = 0; i < pembayaranCount; i++) {
        Catatan_pembayaran pembayaran = daftarPembayaran[i];
        if (pembayaran.getId_pengiriman() == id_pengiriman) {
            pembayaran.setStatus_pembayaran("Lunas");
            pembayaran.setMetode_pembayaran(metodePembayaran);
            System.out.println("Pembayaran berhasil dilakukan!");
            System.out.println("ID Pembayaran: " + pembayaran.getId_pembayaran());
            System.out.println("Metode Pembayaran: " + pembayaran.getMetode_pembayaran());
            return;
        }
    }

    System.out.println("ID Pengiriman tidak ditemukan.");
}

```

```

public static void validasiDriver() {
    Scanner scanner = new Scanner(System.in);

    System.out.println("=====");
    System.out.println("|      Validasi Driver      |");
    System.out.println("=====");

    boolean adaDriver = false;
    for (int i = 0; i < driverCount; i++) {
        Driver driver = drivers[i];
        if (!driver.isStatus()) {
            adaDriver = true;
            System.out.println("ID Driver: " + driver.getId());
            System.out.println("Nama: " + driver.getNama());
            System.out.println("Alamat: " + driver.getAlamat());
            System.out.println("No. Telepon: " + driver.getNo_telp());
            System.out.println("Plat Kendaraan: " + driver.getPlat_kendaraan());
            System.out.println("Nama Kendaraan: " + driver.getNama_kendaraan());
            System.out.println("-----");
        }
    }

    if (!adaDriver) {
        System.out.println("Tidak ada driver yang perlu divalidasi.");
        return;
    }

    System.out.print("Masukkan ID Driver yang ingin divalidasi: ");
    int id_driver = scanner.nextInt();
    scanner.nextLine();

    for (int i = 0; i < driverCount; i++) {
        Driver driver = drivers[i];
        if (driver.getId() == id_driver) {
            driver.setStatus(true);
            System.out.println("Driver dengan ID " + id_driver + " berhasil divalidasi!");
            return;
        }
    }

    System.out.println("ID Driver tidak ditemukan!");
}

```



```

public static void registerCustomer() {
    Scanner scanner = new Scanner(System.in);

    if (customerCount >= MAX_CUSTOMERS) {
        System.out.println("Maaf, kapasitas customer penuh!");
        return;
    }

    System.out.println("=====");
    System.out.println("|      Register Customer      |");
    System.out.println("=====");

    System.out.print("Masukkan Nama: ");
    String nama = scanner.nextLine();

    System.out.print("Masukkan Alamat: ");
    String alamat = scanner.nextLine();

    System.out.print("Masukkan No. Telepon: ");
    String noTelp = scanner.nextLine();

    System.out.print("Masukkan Password: ");
    String password = scanner.nextLine();

    Customer newCustomer = new Customer();
    newCustomer.setNama(nama);
    newCustomer.setAlamat(alamat);
    newCustomer.setNo_telp(noTelp);
    newCustomer.setPassword(password);

    customers[customerCount] = newCustomer;
    customerCount++;

    System.out.println("Registrasi Customer Berhasil!");
    System.out.println("ID Customer: " + newCustomer.getId());
    System.out.println("Nama: " + newCustomer.getNama());
    System.out.println("Alamat: " + newCustomer.getAlamat());
    System.out.println("No. Telepon: " + newCustomer.getNo_telp());
}

```

```

public static void registerDriver() {
    Scanner scanner = new Scanner(System.in);

    if (driverCount >= MAX_DRIVERS) {
        System.out.println("Maaf, kapasitas driver penuh!");
        return;
    }

    System.out.println("=====");
    System.out.println("|      Register Driver      |");
    System.out.println("=====");

    System.out.print("Masukkan Nama: ");
    String nama = scanner.nextLine();

    System.out.print("Masukkan Alamat: ");
    String alamat = scanner.nextLine();

    System.out.print("Masukkan No. Telepon: ");
    String noTelp = scanner.nextLine();

    System.out.print("Masukkan Password: ");
    String password = scanner.nextLine();

    System.out.print("Masukkan Plat Kendaraan: ");
    String platKendaraan = scanner.nextLine();

    System.out.print("Masukkan Nama Kendaraan: ");
    String namaKendaraan = scanner.nextLine();

    Driver newDriver = new Driver();
    newDriver.setNama(nama);
    newDriver.setAlamat(alamat);
    newDriver.setNo_telp(noTelp);
    newDriver.setPassword(password);
    newDriver.setPlat_kendaraan(platKendaraan);
    newDriver.setNama_kendaraan(namaKendaraan);

    drivers[driverCount] = newDriver;
    driverCount++;

    System.out.println("Registrasi Driver Berhasil!");
    System.out.println("ID Driver: " + newDriver.getId());
    System.out.println("Nama: " + newDriver.getNama());
    System.out.println("Alamat: " + newDriver.getAlamat());
    System.out.println("No. Telepon: " + newDriver.getNo_telp());
    System.out.println("Plat Kendaraan: " + newDriver.getPlat_kendaraan());
    System.out.println("Nama Kendaraan: " + newDriver.getNama_kendaraan());
    System.out.println("Status: " + newDriver.isStatus());
}

```

```

public static void login() {
    Scanner scanner = new Scanner(System.in);

    System.out.println("=====");
    System.out.println("|          Login          |");
    System.out.println("|  1. Login Customer      |");
    System.out.println("|  2. Login Driver        |");
    System.out.println("|  3. Login Admin         |");
    System.out.println("|  4. Kembali             |");
    System.out.println("=====");
    System.out.print("Pilih Menu : ");

    String pilihan = scanner.nextLine();
    switch (pilihan) {
        case "1" -> loginCustomer();
        case "2" -> loginDriver();
        case "3" -> loginAdmin();
        case "4" -> System.out.println("Kembali ke Menu Utama");
        default -> System.out.println("Pilihan tidak ada");
    }
}

public static void lihatPengirimanSelesai() {
    System.out.println("=====");
    System.out.println("|  Pengiriman Selesai    |");
    System.out.println("=====");

    boolean adaPengiriman = false;
    for (int i = 0; i < pengirimanCount; i++) {
        Catatan_pengiriman pengiriman = daftarPengiriman[i];
        if (pengiriman.getStatus_pengiriman().equals("selesai")) {
            adaPengiriman = true;
            System.out.println("ID Pengiriman: " + pengiriman.getId_pengiriman());
            System.out.println("ID Barang: " + pengiriman.getId_barang());
            System.out.println("Status: " + pengiriman.getStatus_pengiriman());
            System.out.println("Tanggal Pengiriman: " + pengiriman.getTanggal_pengiriman());
            System.out.println("-----");
        }
    }

    if (!adaPengiriman) {
        System.out.println("Tidak ada pengiriman yang selesai.");
    }
}

public static void displayPengirimanInfo(Catatan_pengiriman pengiriman) {
    System.out.println("ID Pengiriman: " + pengiriman.getId_pengiriman());
    System.out.println("Status: " + pengiriman.getStatus_pengiriman());
    System.out.println("Tanggal: " + pengiriman.getTanggal_pengiriman());
}

```

```
public static void displayPengirimanInfo(Catatan_pengiriman pengiriman, boolean detailed) {  
    displayPengirimanInfo(pengiriman);  
    if (detailed) {  
        for (Barang barang : daftarBarang) {  
            if (barang != null && barang.getId_barang() == pengiriman.getId_barang()) {  
                System.out.println("Nama Barang: " + barang.getNama());  
                System.out.println("Alamat Tujuan: " + barang.getAlamat_tujuan());  
                break;  
            }  
        }  
  
        if (pengiriman.getId_driver() > 0) {  
            for (Driver driver : drivers) {  
                if (driver != null && driver.getId() == pengiriman.getId_driver()) {  
                    System.out.println("Driver: " + driver.getNama());  
                    System.out.println("Kendaraan: " + driver.getNama_kendaraan());  
                    break;  
                }  
            }  
        }  
    }  
}
```

```

public static void ubahStatusPengirimanDriver() {
    Scanner scanner = new Scanner(System.in);

    System.out.println("-----");
    System.out.println("| Ubah Status Pengiriman |");
    System.out.println("-----");

    boolean adaPengiriman = false;
    for (int i = 0; i < pengirimanCount; i++) {
        Catatan_pengiriman pengiriman = daftarPengiriman[i];
        if (pengiriman.getId_driver() == session_user && pengiriman.getStatus_pengiriman().equals("ongoing")) {
            adaPengiriman = true;
            System.out.println("ID Pengiriman: " + pengiriman.getId_pengiriman());
            System.out.println("ID Barang: " + pengiriman.getId_barang());
            System.out.println("Status: " + pengiriman.getStatus_pengiriman());
            System.out.println("Tanggal Pengiriman: " + pengiriman.getTanggal_pengiriman());
            System.out.println("-----");
        }
    }

    if (!adaPengiriman) {
        System.out.println("Tidak ada pengiriman yang bisa diubah statusnya.");
        return;
    }

    System.out.print("Masukkan ID Pengiriman yang ingin diubah statusnya: ");
    int id_pengiriman = scanner.nextInt();
    scanner.nextLine();

    System.out.println("Pilih Metode Pembayaran:");
    System.out.println("1. Cash");
    System.out.println("2. Transfer");
    System.out.print("Pilih Metode : ");
    String metode = scanner.nextLine();

    String metodePembayaran = metode.equals("1") ? "Cash" : "Transfer";

    for (int i = 0; i < pengirimanCount; i++) {
        Catatan_pengiriman pengiriman = daftarPengiriman[i];
        if (pengiriman.getId_pengiriman() == id_pengiriman && pengiriman.getId_driver() == session_user) {
            if (pengiriman.getStatus_pengiriman().equals("ongoing")) {
                pengiriman.setStatus_pengiriman("selesai");
                System.out.println("Status pengiriman berhasil diubah menjadi selesai!");

                buatCatatanPembayaran(id_pengiriman, pengiriman.getHarga(), metodePembayaran);
            } else {
                System.out.println("Pengiriman ini sudah selesai atau tidak bisa diubah.");
            }
            return;
        }
    }

    System.out.println("ID Pengiriman tidak ditemukan atau tidak bisa diubah.");
}

```

```
public static void buatCatatanPembayaran(int id_pengiriman, double harga, String metodePembayaran) {
    if (pembayaranCount >= MAX_PEMBAYARAN) {
        System.out.println("Maaf, kapasitas catatan pembayaran penuh!");
        return;
    }

    Catatan_pembayaran newPembayaran = new Catatan_pembayaran();
    newPembayaran.setId_pembayaran(pembayaranCount + 1);
    newPembayaran.setId_pengiriman(id_pengiriman);
    newPembayaran.setStatus_pembayaran("Lunas");
    newPembayaran.setMetode_pembayaran(metodePembayaran);

    SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
    String tanggalSekarang = formatter.format(new Date());
    newPembayaran.setTanggal_pembayaran(tanggalSekarang);

    daftarPembayaran[pembayaranCount] = newPembayaran;
    pembayaranCount++;

    System.out.println("Catatan pembayaran berhasil dibuat!");
    System.out.println("ID Pembayaran: " + newPembayaran.getId_pembayaran());
    System.out.println("ID Pengiriman: " + newPembayaran.getId_pengiriman());
    System.out.println("Status Pembayaran: " + newPembayaran.getStatus_pembayaran());
    System.out.println("Metode Pembayaran: " + newPembayaran.getMetode_pembayaran());
    System.out.println("Tanggal Pembayaran: " + newPembayaran.getTanggal_pembayaran());
}
```

```

public static void lihatPengirimanMenungguDriver() {
    Scanner scanner = new Scanner(System.in);

    System.out.println("=====");
    System.out.println("| Pengiriman Menunggu Driver |");
    System.out.println("=====");

    boolean adaPengiriman = false;

    for (int i = 0; i < pengirimanCount; i++) {
        Catatan_pengiriman pengiriman = daftarPengiriman[i];
        if (pengiriman.getStatus_pengiriman().equals("Menunggu Driver")) {
            adaPengiriman = true;
            System.out.println("ID Pengiriman: " + pengiriman.getId_pengiriman());
            System.out.println("ID Barang: " + pengiriman.getId_barang());
            System.out.println("Status: " + pengiriman.getStatus_pengiriman());
            System.out.println("Tanggal Pengiriman: " + pengiriman.getTanggal_pengiriman());
            System.out.println("-----");
        }
    }

    if (!adaPengiriman) {
        System.out.println("Tidak ada pengiriman yang menunggu driver.");
        return;
    }

    System.out.println("1. Ubah Status Pesanan");
    System.out.println("2. Kembali");
    System.out.print("Pilih Menu : ");
    String pilihan = scanner.nextLine();

    if (pilihan.equals("1")) {
        System.out.print("Masukkan ID Pengiriman yang ingin diubah: ");
        int id_pengiriman = scanner.nextInt();
        scanner.nextLine();

        lihatDaftarDriver();
        System.out.print("Masukkan ID Driver yang ingin ditugaskan: ");
        int id_driver = scanner.nextInt();
        scanner.nextLine();

        System.out.print("Masukkan Harga Pengiriman: ");
        double harga = scanner.nextDouble();
        scanner.nextLine();

        ubahStatusPengiriman(id_pengiriman, id_driver, harga);
        buatCatatanPembayaran(id_pengiriman, harga);
    }
}

```

```

public static void ubahStatusPengiriman(int id_pengiriman, int id_driver, double harga) {
    for (int i = 0; i < pengirimanCount; i++) {
        Catatan_pengiriman pengiriman = daftarPengiriman[i];
        if (pengiriman.getId_pengiriman() == id_pengiriman) {
            pengiriman.setStatus_pengiriman("ongoing");
            pengiriman.setId_driver(id_driver);
            pengiriman.setHarga(harga);
            System.out.println("Status pengiriman berhasil diubah menjadi ongoing!");
            System.out.println("Driver dengan ID " + id_driver + " telah ditugaskan.");
            System.out.println("Harga Pengiriman: " + harga);
            return;
        }
    }
    System.out.println("ID Pengiriman tidak ditemukan!");
}

public static void buatCatatanPembayaran(int id_pengiriman, double harga) {
    if (pembayaranCount >= MAX_PEMBAYARAN) {
        System.out.println("Maaf, kapasitas catatan pembayaran penuh!");
        return;
    }

    Catatan_pembayaran newPembayaran = new Catatan_pembayaran();
    newPembayaran.setId_pembayaran(pembayaranCount + 1);
    newPembayaran.setId_pengiriman(id_pengiriman);
    newPembayaran.setStatus_pembayaran("ongoing");
    newPembayaran.setMetode_pembayaran("Menunggu Konfirmasi");

    SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
    String tanggalSekarang = formatter.format(new Date());
    newPembayaran.setTanggal_pembayaran(tanggalSekarang);

    daftarPembayaran[pembayaranCount] = newPembayaran;
    pembayaranCount++;

    System.out.println("Catatan pembayaran berhasil dibuat!");
    System.out.println("ID Pembayaran: " + newPembayaran.getId_pembayaran());
    System.out.println("ID Pengiriman: " + newPembayaran.getId_pengiriman());
    System.out.println("Status Pembayaran: " + newPembayaran.getStatus_pembayaran());
    System.out.println("Metode Pembayaran: " + newPembayaran.getMetode_pembayaran());
    System.out.println("Tanggal Pembayaran: " + newPembayaran.getTanggal_pembayaran());
}

```



```

public static void menuAdmin() {
    Scanner scanner = new Scanner(System.in);

    while (true) {
        System.out.println("-----");
        System.out.println("|           Menu Admin           |");
        System.out.println("| 1. Lihat Pengiriman Menunggu Driver |");
        System.out.println("| 2. Lihat Pengiriman Selesai |");
        System.out.println("| 3. Hapus Pesanan |");
        System.out.println("| 4. Validasi Driver |");
        System.out.println("| 5. Lihat Daftar Driver |");
        System.out.println("| 6. Logout |");
        System.out.println("-----");
        System.out.print("Pilih Menu : ");
        String pilihan = scanner.nextLine();

        switch (pilihan) {
            case "1" -> lihatPengirimanMenungguDriver();
            case "2" -> lihatPengirimanSelesai();
            case "3" -> hapusPesanan();
            case "4" -> validasiDriver();
            case "5" -> lihatDaftarDriver();
            case "6" -> {
                System.out.println("Logout berhasil!");
                return;
            }
            default -> System.out.println("Pilihan tidak ada!");
        }
    }
}

public static void lihatDaftarDriver() {
    System.out.println("-----");
    System.out.println("|       Daftar Driver       |");
    System.out.println("-----");

    if (driverCount == 0) {
        System.out.println("Tidak ada driver yang terdaftar.");
        return;
    }

    for (int i = 0; i < driverCount; i++) {
        Driver driver = drivers[i];
        System.out.println("ID Driver: " + driver.getId());
        System.out.println("Nama: " + driver.getNama());
        System.out.println("Alamat: " + driver.getAlamat());
        System.out.println("No. Telepon: " + driver.getNo_telp());
        System.out.println("Plat Kendaraan: " + driver.getPlat_kendaraan());
        System.out.println("Nama Kendaraan: " + driver.getNama_kendaraan());
        System.out.println("Status: " + (driver.isStatus() ? "Aktif" : "Menunggu Validasi"));
        System.out.println("-----");
    }
}

```

```

public static void hapusPesanan() {
    Scanner scanner = new Scanner(System.in);

    System.out.println("=====");
    System.out.println("|          Hapus Pesanan          |");
    System.out.println("=====");

    System.out.print("Masukkan ID Pengiriman yang ingin dihapus: ");
    int id_pengiriman = scanner.nextInt();
    scanner.nextLine();

    for (int i = 0; i < pengirimanCount; i++) {
        if (daftarPengiriman[i].getId_pengiriman() == id_pengiriman) {
            for (int j = i; j < pengirimanCount - 1; j++) {
                daftarPengiriman[j] = daftarPengiriman[j + 1];
            }
            pengirimanCount--;
            System.out.println("Pesanan dengan ID " + id_pengiriman + " berhasil dihapus!");
            return;
        }
    }

    System.out.println("ID Pengiriman tidak ditemukan!");
}

```

```

public static void loginAdmin() {
    Scanner scanner = new Scanner(System.in);

    System.out.println("=====");
    System.out.println("|          Login Admin          |");
    System.out.println("=====");

    System.out.print("Masukkan Username: ");
    String username = scanner.nextLine();

    System.out.print("Masukkan Password: ");
    String password = scanner.nextLine();

    if (username.equals("admin") && password.equals("admin")) {
        System.out.println("Login Admin Berhasil!");
        menuAdmin();
    } else {
        System.out.println("Username atau Password salah!");
    }
}

public static void loginCustomer() {
    Scanner scanner = new Scanner(System.in);

    System.out.println("=====");
    System.out.println("|          Login Customer        |");
    System.out.println("=====");

    System.out.print("Masukkan Nama: ");
    String nama = scanner.nextLine();

    System.out.print("Masukkan Password: ");
    String password = scanner.nextLine();

    for (int i = 0; i < customerCount; i++) {
        Customer customer = customers[i];
        if (customer.getNama().equals(nama) && customer.getPassword().equals(password)) {
            System.out.println("Login Berhasil!");
            System.out.println("Selamat Datang, " + customer.getNama() + "!");
            session_user = customer.getId();
            menuCustomer();
            return;
        }
    }

    System.out.println("Nama atau Password salah!");
}

```

```

public static void loginDriver() {
    Scanner scanner = new Scanner(System.in);

    System.out.println("-----");
    System.out.println("|          Login Driver          |");
    System.out.println("-----");

    System.out.print("Masukkan Nama: ");
    String nama = scanner.nextLine();

    System.out.print("Masukkan Password: ");
    String password = scanner.nextLine();

    for (int i = 0; i < driverCount; i++) {
        Driver driver = drivers[i];
        if (driver.getNama().equals(nama) && driver.getPassword().equals(password)) {
            if (!driver.isStatus()) {
                System.out.println("Akun Anda masih menunggu konfirmasi.");
            } else {
                System.out.println("Login Berhasil!");
                System.out.println("Selamat Datang, " + driver.getNama() + "!");
                session_user = driver.getId();
                menuDriver();
            }
            return;
        }
    }

    System.out.println("Nama atau Password salah!");
}

public static void menuDriver() {
    Scanner scanner = new Scanner(System.in);

    while (true) {
        System.out.println("-----");
        System.out.println("|          Menu Driver          |");
        System.out.println("| 1. Lihat Barang Dikirim      |");
        System.out.println("| 2. Ubah Status Pengiriman    |");
        System.out.println("| 3. Logout                    |");
        System.out.println("-----");
        System.out.print("Pilih Menu : ");
        String pilihan = scanner.nextLine();

        switch (pilihan) {
            case "1" -> lihatBarangDikirim();
            case "2" -> ubahStatusPengirimanDriver();
            case "3" -> {
                System.out.println("Logout berhasil!");
                return;
            }
            default -> System.out.println("Pilihan tidak ada!");
        }
    }
}

```

```

public static void lihatBarangDikirim() {
    System.out.println("=====");
    System.out.println("|      Barang Dikirim      |");
    System.out.println("=====");

    boolean adaPengiriman = false;

    for (int i = 0; i < pengirimanCount; i++) {
        Catatan_pengiriman pengiriman = daftarPengiriman[i];
        if (pengiriman.getId_driver() == session_user) {
            adaPengiriman = true;
            displayPengirimanInfo(pengiriman, true);
            System.out.println("-----");
        }
    }

    if (!adaPengiriman) {
        System.out.println("Tidak ada barang yang dikirim oleh Anda.");
    }
}

public static void ubahStatusPengiriman(int id_pengiriman, int id_driver) {
    for (int i = 0; i < pengirimanCount; i++) {
        Catatan_pengiriman pengiriman = daftarPengiriman[i];
        if (pengiriman.getId_pengiriman() == id_pengiriman) {
            pengiriman.setStatus_pengiriman("ongoing");
            pengiriman.setId_driver(id_driver);
            System.out.println("Status pengiriman berhasil diubah menjadi ongoing!");
            System.out.println("Driver dengan ID " + id_driver + " telah ditugaskan.");
            return;
        }
    }
    System.out.println("ID Pengiriman tidak ditemukan!");
}
}

```

User.java

```

package model;

public class User {
    protected int id;
    protected String nama;
    protected String alamat;
    protected String no_telp;
    protected String password;

    public User() {

    }

    public void displayInfo() {
        System.out.println("ID: " + id);
        System.out.println("Nama: " + nama);
        System.out.println("Alamat: " + alamat);
        System.out.println("No. Telepon: " + no_telp);
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getNama() {
        return nama;
    }

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

    public String getAlamat() {
        return alamat;
    }

    public void setAlamat(String alamat) {
        this.alamat = alamat;
    }

    public String getNo_telp() {
        return no_telp;
    }

    public void setNo_telp(String no_telp) {
        this.no_telp = no_telp;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}

```

Driver.java

```

package model;

public class Driver extends User {
    private String plat_kendaraan;
    private String nama_kendaraan;
    private boolean status;
    private static int lastId = 0;

    public Driver() {
        super();
        this.id = ++lastId;
        this.status = false;
    }

    @Override
    public void displayInfo() {
        System.out.println("--- Informasi Driver ---");
        super.displayInfo();
        System.out.println("Plat Kendaraan: " + plat_kendaraan);
        System.out.println("Nama Kendaraan: " + nama_kendaraan);
        System.out.println("Status: " + (status ? "Aktif" : "Menunggu Validasi"));
    }

    public int getId_driver() {
        return this.id;
    }

    public String getPlat_kendaraan() {
        return plat_kendaraan;
    }

    public void setPlat_kendaraan(String plat_kendaraan) {
        this.plat_kendaraan = plat_kendaraan;
    }

    public String getNama_kendaraan() {
        return nama_kendaraan;
    }

    public void setNama_kendaraan(String nama_kendaraan) {
        this.nama_kendaraan = nama_kendaraan;
    }

    public boolean isStatus() {
        return status;
    }

    public void setStatus(boolean status) {
        this.status = status;
    }

    @Override
    public String toString() {
        return "Driver{" +
            "id=" + id +
            ", nama=" + nama + '\'' +
            ", no_telp=" + no_telp + '\'' +
            ", password=" + password + '\'' +
            ", alamat=" + alamat + '\'' +
            ", plat_kendaraan=" + plat_kendaraan + '\'' +
            ", nama_kendaraan=" + nama_kendaraan + '\'' +
            ", status=" + status +
            '}';
    }
}

```

Customer.java

```

model > Customer.java > Language Support for Java(TM) by Red Hat > Customer
package model;

public class Customer extends User {
    private static int lastId = 0;

    public Customer() {
        super();
        this.id = ++lastId;
    }

    @Override
    public void displayInfo() {
        System.out.println("--- Informasi Customer ---");
        super.displayInfo();
        System.out.println("Role: Customer");
    }

    public int getId_customer() {
        return this.id;
    }

    @Override
    public String toString() {
        return "Customer(" +
            "id=" + id +
            ", nama=" + nama + '\'' +
            ", alamat=" + alamat + '\'' +
            ", no_telp=" + no_telp + '\'' +
            ", password=" + password + '\'' +
            ')';
    }
}

```

Catatan_pengiriman.java

```

model > Catatan_pengiriman.java > Language Support for Java(TM) by Red Hat > Catatan_pengi
public class Catatan_pengiriman {

    public String getTanggal_pengiriman() {
        return tanggal_pengiriman;
    }

    public void setTanggal_pengiriman(String tanggal_pengiriman) {
        this.tanggal_pengiriman = tanggal_pengiriman;
    }

    public double getHarga() {
        return harga;
    }

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

```

Catatan_pembayaran.java


```

package model;

public class Catatan_pembayaran {
    int id_pembayaran;
    int id_pengiriman;
    String status_pembayaran;
    String tanggal_pembayaran;
    String metode_pembayaran;

    public int getId_pembayaran() {
        return id_pembayaran;
    }

    public void setId_pembayaran(int id_pembayaran) {
        this.id_pembayaran = id_pembayaran;
    }

    public int getId_pengiriman() {
        return id_pengiriman;
    }

    public void setId_pengiriman(int id_pengiriman) {
        this.id_pengiriman = id_pengiriman;
    }

    public String getStatus_pembayaran() {
        return status_pembayaran;
    }

    public void setStatus_pembayaran(String status_pembayaran) {
        this.status_pembayaran = status_pembayaran;
    }

    public String getTanggal_pembayaran() {
        return tanggal_pembayaran;
    }

    public void setTanggal_pembayaran(String tanggal_pembayaran) {
        this.tanggal_pembayaran = tanggal_pembayaran;
    }

    public String getMetode_pembayaran() {
        return metode_pembayaran;
    }

    public void setMetode_pembayaran(String metode_pembayaran) {
        this.metode_pembayaran = metode_pembayaran;
    }
}

```

Barang.java

model / Barang.java / Language Support for Java(TM) by Red Hat / Barang

```
package model;

public class Barang {
    private int id_barang;
    private String nama;
    private int id_customer;
    private String alamat_asal;
    private String alamat_tujuan;

    public int getId_barang() {
        return id_barang;
    }

    public void setId_barang(int id_barang) {
        this.id_barang = id_barang;
    }

    public String getNama() {
        return nama;
    }

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

    public int getId_customer() {
        return id_customer;
    }

    public void setId_customer(int id_customer) {
        this.id_customer = id_customer;
    }

    public String getAlamat_asal() {
        return alamat_asal;
    }

    public void setAlamat_asal(String alamat_asal) {
        this.alamat_asal = alamat_asal;
    }

    public String getAlamat_tujuan() {
        return alamat_tujuan;
    }

    public void setAlamat_tujuan(String alamat_tujuan) {
        this.alamat_tujuan = alamat_tujuan;
    }
}
```

1