Algorithm and Programming

CAR SHOWROOM MANAGEMENT SYSTEM

with Java

Kelompok 1

- Batara Fadillah Putra (63785)
- Evan Hendraloka (67469)
- Kenny Budiarso Lawson (81065)
- Valen Claudia Chuardi (71430)

Pembahasan

- Penjelasan Sistem
- Penjelasan Kode Sistem
- Flowchart Sistem
- Demo Sistem

Algorithm and Programming

Penjelasan Car Showroom Management System

PENJELASAN

Aplikasi manajemen showroom mobil ini dirancang untuk mengelola data mobil, transaksi penjualan, dan informasi sales. Admin dapat menambah, mengedit, menghapus data, mencatat transaksi, melihat histori penjualan, dan menghitung omzet. Dengan antarmuka user-friendly, navigasi menjadi mudah, serta fitur undo dan redo memudahkan pengelolaan data. Sistem ini bertujuan meningkatkan efisiensi pengelolaan showroom, mendukung pengambilan keputusan, dan memperbaiki layanan pelanggan.

Flowchart

Let's begin.

FLOWCHART CAR SHOWROOM

Penjelasan Code

MAIN

```java
public class Main {
    static Scanner sc = new Scanner(System.in);
    static Admin admin = new Admin("admin", "12345");
    static Mobil[] daftarMobil = new Mobil[100];
    static Sales[] daftarSales = new Sales[100];
    static int jumlahMobil = 0;
    static int jumlahSales = 0;
    static boolean login = false;
    static int pin = 999;
    static Transaksi transaksiQueue = new Transaksi();
    static int transaksiCounter = 1;
    static int mobilCounter = 7;
    static Stack<Mobil[]> undoStackMobil = new Stack<>();
    static Stack<Mobil[]> redoStackMobil = new Stack<>();
    static Stack<Sales[]> undoStackSales = new Stack<>();
    static Stack<Sales[]> redoStackSales = new Stack<>();
```

- Scanner sc: Objek ini digunakan untuk membaca input dari pengguna.
- Admin admin: Objek ini menyimpan informasi admin dengan username dan PIN.
- Mobil[] daftarMobil: Array yang menyimpan daftar objek Mobil.
- Sales[] daftarSales: Array yang menyimpan daftar objek Sales.
- int jumlahMobil: Variabel yang menyimpan jumlah mobil yang ada di daftarMobil.
- int jumlahSales: Variabel yang menyimpan jumlah sales yang ada di daftarSales.
- Transaksi transaksiQueue: Objek ini menyimpan daftar transaksi.
- int transaksiCounter: Variabel yang digunakan untuk menghitung jumlah transaksi.
- Stack<Mobil[]> undoStackMobil: Stack yang menyimpan status daftarMobil sebelum perubahan untuk fitur undo.
- Stack<Mobil[]> redoStackMobil: Stack yang menyimpan status daftarMobil setelah undo untuk fitur redo.
- Stack<Sales[]> undoStackSales: Stack yang menyimpan status daftarSales sebelum perubahan untuk fitur undo.
- Stack<Sales[]> redoStackSales: Stack yang menyimpan status daftarSales setelah undo untuk fitur redo.

```java
public static void manajemenMobil() {
    int pilihanAktivitas;
    do {
        System.out.println(x:"\n\t=== Manajemen Mobil ===\t");
        System.out.println(x:"1. Daftar Mobil");
        System.out.println(x:"2. Tambah Mobil");
        System.out.println(x:"3. Edit Info Mobil");
        System.out.println(x:"4. Hapus Mobil");
        System.out.println(x:"5. Urutkan Mobil");
        System.out.println(x:"0. Kembali ke Menu Utama");
        System.out.print(s:"Pilih aktivitas: ");
        pilihanAktivitas = sc.nextInt();

        switch (pilihanAktivitas) {
            case 1:
                lihatMobil();
                break;
            case 2:
                tambahMobil();
                break;
            case 3:
                editMobil();
                break;
            case 4:
                hapusMobil();
                break;
            case 5:
                urutkanMobil();
                break;
            case 0:
                System.out.println(x:"Kembali ke Menu Utama.");
                return;
            default:
                System.out.println(x:"Pilihan tidak valid.");
                break;
        }
    } while (pilihanAktivitas != 5);
}
```

- manajemenMobil(): Fungsi yang menampilkan menu manajemen mobil dan memanggil fungsi terkait berdasarkan pilihan pengguna.
- tambahMobil(): Fungsi untuk menambahkan mobil baru ke daftarMobil. Menyimpan status sebelum perubahan ke undoStackMobil.
- editMobil(): Fungsi untuk mengedit informasi mobil yang ada di daftarMobil. Menyimpan status sebelum perubahan ke undoStackMobil.
- hapusMobil(): Fungsi untuk menghapus mobil dari daftarMobil. Menyimpan status sebelum perubahan ke undoStackMobil.
- lihatMobil(): Fungsi untuk menampilkan daftar mobil yang ada di daftarMobil.

```java
public static void manajemenTransaksi() {
    int pilihan;
    do {
        System.out.println(x:"\nMenu Manajemen Transaksi:");
        System.out.println(x:"1. Transaksi Baru");
        System.out.println(x:"2. Histori Transaksi");
        System.out.println(x:"3. Edit Histori Transaksi");
        System.out.println(x:"4. Hapus Histori Transaksi");
        System.out.println(x:"5. Omzet Penjualan");
        System.out.println(x:"0. Kembali ke Menu Utama");
        System.out.print(s:"Pilih: ");
        pilihan = sc.nextInt();

        switch (pilihan) {
            case 1:
                tambahTransaksi();
                break;
            case 2:
                lihatTransaksi();
                break;
            case 3:
                editTransaksi();
                break;
            case 4:
                hapusTransaksi();
                break;
            case 5:
                omzetPenjualan();
                break;
            case 0:
                System.out.println(x:"Kembali ke Menu Utama.");
                return;
            default:
                System.out.println(x:"Pilihan tidak valid.");
                break;
        }
    } while (pilihan != 6);
}
```

- manajemenTransaksi(): Fungsi yang menampilkan menu manajemen transaksi dan memanggil fungsi terkait berdasarkan pilihan pengguna.
- tambahTransaksi(): Fungsi untuk menambahkan transaksi baru ke transaksiQueue.
- lihatTransaksi(): Fungsi untuk menampilkan daftar transaksi yang ada di transaksiQueue.
- editTransaksi(): Fungsi untuk mengedit informasi transaksi yang ada di transaksiQueue.
- hapusTransaksi(): Fungsi untuk menghapus transaksi dari transaksiQueue.
- omzetPenjualan(): Fungsi untuk menampilkan total omzet penjualan dari transaksi yang ada di transaksiQueue.

```java
public static void manajemenSales() {
    int pilihanAktivitas;
    do {
        System.out.println(x:"\n\t=== Manajemen Sales ===\t");
        System.out.println(x:"1. Daftar Sales");
        System.out.println(x:"2. Tambah Sales Baru");
        System.out.println(x:"3. Edit Info Sales");
        System.out.println(x:"4. Hapus Sales");
        System.out.println(x:"5. Performa Sales");
        System.out.println(x:"0. Kembali ke Menu Utama");
        System.out.print(s:"Pilih aktivitas: ");
        pilihanAktivitas = sc.nextInt();

        switch (pilihanAktivitas) {
            case 1:
                lihatSales();
                break;
            case 2:
                tambahSales();
                break;
            case 3:
                editSales();
                break;
            case 4:
                hapusSales();
                break;
            case 5:
                performaSales();
                break;
            case 0:
                System.out.println(x:"Kembali ke Menu Utama.");
                break;
            default:
                System.out.println(x:"Pilihan tidak valid.");
                break;
        }
    } while (pilihanAktivitas != 0);
}
```

- manajemenSales(): Fungsi yang menampilkan menu manajemen sales dan memanggil fungsi terkait berdasarkan pilihan pengguna.
- tambahSales(): Fungsi untuk menambahkan sales baru ke daftarSales. Menyimpan status sebelum perubahan ke undoStackSales.
- editSales(): Fungsi untuk mengedit informasi sales yang ada di daftarSales. Menyimpan status sebelum perubahan ke undoStackSales.
- hapusSales(): Fungsi untuk menghapus sales dari daftarSales. Menyimpan status sebelum perubahan ke undoStackSales.
- lihatSales(): Fungsi untuk menampilkan daftar sales yang ada di daftarSales.
- performaSales(): Fungsi untuk menampilkan performa sales berdasarkan komisi dari transaksi penjualan mobil.

```
    public static void undo() {
        if (!undoStackMobil.isEmpty()) {
            redoStackMobil.push(daftarMobil.clone());
            daftarMobil = undoStackMobil.pop();
            System.out.println(x:"Undo berhasil dilakukan pada data mobil.");
        } else if (!undoStackSales.isEmpty()) {
            redoStackSales.push(daftarSales.clone());
            daftarSales = undoStackSales.pop();
            System.out.println(x:"Undo berhasil dilakukan pada data sales.");
        } else {
            System.out.println(x:"Tidak ada aksi yang bisa di-undo.");
        }
    }

    public static void redo() {
        if (!redoStackMobil.isEmpty()) {
            undoStackMobil.push(daftarMobil.clone());
            daftarMobil = redoStackMobil.pop();
            System.out.println(x:"Redo berhasil dilakukan pada data mobil.");
        } else if (!redoStackSales.isEmpty()) {
            undoStackSales.push(daftarSales.clone());
            daftarSales = redoStackSales.pop();
            System.out.println(x:"Redo berhasil dilakukan pada data sales.");
        } else {
            System.out.println(x:"Tidak ada aksi yang bisa di-redo.");
        }
    }
}
```

- undo(): Fungsi untuk mengembalikan status daftarMobil atau daftarSales ke status sebelum perubahan terakhir. Menyimpan status saat ini ke redoStack.
- redo(): Fungsi untuk mengembalikan status daftarMobil atau daftarSales ke status setelah undo terakhir. Menyimpan status saat ini ke undoStack.

Penjelasan Tambahan

- **Undo/Redo: Fitur ini memungkinkan pengguna untuk membatalkan (undo) atau mengulangi (redo) perubahan yang telah dilakukan pada data mobil atau sales. Setiap kali ada perubahan pada data mobil atau sales, status sebelum perubahan disimpan ke undoStack. Jika pengguna melakukan undo, status saat ini disimpan ke redoStack dan status sebelumnya diambil dari undoStack. Jika pengguna melakukan redo, status saat ini disimpan ke undoStack dan status setelah undo diambil dari redoStack.**
- **Transaksi: Objek Transaksi menyimpan daftar transaksi penjualan mobil. Setiap transaksi mencatat informasi seperti mobil yang dijual, sales yang melakukan penjualan, dan metode pembayaran.**

ADMIN

```java
public class Admin {
    private String username;
    private String pin;

    public Admin(String username, String pin) {
        this.username = username;
        this.pin = pin;
    }

    public String getUsername() {
        return username;
    }

    public String getPin() {
        return pin;
    }
}
```

- String username: Variabel ini menyimpan username dari admin.
- String pin: Variabel ini menyimpan PIN dari admin.
- Konstruktor
- Admin(String username, String pin): Konstruktor ini digunakan untuk membuat objek Admin baru dengan username dan PIN yang diberikan sebagai parameter.
- Parameter username: Username yang akan disimpan dalam objek Admin.
- Parameter pin: PIN yang akan disimpan dalam objek Admin.
- Fungsi
- getUsername(): Fungsi ini mengembalikan nilai dari variabel username.
- Return: Mengembalikan username dari admin.
- getPin(): Fungsi ini mengembalikan nilai dari variabel pin.
- Return: Mengembalikan PIN dari admin.

MOBIL

```java
public class Mobil {
    String id;
    String merk;
    String nama;
    int kapasitas;
    int tahun;
    String tipe;
    int hargaCash;
    int hargaKredit;

    public Mobil(String id, String merk, String nama, int kapasitas, int tahun, String tipe, int hargaCash, int hargaKredit) {
        this.id = id;
        this.merk = merk;
        this.nama = nama;
        this.kapasitas = kapasitas;
        this.tahun = tahun;
        this.tipe = tipe;
        this.hargaCash = hargaCash;
        this.hargaKredit = hargaKredit;
    }
```

- String id: Variabel ini menyimpan ID unik untuk setiap mobil.
- String merk: Variabel ini menyimpan merk mobil.
- String nama: Variabel ini menyimpan nama mobil.
- int kapasitas: Variabel ini menyimpan kapasitas mesin mobil dalam cc.
- int tahun: Variabel ini menyimpan tahun produksi mobil.
- String tipe: Variabel ini menyimpan tipe mobil.
- int hargaCash: Variabel ini menyimpan harga mobil jika dibeli secara tunai.
- int hargaKredit: Variabel ini menyimpan harga mobil jika dibeli secara kredit.

```java
public Mobil(String id, String merk, String nama, int kapasitas, int tahun, String tipe, int hargaCash, int hargaKredit) {
    this.id = id;
    this.merk = merk;
    this.nama = nama;
    this.kapasitas = kapasitas;
    this.tahun = tahun;
    this.tipe = tipe;
    this.hargaCash = hargaCash;
    this.hargaKredit = hargaKredit;
}
```

- Mobil(String id, String merk, String nama, int kapasitas, int tahun, String tipe, int hargaCash, int hargaKredit): digunakan untuk membuat objek Mobil baru dengan semua atribut yang diberikan sebagai parameter.
- Parameter id: ID unik untuk mobil.
- Parameter merk: Merk mobil.
- Parameter nama: Nama mobil.
- Parameter kapasitas: Kapasitas mesin mobil dalam cc.
- Parameter tahun: Tahun produksi mobil.
- Parameter tipe: Tipe mobil.
- Parameter hargaCash: Harga mobil jika dibeli secara tunai.
- Parameter hargaKredit: Harga mobil jika dibeli secara kredit.

```java
public String getId() {
    return id;
}

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

public String getMerk() {
    return merk;
}

public void setMerk(String merk) {
        this.merk = merk;
    }

public String getNama() {
    return nama;
}

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

public int getKapasitas() {
    return kapasitas;
}

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

```java
public int getTahun() {
    return tahun;
}

public void setTahun(int tahun) {
    this.tahun = tahun;
}

public String getTipe() {
    return tipe;
}

public void setTipe(String tipe) {
    this.tipe = tipe;
}

public int getHargaCash() {
    return hargaCash;
}

public int getHargaKredit() {
    return hargaKredit;
}

public void setHargaCash(int hargaCash) {
    this.hargaCash = hargaCash;
}

public void setHargaKredit(int hargaKredit) {
    this.hargaKredit = hargaKredit;
}
```

- getId(): Mengembalikan nilai dari variabel id.
- getMerk(): Mengembalikan nilai dari variabel merk.
- getNama(): Mengembalikan nilai dari variabel nama.
- getKapasitas(): Mengembalikan nilai dari variabel kapasitas.
- getTahun(): Mengembalikan nilai dari variabel tahun.
- getTipe(): Mengembalikan nilai dari variabel tipe.
- getHargaCash(): Mengembalikan nilai dari variabel hargaCash.
- getHargaKredit(): Mengembalikan nilai dari variabel hargaKredit.
- Fungsi Setter
- setMerk(String merk): Mengatur nilai dari variabel merk.
- Parameter merk: Merk baru untuk mobil.
- setNama(String nama): Mengatur nilai dari variabel nama.
- Parameter nama: Nama baru untuk mobil.
- setKapasitas(int kapasitas): Mengatur nilai dari variabel kapasitas.
- Parameter kapasitas: Kapasitas mesin baru untuk mobil dalam cc.
- setTahun(int tahun): Mengatur nilai dari variabel tahun.
- Parameter tahun: Tahun produksi baru untuk mobil.
- setTipe(String tipe): Mengatur nilai dari variabel tipe.
- Parameter tipe: Tipe baru untuk mobil.
- setHargaCash(int hargaCash): Mengatur nilai dari variabel hargaCash.
- Parameter hargaCash: Harga tunai baru untuk mobil.
- setHargaKredit(int hargaKredit): Mengatur nilai dari variabel hargaKredit.

```
public String toString() {
    return "ID: " + id + ", Merk: " + merk + ", Nama: " + nama +
    ", Kapasitas Mesin: " + kapasitas + " cc" + ", Tahun Produksi: " + tahun +
    ", Tipe: " + tipe + ", Harga Cash: Rp " + hargaCash + ", Harga Kredit: Rp " + hargaKredit;
}
```

- getId(): Mengembalikan nilai dari variabel id.
- getMerk(): Mengembalikan nilai dari variabel merk.
- getNama(): Mengembalikan nilai dari variabel nama.
- getKapasitas(): Mengembalikan nilai dari variabel kapasitas.
- getTahun(): Mengembalikan nilai dari variabel tahun.
- getTipe(): Mengembalikan nilai dari variabel tipe.
- getHargaCash(): Mengembalikan nilai dari variabel hargaCash.
- getHargaKredit(): Mengembalikan nilai dari variabel hargaKredit.

SALES

```java
public class Sales {
    String id;
    String nama;

    public String getId() {
        return id;
    }

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

    public String getNama() {
        return nama;
    }

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

    public Sales(String id, String nama) {
        this.id = id;
        this.nama = nama;
    }

    public String toString() {
        return "ID: " + id + ", Nama: " + nama;
    }
```

- String id: Variabel ini menyimpan ID unik untuk setiap sales.
- String nama: Variabel ini menyimpan nama dari sales.

Konstruktor

- Sales(String id, String nama): Konstruktor ini digunakan untuk membuat objek Sales baru dengan ID dan nama yang diberikan sebagai parameter.
- Parameter id: ID unik untuk sales.
- Parameter nama: Nama dari sales.

Fungsi Getter

- getId(): Mengembalikan nilai dari variabel id.
- Return: Mengembalikan ID dari sales.
- getNama(): Mengembalikan nilai dari variabel nama.
- Return: Mengembalikan nama dari sales.
- Parameter nama: Nama baru untuk sales.

Fungsi toString

- toString(): Mengembalikan representasi string dari objek Sales, yang mencakup ID dan nama sales dalam format yang mudah dibaca.
- Return: String yang berisi informasi lengkap tentang sales, termasuk ID dan nama.

TRANSAKSI

```java
public class Transaksi {
    private TransaksiDetail[] transaksiQueue; // Array sebagai queue
    private int front, rear, size, capacity;

    public Transaksi() {
        this.capacity = 20; // Kapasitas default
        this.transaksiQueue = new TransaksiDetail[capacity];
        this.front = 0; // Posisi elemen pertama
        this.rear = -1; // Posisi elemen terakhir
        this.size = 0;  // Jumlah elemen dalam queue
    }

    public void tambahTransaksi(TransaksiDetail transaksi) {
        if (isFull()) {
            System.out.println("Antrian transaksi penuh. Tidak bisa menambahkan transaksi.");
            return;
        }
        rear = (rear + 1) % capacity;
        transaksiQueue[rear] = transaksi;
        size++;
        System.out.println("Transaksi berhasil ditambahkan: " + transaksi);
    }

    // Lihat semua transaksi dalam queue
    public void lihatTransaksi() {
        if (isEmpty()) {
            System.out.println("Antrian transaksi kosong.");
            return;
        }
        System.out.println("Daftar Transaksi:");
        for (int i = 0; i < size; i++) {
            int index = (front + i) % capacity;
            System.out.println(transaksiQueue[index]);
        }
    }
```

- TransaksiDetail[] transaksiQueue: Array yang menyimpan objek TransaksiDetail dalam bentuk queue (antrian).
- int front: Indeks dari elemen pertama dalam queue.
- int rear: Indeks dari elemen terakhir dalam queue.
- int size: Jumlah elemen saat ini dalam queue.
- int capacity: Kapasitas maksimum dari queue.

Fungsi

- isFull(): Fungsi ini memeriksa apakah queue sudah penuh.
- Return: Mengembalikan true jika size sama dengan capacity, yang berarti queue penuh. Mengembalikan false jika tidak.
- isEmpty(): Fungsi ini memeriksa apakah queue kosong.
- Return: Mengembalikan true jika size sama dengan 0, yang berarti queue kosong. Mengembalikan false jika tidak.

```java
// Hapus transaksi dari queue
public void hapusTransaksi() {
    if (isEmpty()) {
        System.out.println("Antrian transaksi kosong. Tidak ada transaksi yang dihapus.");
        return;
    }
    TransaksiDetail removed = transaksiQueue[front];
    front = (front + 1) % capacity;
    size--;
    System.out.println("Transaksi dihapus: " + removed);
}

// Cek apakah queue penuh
private boolean isFull() {
    return size == capacity;
}

// Cek apakah queue kosong
private boolean isEmpty() {
    return size == 0;
}

public int getSize() {
    return size;
}

public TransaksiDetail get(int index) {
    return transaksiQueue[(front + index) % capacity];
}
```

- getSize(): Fungsi ini mengembalikan jumlah elemen saat ini dalam queue.
- Return: Mengembalikan nilai dari variabel size.
- get(int index): Fungsi ini mengembalikan elemen pada posisi tertentu dalam queue.
- Parameter index: Indeks dari elemen yang ingin diambil dari queue.
- Return: Mengembalikan elemen TransaksiDetail pada posisi index dalam queue. Indeks dihitung dengan menggunakan operasi modulo untuk menangani wrap-around dalam queue.

TRANSAKSI DETAIL

```
public class TransaksiDetail {
    private String idTransaksi;
    private String idMobil;
    private String namaPembeli;
    private String noHpPembeli;
    private String metodePembayaran;
    private String tanggalPembelian;
    private String sales;
    private int omzet;
```

- String idTransaksi: Variabel ini menyimpan ID unik untuk setiap transaksi.
- String idMobil: Variabel ini menyimpan ID mobil yang terlibat dalam transaksi.
- String namaPembeli: Variabel ini menyimpan nama pembeli yang melakukan transaksi.
- String noHpPembeli: Variabel ini menyimpan nomor HP pembeli.
- String metodePembayaran: Variabel ini menyimpan metode pembayaran yang digunakan dalam transaksi (misalnya, tunai atau kredit).
- String tanggalPembelian: Variabel ini menyimpan tanggal pembelian mobil.
- String sales: Variabel ini menyimpan nama sales yang menangani transaksi.

```
public TransaksiDetail(String idTransaksi, String idMobil, String namaPembeli, String noHpPembeli, String metodePembayaran, String tanggalPembelian, String sales, int
    this.idTransaksi = idTransaksi;
    this.idMobil = idMobil;
    this.namaPembeli = namaPembeli;
    this.noHpPembeli = noHpPembeli;
    this.metodePembayaran = metodePembayaran;
    this.tanggalPembelian = tanggalPembelian;
    this.sales = sales;
    this.omzet = omzet;
}
```

- TransaksiDetail(String idTransaksi, String idMobil, String namaPembeli, String noHpPembeli, String metodePembayaran, String tanggalPembelian, String sales): Konstruktor ini digunakan untuk membuat objek TransaksiDetail baru dengan semua atribut yang diberikan sebagai parameter.
- Parameter idTransaksi: ID unik untuk transaksi.
- Parameter idMobil: ID mobil yang terlibat dalam transaksi.
- Parameter namaPembeli: Nama pembeli yang melakukan transaksi.
- Parameter noHpPembeli: Nomor HP pembeli.
- Parameter metodePembayaran: Metode pembayaran yang digunakan dalam transaksi.
- Parameter tanggalPembelian: Tanggal pembelian mobil.
- Parameter sales: Nama sales yang menangani transaksi.

```java
public String getIdTransaksi() {
    return idTransaksi;
}

public String getIdMobil() {
    return idMobil;
}

public String getNamaPembeli() {
    return namaPembeli;
}

public String getNoHpPembeli() {
    return noHpPembeli;
}

public String getMetodePembayaran() {
    return metodePembayaran;
}

public String getTanggalPembelian() {
    return tanggalPembelian;
}

public String getSales() {
    return sales;
}

public int getOmzet() {
    return omzet;
}
```

```java
public void setIdMobil(String idMobil) {
    this.idMobil = idMobil;
}

public void setNamaPembeli(String namaPembeli) {
    this.namaPembeli = namaPembeli;
}

public void setNoHpPembeli(String noHpPembeli) {
    this.noHpPembeli = noHpPembeli;
}

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

public void setTanggalPembelian(String tanggalPembelian) {
    this.tanggalPembelian = tanggalPembelian;
}

public void setSales(String sales) {
    this.sales = sales;
}

public void setOmzet(int omzet) {
    this.omzet = omzet;
}
```

- getIdTransaksi(): Mengembalikan nilai dari variabel idTransaksi.
- Return: Mengembalikan ID transaksi.
- getIdMobil(): Mengembalikan nilai dari variabel idMobil.
- Return: Mengembalikan ID mobil yang terlibat dalam transaksi.
- getNamaPembeli(): Mengembalikan nilai dari variabel namaPembeli.
- Return: Mengembalikan nama pembeli.
- getNoHpPembeli(): Mengembalikan nilai dari variabel noHpPembeli.
- Return: Mengembalikan nomor HP pembeli.
- getMetodePembayaran(): Mengembalikan nilai dari variabel metodePembayaran.
- Return: Mengembalikan metode pembayaran yang digunakan dalam transaksi.
- getTanggalPembelian(): Mengembalikan nilai dari variabel tanggalPembelian.
- Return: Mengembalikan tanggal pembelian mobil.
- getSales(): Mengembalikan nilai dari variabel sales.
- Return: Mengembalikan nama sales yang menangani transaksi.

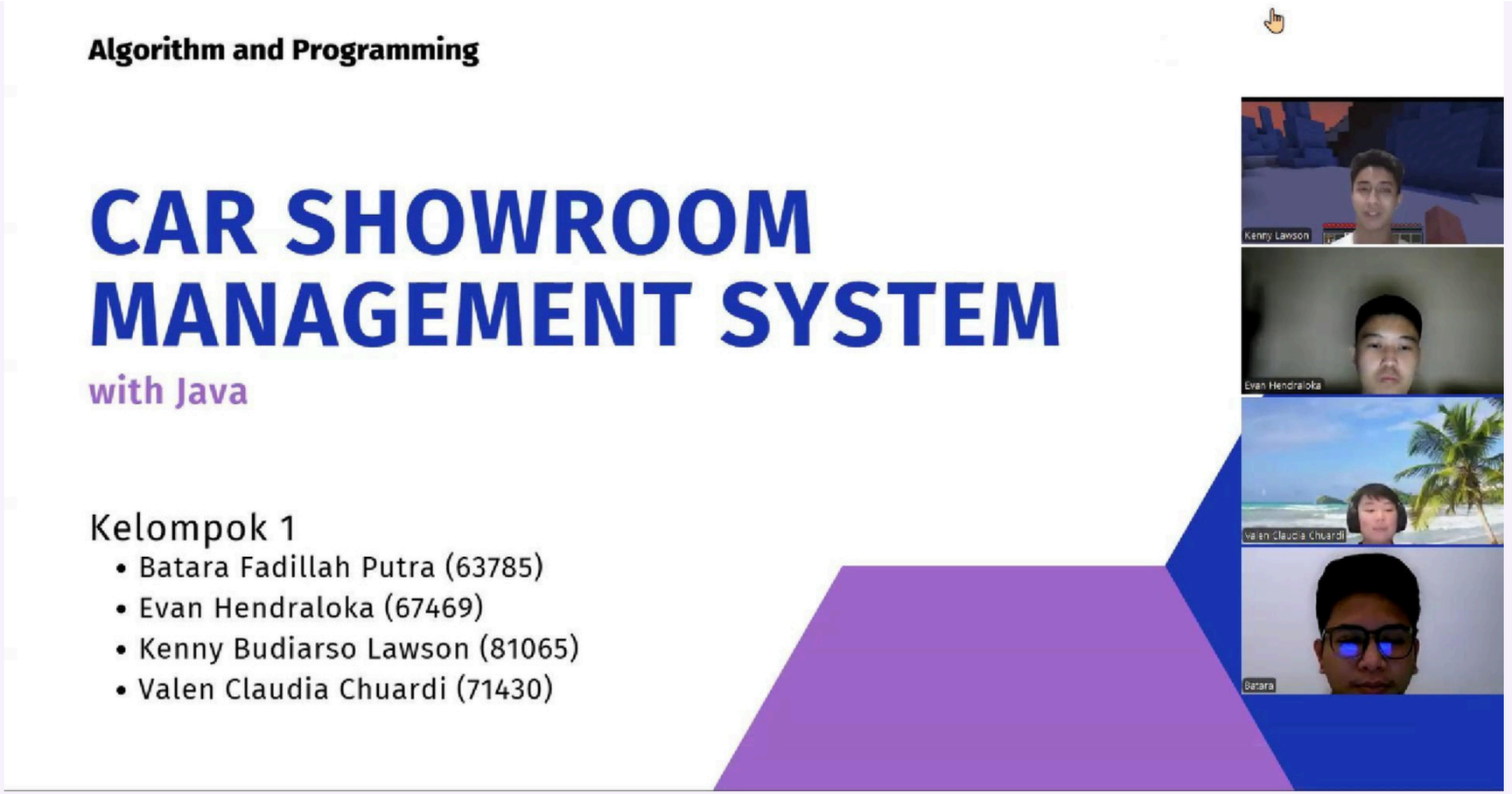
```java
@Override
public String toString() {
    return "ID Transaksi: " + idTransaksi + ", ID Mobil: " + idMobil + ", Nama Pembeli: " + namaPembeli + ", No. HP Pembeli: " + noHpPembeli + ", Metode Pembayaran: "
```

DEMO APLIKASI SISTEM

Thank you

Thank you for participating.

LINK VIDEO PRESENTASI



LINK GOOGLE DRIVE VIDEO PRESENTASI