

**UJIAN TENGAH SEMESTER (UTS)
GANJIL 2025/2026**

**Mata Kuliah:
PEMROGRAMAN BERORIENTASI OBJEK**

Kelas D3TI2.C

Studi Kasus: Aplikasi Pengelolaan Pangkalan Gas LPG 3Kg Berbasis Console



Oleh:
Cania Nabilatul Adawah
2403102

**D3 TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI INDRAMAYU
OKTOBER 2025**

Daftar Isi

Deskripsi Permasalahan.....	3
Analisis Kebutuhan.....	3
Analisis Fitur.....	3
Rancangan UML Use Case Diagram.....	4
Rancangan Tampilan Program Aplikasi Console.....	5
Tampilan 1 Menu Utama.....	5
Tampilan 2 Daftar Pegawai	5
Tampilan 3 Rincian Transaksi.....	5
Analisis Class.....	5
Identifikasi Class dan Attribute	5
Identifikasi Method	6
Jenis Relasi dan Alasan.....	6
Multiplicity	7
Rancangan UML Class Diagram.....	8
Rancangan UML Sequence Diagram	10
Sequence Diagram: Login Petugas	10
Sequence Diagram: Tambah Transaksi.....	10
Sequence Diagram: Lihat Stok Gas	11
Sequence Diagram: Lihat Pegawai	12
Pra-Kode Program	13
Kode Program	13
Gas.java.....	13
MainApp.java.....	14
Pangkalan.java	15
Pegawai.java	16
Pembayaran.java.....	17
Transaksi.java.....	18
Compile & Run.....	20
Compile.....	20
Run	20
Testing.....	20
Skenario Lihat Pegawai (Berhasil)	20
Skenario Lihat Stok Gas (Berhasil)	20
Skenario Tambah Transaksi (Berhasil).....	21

Skenario Lihat Transaksi (Berhasil).....	21
Skenario Keluar (Berhasil).....	21
Build (Deploy)	22
URL Repository	22

Deskripsi Permasalahan

Pangkalan gas LPG 3Kg merupakan salah satu usaha yang membutuhkan pengelolaan data yang cepat dan teratur. Selama ini banyak pangkalan yang masih mencatat transaksi pembelian dan pengiriman tabung gas secara manual. Hal ini sering menimbulkan masalah seperti kesalahan pencatatan, hilangnya data transaksi, dan tidak akuratnya jumlah stok gas yang tersedia. Untuk mengatasi masalah tersebut, dibuatlah Aplikasi Pengelolaan Pangkalan Gas LPG 3Kg Berbasis Console yang berfungsi untuk mencatat transaksi pembelian gas, melihat stok gas, menampilkan daftar pegawai pengantar, serta mencatat metode pembayaran pelanggan (BRI, DANA, atau Mandiri).

Analisis Kebutuhan

Aplikasi ini dirancang agar dapat membantu pemilik atau petugas pangkalan gas LPG 3Kg dalam melakukan kegiatan operasional harian secara lebih teratur, efisien, dan bebas dari kesalahan pencatatan manual.

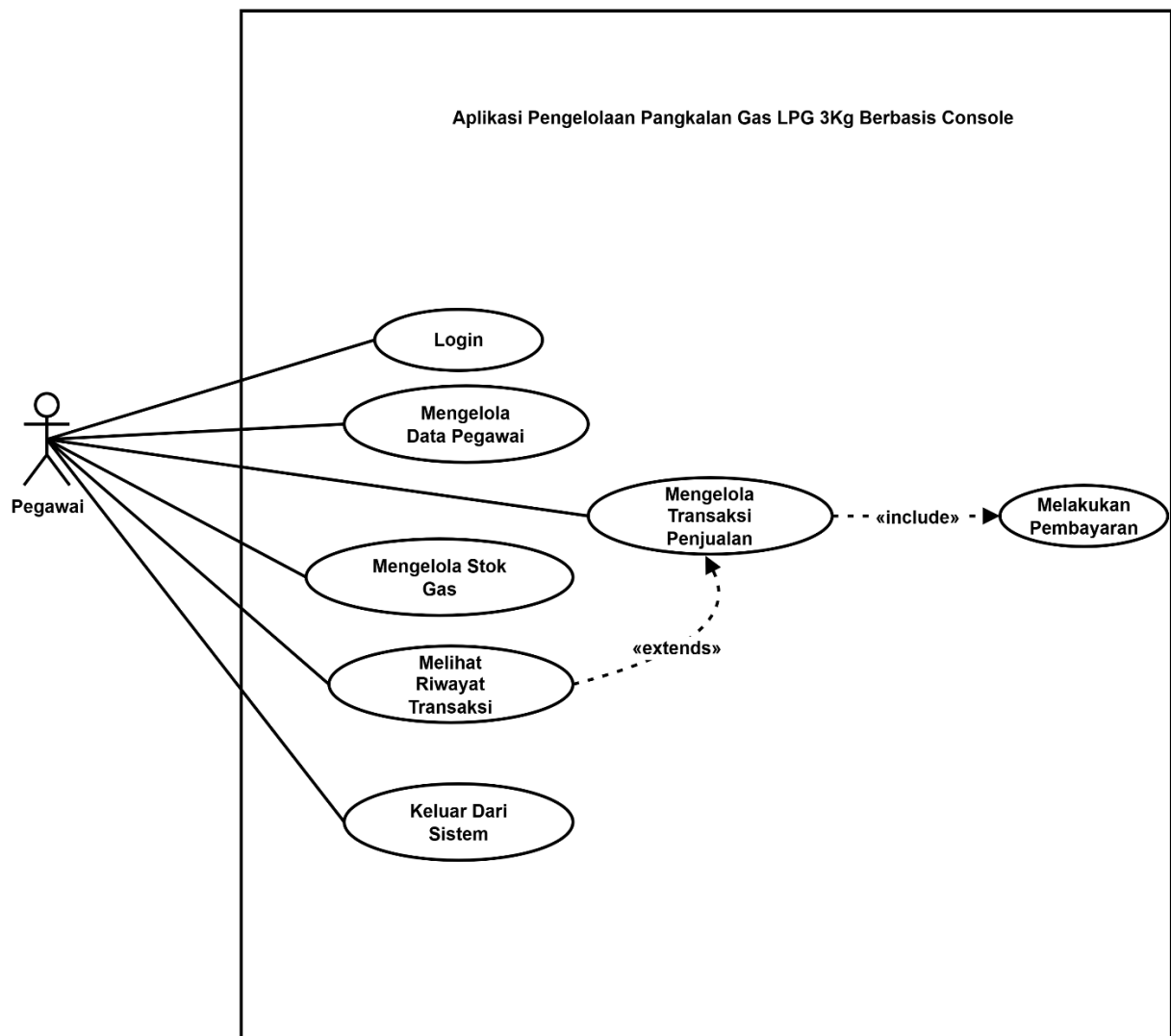
Aplikasi ini dirancang agar dapat:

1. Mencatat transaksi pembelian gas oleh pelanggan.
2. Menampilkan daftar pegawai pengantar tabung gas.
3. Mengatur dan menampilkan stok gas yang tersedia.
4. Mengurangi stok gas secara otomatis setiap kali transaksi dilakukan.
5. Menampilkan metode pembayaran pelanggan.
6. Menampilkan laporan hasil transaksi di console.

Analisis Fitur

No	Fitur	Kebutuhan	Alur
1	Login Petugas	Membatasi akses agar hanya petugas yang dapat masuk ke sistem.	Petugas membuka aplikasi lalu mengisi username dan password. Jika benar, masuk ke menu utama, jika salah diminta ulang.
2	Lihat Pegawai	Menampilkan daftar pegawai pengantar gas.	Petugas memilih menu “Lihat Pegawai”, sistem menampilkan ID dan nama pegawai di console.
3	Lihat Stok Gas	Menampilkan jumlah tabung gas yang tersedia.	Setelah menu “Lihat Stok Gas” dipilih, sistem menampilkan jumlah stok terkini di console.
4	Tambah Transaksi	Mencatat pembelian gas oleh pelanggan.	Petugas mengisi nama pelanggan, jumlah tabung, metode pembayaran, dan memilih pegawai pengantar. Sistem menghitung total harga dan menampilkan hasil transaksi.
5	Pembayaran	Menyimpan metode pembayaran pelanggan (BRI, DANA, Mandiri).	Saat transaksi, petugas memilih metode pembayaran lalu sistem mencatat dan menyimpannya dalam rincian transaksi.
6	Keluar Aplikasi	Menutup aplikasi setelah semua proses selesai.	Petugas memilih menu “Keluar”, sistem menampilkan pesan terima kasih dan program berhenti berjalan.

Rancangan UML Use Case Diagram



Rancangan Tampilan Program Aplikasi Console

Tampilan 1 Menu Utama

```
==== MENU PANGKALAN GAS LPG 3KG ====
1. Lihat Pegawai
2. Lihat Stok Gas
3. Tambah Transaksi
4. Keluar
Pilih menu: _
```

Tampilan 2 Daftar Pegawai

```
==== Daftar Pegawai ====
ID Pegawai: P01 | Nama: Andi
ID Pegawai: P02 | Nama: Budi
```

Tampilan 3 Rincian Transaksi

```
==== Rincian Transaksi ====
Pelanggan: Siti
Jumlah Tabung: 2
Total Bayar: Rp 36000
Diantar oleh: Andi
Metode Bayar: DANA
=====
```

Analisis Class

Identifikasi Class dan Attribute

Berdasarkan analisis kebutuhan, didapatkan daftar identifikasi class dan attribute sebagai berikut:

No	Class (Kata Benda)	Attribute (Kata Benda)
1	Pangkalan	NamaPangkalan: String
		Alamat: String
		daftarPegawai: ArrayList<Pegawai>
		stokGas: Gas

		daftarTransaksi: ArrayList<Transaksi>
2	Pegawai	idPegawai: String namaPegawai: String
3	Gas	idGas: String harga: double jumlahStok: int
4	Transaksi	idTransaksi: String namaPelanggan: String jumlahTabung: int totalBayar: double pegawai: Pegawai pembayaran: Pembayaran
5	Pembayaran	metode: String jumlahBayar: double Status: String

Identifikasi Method

Berdasarkan analisis kebutuhan dan aktivitas dalam sistem, didapatkan daftar identifikasi method sebagai berikut:

No	Class (Kata Benda)	Method (Kata Kerja)	Parameter (Kata Benda)
1	Pangkalan	tambahPegawai(): void	pegawai: Pegawai
		tampilkanPegawai(): void	-
		tampilkanStok(): void	-
		tambahTransaksi(): void	pelanggan: String, jumlah: int, metode: String, idPegawai: String
		tampilkanTransaksi(): void	-
2	Pegawai	getNama(): String	-
		getId(): String	-
3	Gas	kurangiStok(): void	Jumlah: int
		getHarga(): double	-
4	Transaksi	hitungTotal(): double	harga: double, jumlah: int
		tampilkanRincian(): void	-
5	Pembayaran	prosesPembayaran(): void	metode: String, jumlah: double

Jenis Relasi dan Alasan

a. Dependency

No	Class Awal	Class Tujuan	Alasan
1	Main	Pangkalan	Relasi ini bersifat sementara, di mana kelas Main hanya bergantung pada kelas Pangkalan untuk menjalankan program utama (entry point). Setelah program selesai, objek tidak lagi digunakan.

b. Unidirectional Association

No	Class Awal	Class Tujuan	Alasan
1	Transaksi	Pegawai	Transaksi mengenali pegawai yang bertugas mengantar gas, tetapi pegawai tidak mengetahui rincian transaksi. Oleh karena itu, arah relasi hanya satu arah dari Transaksi ke Pegawai.

c. Bidirectional Association

Tidak digunakan karena tidak ada dua kelas yang saling mengenal dan membutuhkan data satu sama lain secara langsung.

d. Aggregation

No	Class Awal	Class Tujuan	Alasan
1	Pangkalan	Pegawai	Pangkalan memiliki beberapa pegawai, tetapi pegawai tetap dapat eksis walau objek Pangkalan

			dihapus. Relasi ini menggambarkan hubungan “memiliki tetapi tidak bergantung penuh”.
--	--	--	--

e. Composition

No	Class Awal	Class Tujuan	Alasan
1	Pangkalan	Transaksi	Hubungan ini menunjukkan kepemilikan penuh (<i>strong ownership</i>). Misalnya, jika Pangkalan dihapus maka daftar Transaksi dan data Gas ikut terhapus. Begitu pula Pembayaran yang tidak bisa eksis tanpa Transaksi.
	Pangkalan	Gas	
	Transaksi	Pembayaran	

f. Generalization

Tidak digunakan karena sistem tidak memiliki pewarisan antar kelas (tidak ada superclass dan subclass).

g. Realization

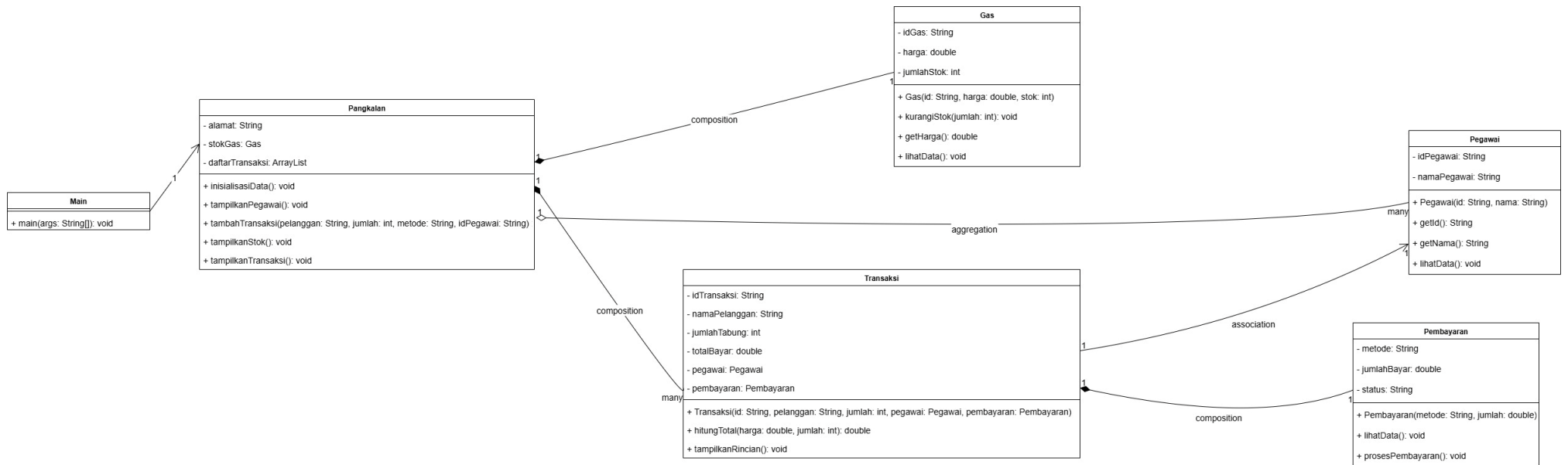
Tidak digunakan karena tidak terdapat interface yang diimplementasikan.

Multiplicity

Berikut adalah penentuan kardinalitas (multiplicity) antar *class* yang saling terhubung pada relasi dependency, association, aggregation, dan composition dalam sistem Aplikasi Pengelolaan Pangkalan Gas LPG 3Kg Berbasis Console:

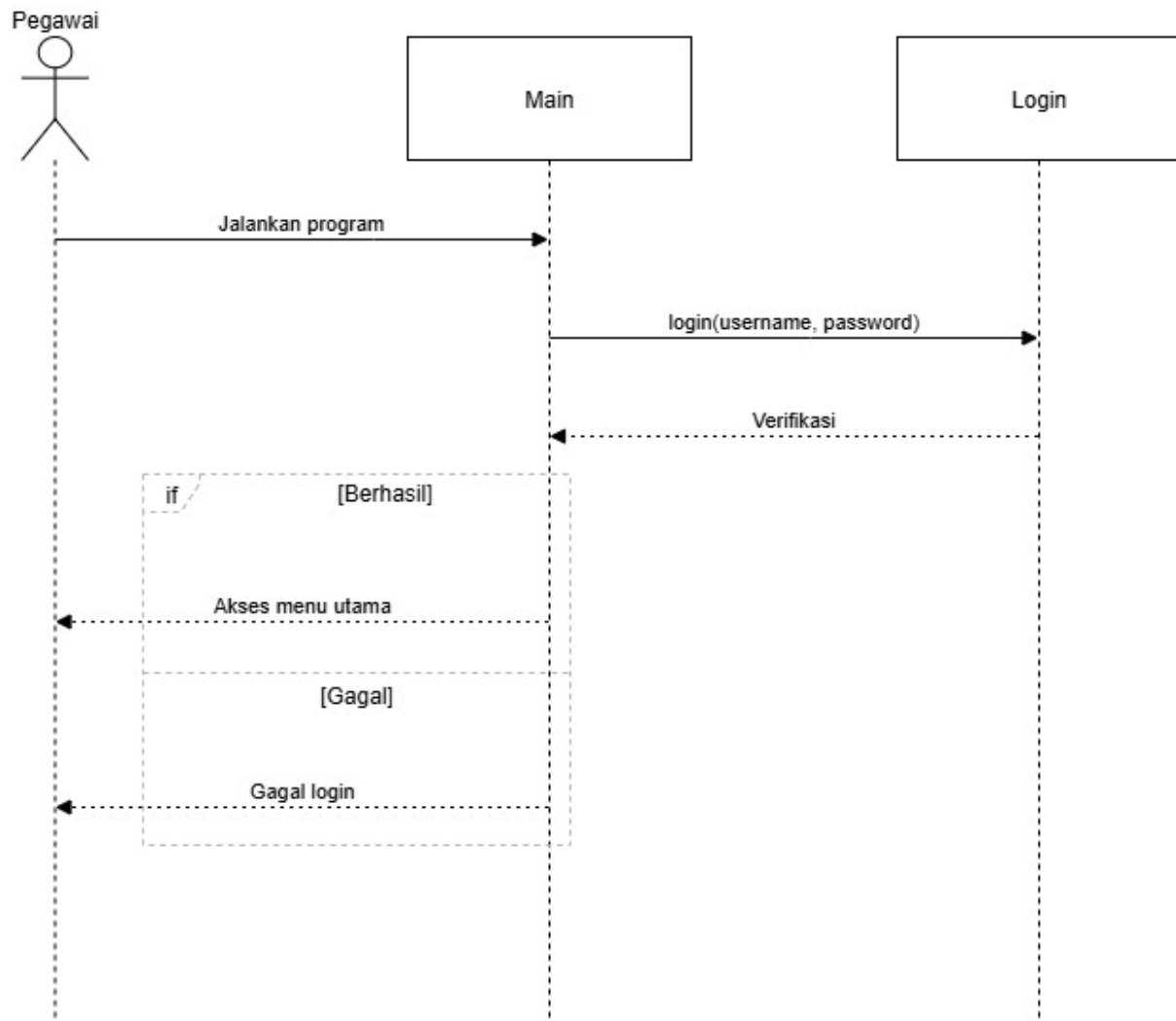
No	Class Awal	Class Tujuan	Relasi	Multiplicity	Keterangan
1	Main	Pangkalan	Dependency	1 & 1	Satu kelas Main hanya bergantung pada satu objek Pangkalan untuk menjalankan proses utama aplikasi (entry point).
2	Transaksi	Pegawai	Unidirectional Association	1 & 1	Setiap Transaksi hanya ditangani oleh satu Pegawai. Namun, satu Pegawai dapat menangani beberapa Transaksi (arah relasi satu arah).
3	Pangkalan	Pegawai	Aggregation	1 & ...*	Satu Pangkalan dapat memiliki banyak Pegawai, namun Pegawai tetap dapat berdiri sendiri jika Pangkalan dihapus.
4	Pangkalan	Gas	Composition	1 & 1	Satu Pangkalan hanya memiliki satu stok Gas aktif yang dikelola. Bila Pangkalan dihapus, data Gas ikut hilang.
5	Pangkalan	Transaksi	Composition	1 & ...*	Satu Pangkalan dapat memiliki banyak Transaksi. Jika Pangkalan dihapus, seluruh Transaksi yang terkait juga terhapus.
6	Transaksi	Pembayaran	Composition	1 & 1	Setiap Transaksi memiliki satu Pembayaran, dan Pembayaran tidak dapat berdiri tanpa Transaksi.

Rancangan UML Class Diagram

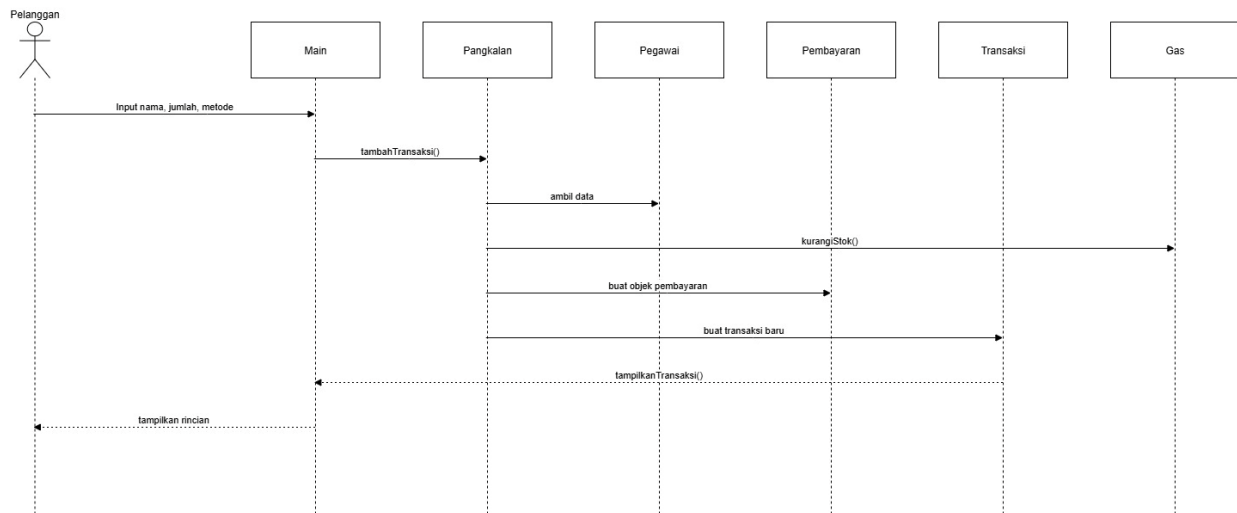


Rancangan UML Sequence Diagram

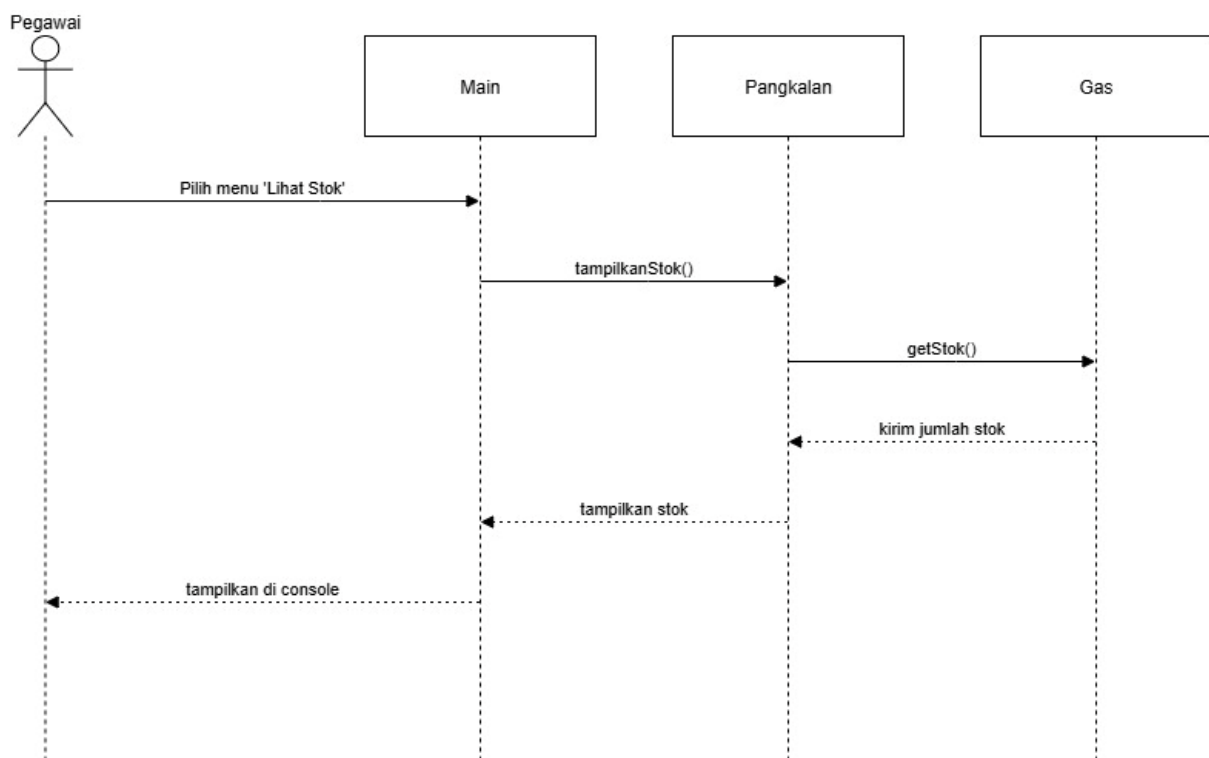
Sequence Diagram: Login Petugas



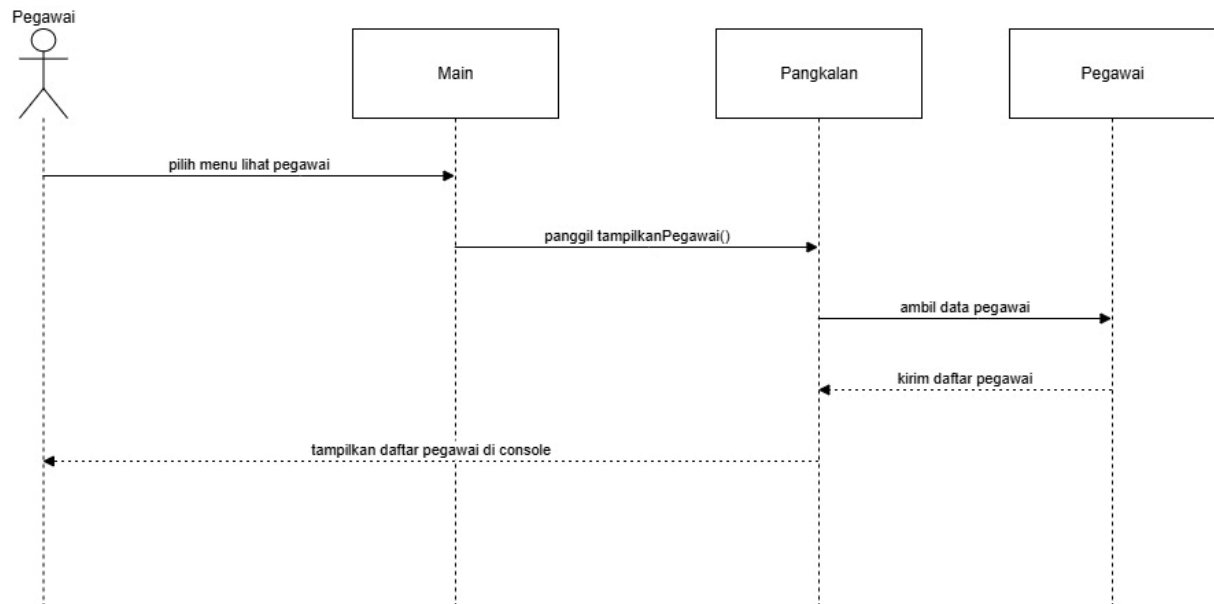
Sequence Diagram: Tambah Transaksi



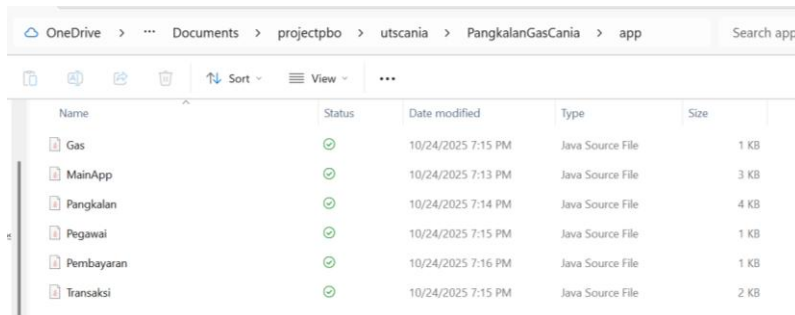
Sequence Diagram: Lihat Stok Gas



Sequence Diagram: Lihat Pegawai



Pra-Code Program



The screenshot shows a file explorer window with the path: OneDrive > Documents > projectpbo > utscania > PangkalanGasCania > app. The search bar contains 'app'. The file list is as follows:

Name	Status	Date modified	Type	Size
Gas	✓	10/24/2025 7:15 PM	Java Source File	1 KB
MainApp	✓	10/24/2025 7:13 PM	Java Source File	3 KB
Pangkalan	✓	10/24/2025 7:14 PM	Java Source File	4 KB
Pegawai	✓	10/24/2025 7:15 PM	Java Source File	1 KB
Pembayaran	✓	10/24/2025 7:16 PM	Java Source File	1 KB
Transaksi	✓	10/24/2025 7:15 PM	Java Source File	2 KB

Kode Program

Gas.java

```
package app;

/**
 * Kelas yang menyimpan informasi stok gas LPG
 * Termasuk ID gas, harga per tabung, dan jumlah stok
 */
public class Gas {
    private String idGas;
    private double harga;
    private int jumlahStok;

    public Gas(String id, double harga, int stok) {
        this.idGas = id;
        this.harga = harga;
        this.jumlahStok = stok;
    }

    // Mengurangi jumlah stok gas setelah transaksi
    public void kurangiStok(int jumlah) {
        this.jumlahStok -= jumlah;
    }

    // Getter harga gas
    public double getHarga() {
        return harga;
    }

    // Getter jumlah stok
    public int getJumlahStok() {
        return jumlahStok;
    }

    // Menampilkan data stok gas
    public void lihatData() {
        System.out.println("ID Gas: " + idGas + " | Harga: Rp" + harga + " | Stok: " + jumlahStok + " tabung");
    }
}
```

Gas
- idGas: String - harga: double - jumlahStok: int
+ Gas(id: String, harga: double, stok: int) + kurangiStok(jumlah: int): void + getHarga(): double + lihatData(): void

MainApp.java

```
package app;

import java.util.Scanner;

/**
 * Kelas utama untuk menjalankan program aplikasi console
 * Berisi menu utama dan logika pemilihan fitur
 */
public class MainApp {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Membuat objek pangkalan (kelas utama)
        Pangkalan pangkalan = new Pangkalan("Pangkalan LPG Sejahtera", "Jl. Merdeka No.10");
        pangkalan.inisialisasiData(); // Inisialisasi pegawai dan stok gas

        int pilihan;
        do {
            // Menampilkan menu utama
            System.out.println("=== MENU PANGKALAN GAS LPG 3KG ===");
            System.out.println("1. Lihat Pegawai");
            System.out.println("2. Lihat Stok Gas");
            System.out.println("3. Tambah Transaksi");
            System.out.println("4. Lihat Transaksi");
            System.out.println("5. Keluar");
            System.out.print("Pilih menu: ");
            pilihan = input.nextInt();
            input.nextLine(); // membersihkan buffer input

            // Menjalankan menu berdasarkan pilihan
            switch (pilihan) {
                case 1:
                    pangkalan.tampilkanPegawai(); // menampilkan data pegawai
                    break;
                case 2:
                    pangkalan.tampilkanStok(); // menampilkan stok gas
                    break;
                case 3:
                    // Input data transaksi baru
                    System.out.print("Nama Pelanggan: ");
                    String nama = input.nextLine();
                    System.out.print("Jumlah Tabung: ");
                    int jumlah = input.nextInt();
                    input.nextLine();
                    System.out.print("Metode Pembayaran (BRI/DANA/Mandiri): ");
                    String metode = input.nextLine();
                    System.out.print("ID Pegawai Pengantar (P01/P02): ");
                    String idPegawai = input.nextLine();

                    pangkalan.tambahTransaksi(nama, jumlah, metode, idPegawai);
                    break;
            }
        } while (pilihan != 5);
    }
}
```

```

        case 4:
            pangkalan.tampilkanTransaksi(); // menampilkan semua transaksi
            break;
        case 5:
            System.out.println("Terima kasih, program selesai.");
            break;
        default:
            System.out.println("Pilihan tidak valid!");
    }
    System.out.println();
} while (pilihan != 5);

input.close(); // menutup scanner
}
}

```

Pangkalan.java

```

package app;

import java.util.ArrayList;

/**
 * Kelas utama yang merepresentasikan pangkalan gas LPG
 * Menyimpan daftar pegawai, stok gas, dan transaksi
 */
public class Pangkalan {
    // Atribut private untuk menerapkan konsep enkapsulasi
    private String namaPangkalan;
    private String alamat;
    private ArrayList<Pegawai> daftarPegawai; // Aggregation ke Pegawai
    private Gas stokGas; // Composition ke Gas
    private ArrayList<Transaksi> daftarTransaksi; // Composition ke Transaksi

    // Konstruktor untuk inisialisasi objek Pangkalan
    public Pangkalan(String nama, String alamat) {
        this.namaPangkalan = nama;
        this.alamat = alamat;
        this.daftarPegawai = new ArrayList<>();
        this.daftarTransaksi = new ArrayList<>();
    }

    // Method untuk menambahkan data awal
    public void inisialisasiData() {
        daftarPegawai.add(new Pegawai("P01", "Andi"));
        daftarPegawai.add(new Pegawai("P02", "Budi"));
        stokGas = new Gas("G01", 18000, 50); // stok awal 50 tabung
    }

    // Menampilkan daftar pegawai
    public void tampilkanPegawai() {
        System.out.println("=== Daftar Pegawai ===");
        for (Pegawai p : daftarPegawai) {
            System.out.println("ID Pegawai: " + p.getId() + " | Nama: " + p.getNama());
        }
    }

    // Menampilkan stok gas yang tersedia
    public void tampilkanStok() {
        System.out.println("=== Stok Gas ===");
        stokGas.lihatData();
    }

    // Menambahkan transaksi baru
    public void tambahTransaksi(String pelanggan, int jumlah, String metode, String idPegawai) {
        // Mencari pegawai berdasarkan ID
        Pegawai pengantar = null;
        for (Pegawai p : daftarPegawai) {
            if (p.getId().equalsIgnoreCase(idPegawai)) {
                pengantar = p;
                break;
            }
        }
    }
}

```



```

    }

    // Validasi data pegawai
    if (pengantar == null) {
        System.out.println("Pegawai tidak ditemukan!");
        return;
    }

    // Cek stok gas cukup atau tidak
    if (stokGas.getJumlahStok() < jumlah) {
        System.out.println("Stok gas tidak mencukupi!");
        return;
    }

    // Kurangi stok gas sesuai pembelian
    stokGas.kurangiStok(jumlah);

    // Buat objek pembayaran dan transaksi baru
    Pembayaran bayar = new Pembayaran(metode, stokGas.getHarga() * jumlah);
    Transaksi t = new Transaksi("T" + (daftarTransaksi.size() + 1), pelanggan, jumlah, pengantar, bayar);
    daftarTransaksi.add(t);

    // Tampilkan hasil transaksi
    System.out.println("Transaksi berhasil ditambahkan!");
    t.tampilkanRincian();
}

// Menampilkan semua transaksi yang tersimpan
public void tampilkanTransaksi() {
    System.out.println("=== Daftar Transaksi ===");
    for (Transaksi t : daftarTransaksi) {
        t.tampilkanRincian();
        System.out.println("-----");
    }
}
}

```

Pangkalan
<ul style="list-style-type: none"> - alamat: String - stokGas: Gas - daftarTransaksi: ArrayList
<ul style="list-style-type: none"> + inisialisasiData(): void + tampilkanPegawai(): void + tambahTransaksi(pelanggan: String, jumlah: int, metode: String, idPegawai: String) + tampilkanStok(): void + tampilkanTransaksi(): void

Pegawai.java

```

package app;

/**
 * Kelas untuk menyimpan data pegawai (petugas)
 * Mewakili aktor yang bertugas mengantar gas ke pelanggan
 */
public class Pegawai {
    private String idPegawai;
    private String namaPegawai;

    public Pegawai(String id, String nama) {
        this.idPegawai = id;
        this.namaPegawai = nama;
    }

    // Mengembalikan ID pegawai

```

```

public String getId() {
    return idPegawai;
}

// Mengembalikan nama pegawai
public String getNama() {
    return namaPegawai;
}

// Menampilkan data pegawai ke console
public void lihatData() {
    System.out.println("Pegawai: " + idPegawai + " - " + namaPegawai);
}
}

```

Pegawai
- idPegawai: String - namaPegawai: String
+ Pegawai(id: String, nama: String) + getId(): String + getNama(): String + lihatData(): void

Pembayaran.java

```

package app;

/**
 * Kelas untuk mencatat data pembayaran pelanggan
 * Termasuk metode, jumlah, dan status pembayaran
 */
public class Pembayaran {
    private String metode;
    private double jumlahBayar;
    private String status;

    public Pembayaran(String metode, double jumlah) {
        this.metode = metode;
        this.jumlahBayar = jumlah;
        this.status = "Lunas"; // default pembayaran berhasil
    }

    // Getter jumlah bayar
    public double getJumlahBayar() {
        return jumlahBayar;
    }

    // Menampilkan proses pembayaran
    public void prosesPembayaran() {
        System.out.println("Pembayaran via " + metode + " sebesar Rp" + jumlahBayar + " berhasil!");
    }

    // Menampilkan data pembayaran
    public void lihatData() {
        System.out.println("Metode Bayar: " + metode + " | Status: " + status);
    }
}

```

--

Pembayaran
- metode: String
- jumlahBayar: double
- status: String
+ Pembayaran(metode: String, jumlah: double)
+ lihatData(): void
+ prosesPembayaran(): void

Transaksi.java

```
package app;

/**
 * Kelas untuk mencatat setiap transaksi pembelian gas
 * Menghubungkan antara pelanggan, pegawai, dan pembayaran
 */
public class Transaksi {
    private String idTransaksi;
    private String namaPelanggan;
    private int jumlahTabung;
    private double totalBayar;
    private Pegawai pegawai; // Association ke Pegawai
    private Pembayaran pembayaran; // Composition ke Pembayaran

    // Konstruktor transaksi
    public Transaksi(String id, String pelanggan, int jumlah, Pegawai pegawai, Pembayaran pembayaran) {
        this.idTransaksi = id;
        this.namaPelanggan = pelanggan;
        this.jumlahTabung = jumlah;
        this.pegawai = pegawai;
        this.pembayaran = pembayaran;
        this.totalBayar = hitungTotal(pembayaran.getJumlahBayar());
    }

    // Menghitung total pembayaran (harga * jumlah tabung)
    public double hitungTotal(double nominal) {
        return nominal;
    }

    // Menampilkan detail transaksi
    public void tampilkanRincian() {
        System.out.println("=== Rincian Transaksi ===");
        System.out.println("ID Transaksi: " + idTransaksi);
        System.out.println("Pelanggan: " + namaPelanggan);
        System.out.println("Jumlah Tabung: " + jumlahTabung);
        System.out.println("Total Bayar: Rp " + totalBayar);
        System.out.println("Diantar oleh: " + pegawai.getNama());
        pembayaran.lihatData(); // menampilkan info pembayaran
    }
}
```

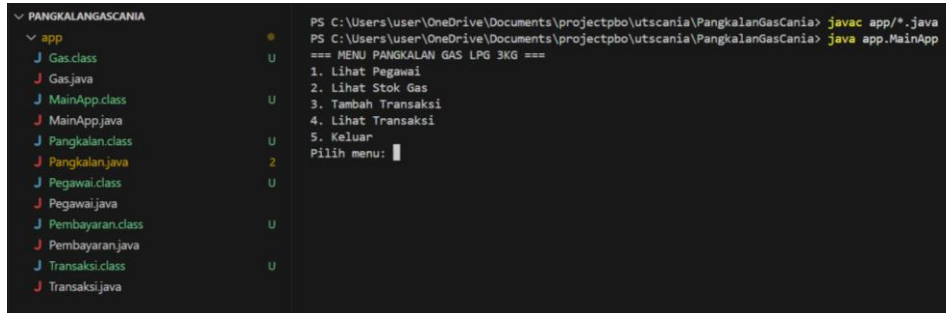
Transaksi
<ul style="list-style-type: none"> - idTransaksi: String - namaPelanggan: String - jumlahTabung: int - totalBayar: double - pegawai: Pegawai - pembayaran: Pembayaran
<ul style="list-style-type: none"> + Transaksi(id: String, pelanggan: String, jumlah: int, pegawai: Pegawai, pembayaran: Pembayaran) + hitungTotal(harga: double, jumlah: int): double + tampilkanRincian(): void

Compile & Run

Compile

```
PS C:\Users\user\OneDrive\Documents\projectpbo\utscania\PangkalanGasCania> javac app/*.java
PS C:\Users\user\OneDrive\Documents\projectpbo\utscania\PangkalanGasCania> java app.MainApp
```

Run

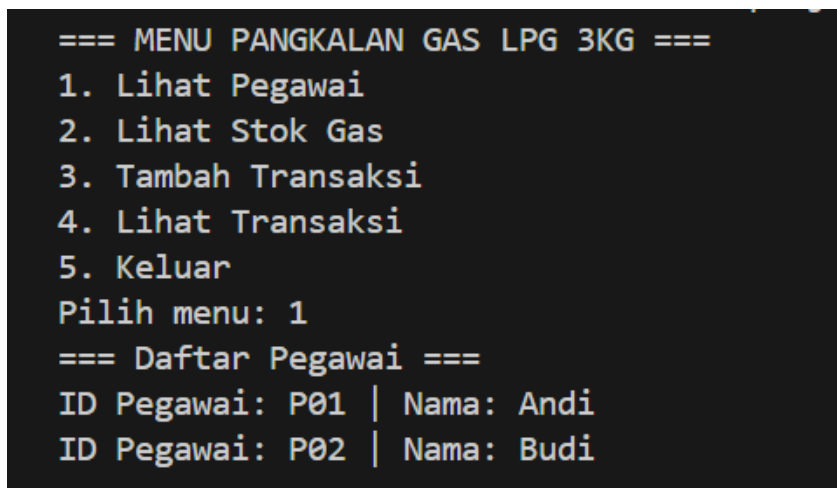


The screenshot shows an IDE with a project named 'PANGKALANGASCANIA'. The project structure on the left includes a package 'app' containing several classes: Gas.class, Gas.java, MainApp.class, MainApp.java, Pangkalan.class, Pangkalan.java, Pegawai.class, Pegawai.java, Pembayaran.class, Pembayaran.java, Transaksi.class, and Transaksi.java. The output console on the right shows the execution of the program, displaying a menu and the first option selected.

```
PS C:\Users\user\OneDrive\Documents\projectpbo\utscania\PangkalanGasCania> javac app/*.java
PS C:\Users\user\OneDrive\Documents\projectpbo\utscania\PangkalanGasCania> java app.MainApp
=== MENU PANGKALAN GAS LPG 3KG ===
1. Lihat Pegawai
2. Lihat Stok Gas
3. Tambah Transaksi
4. Lihat Transaksi
5. Keluar
Pilih menu: 1
```

Testing

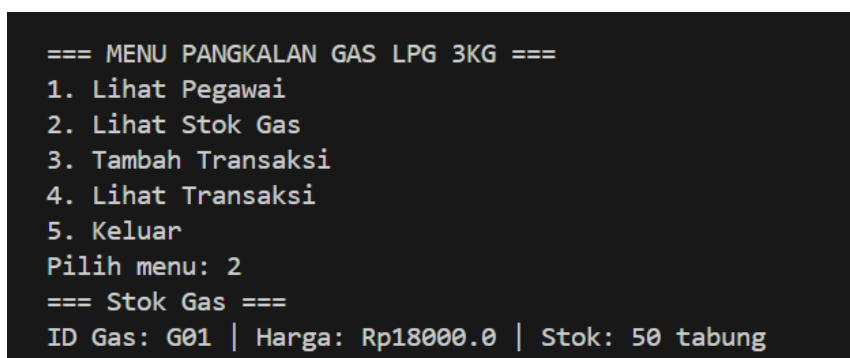
Skenario Lihat Pegawai (Berhasil)



The screenshot shows the console output for the 'Lihat Pegawai' scenario. It displays the menu, the selection of option 1, and the resulting list of employees.

```
=== MENU PANGKALAN GAS LPG 3KG ===
1. Lihat Pegawai
2. Lihat Stok Gas
3. Tambah Transaksi
4. Lihat Transaksi
5. Keluar
Pilih menu: 1
=== Daftar Pegawai ===
ID Pegawai: P01 | Nama: Andi
ID Pegawai: P02 | Nama: Budi
```

Skenario Lihat Stok Gas (Berhasil)



The screenshot shows the console output for the 'Lihat Stok Gas' scenario. It displays the menu, the selection of option 2, and the resulting gas stock information.

```
=== MENU PANGKALAN GAS LPG 3KG ===
1. Lihat Pegawai
2. Lihat Stok Gas
3. Tambah Transaksi
4. Lihat Transaksi
5. Keluar
Pilih menu: 2
=== Stok Gas ===
ID Gas: G01 | Harga: Rp18000.0 | Stok: 50 tabung
```

Skenario Tambah Transaksi (Berhasil)

```
=== MENU PANGKALAN GAS LPG 3KG ===
1. Lihat Pegawai
2. Lihat Stok Gas
3. Tambah Transaksi
4. Lihat Transaksi
5. Keluar
Pilih menu: 3
Nama Pelanggan: Cania
Jumlah Tabung: 2
Metode Pembayaran (BRI/DANA/Mandiri): DANA
ID Pegawai Pengantar (P01/P02): P01
Transaksi berhasil ditambahkan!
=== Rincian Transaksi ===
ID Transaksi: T1
Pelanggan: Cania
Jumlah Tabung: 2
Total Bayar: Rp 36000.0
Diantar oleh: Andi
Metode Bayar: DANA | Status: Lunas
```

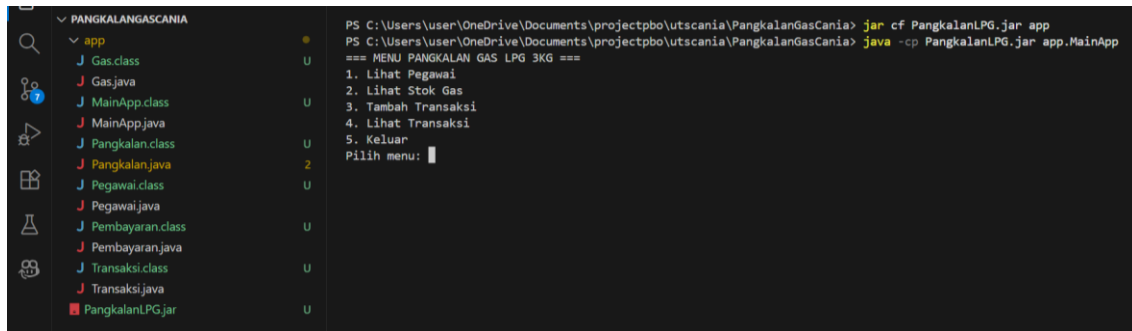
Skenario Lihat Transaksi (Berhasil)

```
=== MENU PANGKALAN GAS LPG 3KG ===
1. Lihat Pegawai
2. Lihat Stok Gas
3. Tambah Transaksi
4. Lihat Transaksi
5. Keluar
Pilih menu: 4
=== Daftar Transaksi ===
=== Rincian Transaksi ===
ID Transaksi: T1
Pelanggan: Cania
Jumlah Tabung: 2
Total Bayar: Rp 36000.0
Diantar oleh: Andi
Metode Bayar: DANA | Status: Lunas
-----
```

Skenario Keluar (Berhasil)

```
=== MENU PANGKALAN GAS LPG 3KG ===
1. Lihat Pegawai
2. Lihat Stok Gas
3. Tambah Transaksi
4. Lihat Transaksi
5. Keluar
Pilih menu: 5
Terima kasih, program selesai.
```

Build (Deploy)



```
PS C:\Users\user\OneDrive\Documents\projectpbo\utscania\PangkalanGasCania> jar cf PangkalanLPG.jar app
PS C:\Users\user\OneDrive\Documents\projectpbo\utscania\PangkalanGasCania> java -cp PangkalanLPG.jar app.MainApp
=== MENU PANGKALAN GAS LPG 3KG ===
1. Lihat Pegawai
2. Lihat Stok Gas
3. Tambah Transaksi
4. Lihat Transaksi
5. Keluar
Pilih menu: 
```

URL Repository

<https://github.com/caniaye/PangkalanGasCania.git>