

## A. Identitas Proyek

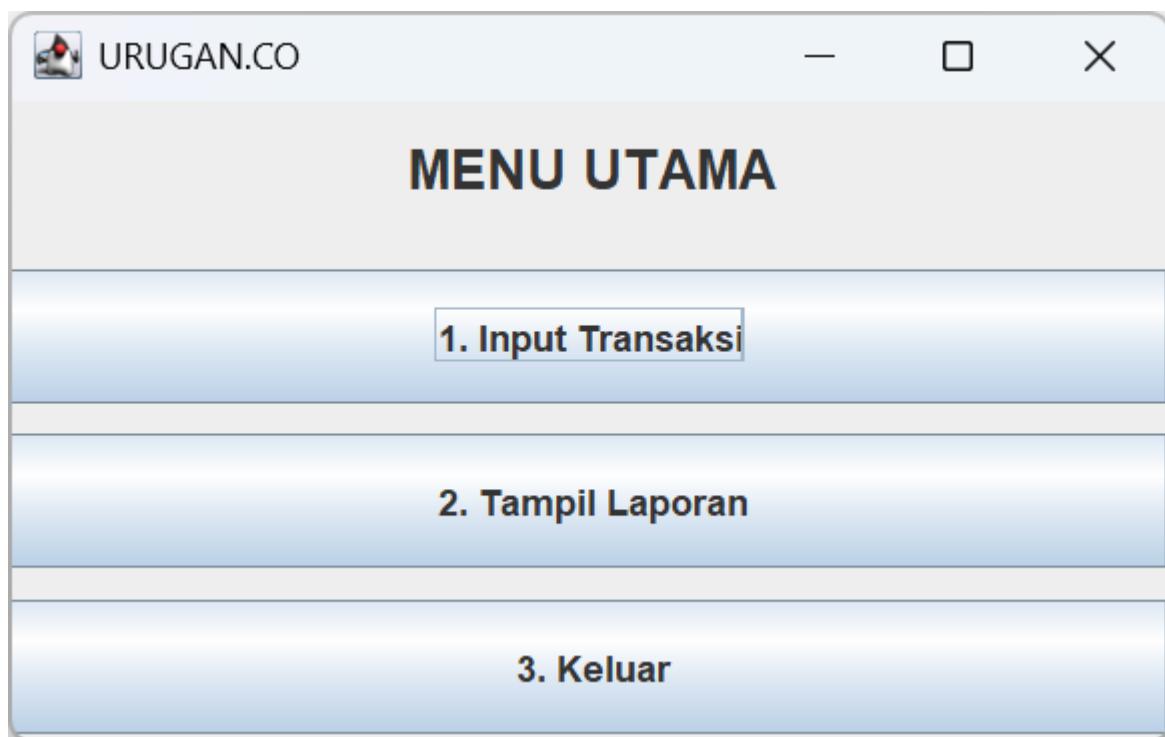
**Judul Proyek** : Sistem Pencatatan Urugan Tanah  
**Mata Kuliah** : Pemrograman Berorientasi Objek  
**Dosen Pengampu** : Tedy Setiadi, M.T  
**Tahun** : 2025

## Anggota Kelompok

1. Muhamad Faizin Al Bahri – 2400018057
2. Muhammad Fauzan Anwar – 2400018012
3. Latifatul Istiana – 2400018037

## Link Repository

<https://github.com/Artmzyyy/Sistem-Pencatatan-Urugan-PBO>



Tampilan Awal Aplikasi

## **B. Persoalan Bisnis dan Deskripsi Proyek**

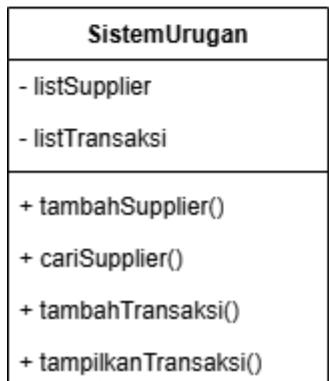
Pada proses pencatatan urugan tanah, sering ditemukan permasalahan seperti pencatatan manual yang tidak terstruktur, kesalahan perhitungan volume muatan, serta kesulitan dalam melihat rekap laporan transaksi. Hal ini dapat menyebabkan data transaksi tidak akurat dan sulit untuk ditelusuri kembali.

Berdasarkan permasalahan tersebut, dibuat sebuah aplikasi pencatatan transaksi urugan tanah berbasis Java dengan pendekatan Pemrograman Berorientasi Objek (PBO). Aplikasi ini bertujuan untuk membantu proses pencatatan transaksi secara terstruktur, menghitung volume muatan secara otomatis, serta menampilkan laporan transaksi dengan lebih rapi dan mudah dipahami.

## **C. Daftar Seluruh Spesifikasi Aplikasi**

- Spesifikasi aplikasi yang dirancang antara lain:
- Input data transaksi urugan tanah
- Input nomor polisi kendaraan dan supplier
- Input dimensi muatan (panjang, lebar, tinggi)
- Perhitungan volume otomatis dalam satuan meter kubik ( $m^3$ )
- Penyimpanan data transaksi menggunakan struktur data
- Menampilkan laporan transaksi
- Antarmuka aplikasi berbasis GUI menggunakan Java Swing

## D. Rancangan Model Diagram UML



Perancangan sistem menggunakan diagram kelas (Class Diagram) yang menggambarkan hubungan antar class dalam aplikasi. Class utama yang digunakan antara lain **Supplier**, **Truk**, **Dimensi**, **Transaksi**, dan **SistemUrugan**. Relasi antar class dirancang sesuai dengan konsep Pemrograman Berorientasi Objek, di mana satu supplier dapat memiliki banyak truk, dan setiap transaksi berkaitan dengan satu truk dan satu dimensi muatan.

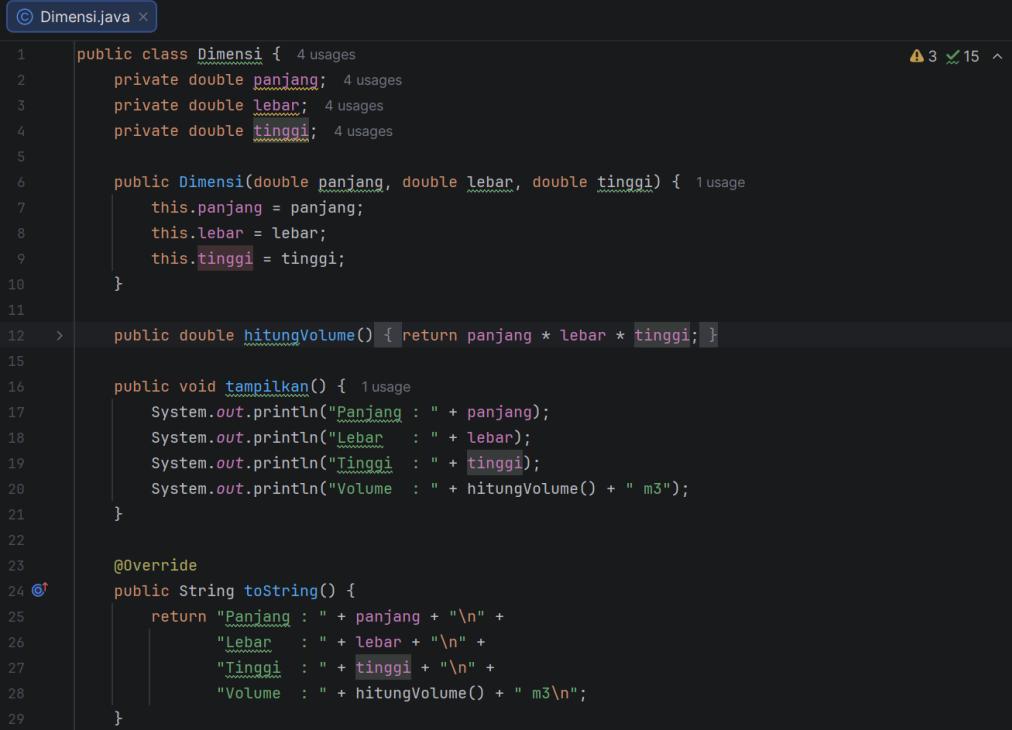
## E. Rancangan Antar Muka Berbasis GUI

Antarmuka aplikasi dirancang menggunakan Java Swing dengan tampilan sederhana dan mudah digunakan. Aplikasi terdiri dari menu utama yang menyediakan pilihan input transaksi, menampilkan laporan, dan keluar dari aplikasi. Selain itu, terdapat form input transaksi untuk memasukkan data urugan serta form laporan untuk menampilkan hasil pencatatan transaksi.

## F. Skrip Program dan Penjelasannya

Aplikasi dikembangkan menggunakan bahasa pemrograman Java dengan menerapkan konsep Pemrograman Berorientasi Objek.

Berikut penjelasan setiap class nya.



```
① Dimensi.java ×
1  public class Dimensi { 4 usages
2      private double panjang; 4 usages
3      private double lebar; 4 usages
4      private double tinggi; 4 usages
5
6      public Dimensi(double panjang, double lebar, double tinggi) { 1 usage
7          this.panjang = panjang;
8          this.lebar = lebar;
9          this.tinggi = tinggi;
10     }
11
12     public double hitungVolume() { return panjang * lebar * tinggi; }
13
14
15     public void tampilkan() { 1 usage
16         System.out.println("Panjang : " + panjang);
17         System.out.println("Lebar : " + lebar);
18         System.out.println("Tinggi : " + tinggi);
19         System.out.println("Volume : " + hitungVolume() + " m³");
20     }
21
22
23     @Override
24     public String toString() {
25         return "Panjang : " + panjang + "\n" +
26                "Lebar : " + lebar + "\n" +
27                "Tinggi : " + tinggi + "\n" +
28                "Volume : " + hitungVolume() + " m³\n";
29     }

```

### Class Dimensi

Class Dimensi digunakan untuk menyimpan data ukuran muatan urugan tanah, yaitu panjang, lebar, dan tinggi. Class ini juga memiliki metode untuk menghitung volume muatan berdasarkan rumus  $\text{panjang} \times \text{lebar} \times \text{tinggi}$ .

Dengan memisahkan perhitungan volume ke dalam class Dimensi, sistem menjadi lebih terstruktur dan sesuai dengan prinsip Pemrograman Berorientasi Objek, di mana setiap class memiliki tanggung jawab yang jelas.

```
1 public class Truk { 4 usages
2     private String noPolisi; 3 usages
3     private Supplier supplier; 4 usages
4     private Dimensi dimensi; 4 usages
5
6     public Truk(String noPolisi, Supplier supplier, Dimensi dimensi) { 1 usage
7         this.noPolisi = noPolisi;
8         this.supplier = supplier;
9         this.dimensi = dimensi;
10    }
11
12    public double getVolume() { return dimensi.hitungVolume(); }
13
14    public Supplier getSupplier() { return supplier; }
15
16    public void tampilkan() { 1 usage
17        System.out.println("No Polisi : " + noPolisi);
18        supplier.tampilkan();
19        dimensi.tampilkan();
20    }
21
22    @Override
23    public String toString() {
24        return "No Polisi : " + noPolisi + "\n" +
25            "Supplier : " + supplier.getNamaSupplier() + "\n" +
26            dimensi.toString();
27    }
28}
```

## Class Truk

Class Truk digunakan untuk merepresentasikan kendaraan yang digunakan dalam proses pengangkutan urugan tanah. Setiap truk memiliki nomor polisi sebagai identitas unik serta terhubung dengan satu supplier tertentu. Selain itu, class ini juga memiliki data dimensi muatan yang digunakan untuk menghitung volume urugan.

Class Truk berperan sebagai objek perantara antara supplier dan transaksi, karena setiap transaksi selalu melibatkan satu truk tertentu.

The screenshot shows a Java code editor with the tab bar at the top containing five files: Dimensi.java, Supplier.java (which is currently selected), Transaksi.java, Truk.java, and SistemUrugan.java. The Supplier.java code is displayed in the main pane:

```
1 public class Supplier { 9 usages
2     private String namaSupplier; 3 usages
3
4     >     public Supplier(String namaSupplier) { this.namaSupplier = namaSupplier; }
5
6     >     public String getNamaSupplier() { return namaSupplier; }
7
8     >     public void tampilkan() { System.out.println("Supplier : " + namaSupplier); }
9
10    >
11
12    >
13
14    >
15
16 }
```

## Class Supplier

Class Supplier digunakan untuk merepresentasikan pihak penyedia urugan tanah. Supplier berperan sebagai pemilik atau pengelola beberapa kendaraan truk yang digunakan untuk mengangkut urugan tanah. Class ini menyimpan informasi dasar supplier, seperti nama supplier, dan menjadi penghubung antara truk dan transaksi yang terjadi.

Dalam sistem, satu objek Supplier dapat berasosiasi dengan lebih dari satu objek Truk, sehingga mencerminkan kondisi nyata di mana satu supplier dapat memiliki banyak kendaraan.

```
1 public class Transaksi { 7 usages
2     private Truk truk; 4 usages
3     private int ritase; 4 usages
4     private String tanggal; 3 usages
5
6     public Transaksi(Truk truk, int ritase, String tanggal) { 1 usage
7         this.truk = truk;
8         this.ritase = ritase;
9         this.tanggal = tanggal;
10    }
11
12    >     public double hitungTotalVolume() { return truk.getVolume() * ritase; }
13
14
15    public void tampilan() { 1 usage
16        System.out.println("Tanggal : " + tanggal);
17        truk.tampilan();
18        System.out.println("Ritase : " + ritase);
19        System.out.println("Total Volume : " + hitungTotalVolume() + " m3");
20    }
21
22
23    @Override
24    public String toString() {
25        return "Tanggal : " + tanggal + "\n" +
26               truk.toString() +
27               "Ritase : " + ritase + "\n" +
28               "Total Volume : " + hitungTotalVolume() + " m3";
29    }
30}
31
```

## Class Transaksi

Class Transaksi digunakan untuk menyimpan data setiap proses pengiriman atau pencatatan urugan tanah. Transaksi mencakup informasi truk yang digunakan, volume urugan, serta data tambahan seperti tanggal transaksi. Class ini berfungsi sebagai representasi dari satu kejadian transaksi yang terjadi dalam sistem.

Setiap objek Transaksi memiliki hubungan dengan satu objek Truk dan secara tidak langsung terhubung dengan supplier melalui truk tersebut.

```
1 import java.util.ArrayList;
2
3 public class SistemUrugan { 4 usages
4     private ArrayList<Supplier> listSupplier = new ArrayList<>(); 2 usages
5     private ArrayList<Transaksi> listTransaksi = new ArrayList<>(); 4 usages
6
7     >     public void tambahSupplier(Supplier s) { listSupplier.add(s); }
8
9
10    public Supplier cariSupplier(String nama) { 1 usage
11        for (Supplier s : listSupplier) {
12            if (s.getNamaSupplier().equalsIgnoreCase(nama)) {
13                return s;
14            }
15        }
16        return null;
17    }
18
19    >     public void tambahTransaksi(Transaksi t) { listTransaksi.add(t); }
20
21    >     public ArrayList<Transaksi> getListTransaksi() { return listTransaksi; }
22
23
24    public void tampilkanSemuaTransaksi() { no usages
25        if(listTransaksi.isEmpty()){
26            System.out.println("Belum ada transaksi!");
27            return;
28        }
29        for (Transaksi t : listTransaksi) {
30            t.tampilkan();
31            System.out.println("-----");
32        }
33    }
34
35    >
36
37    }
38}
```

## Class SistemUrugan

Class SistemUrugan berfungsi sebagai pusat pengelolaan data dalam aplikasi. Class ini bertanggung jawab untuk menyimpan, mengelola, dan menampilkan data transaksi urugan tanah. Di dalam class ini digunakan struktur data untuk menampung kumpulan objek Transaksi.

Class SistemUrugan juga menyediakan metode untuk menambah transaksi baru serta mengambil data transaksi yang akan ditampilkan dalam laporan. Dengan adanya class ini, logika sistem terpisah dari antarmuka pengguna.

Antarmuka GUI berfungsi sebagai penghubung antara pengguna dan sistem

```
1  > import ...
5
6  public class MainGUI extends JFrame {
7
8      private SistemUrugan sistem;  3 usages
9
10     public MainGUI() {  1 usage
11         sistem = new SistemUrugan(); // penyimpan transaksi
12
13         setTitle("URUGAN.CO");
14         setSize( width: 400, height: 250);
15         setLocationRelativeTo(null);
16         setDefaultCloseOperation(EXIT_ON_CLOSE);
17         setLayout(new GridLayout( rows: 4, cols: 1, hgap: 10, vgap: 10));
18
19         JLabel judul = new JLabel( text: "MENU UTAMA", SwingConstants.CENTER);
20         judul.setFont(new Font( name: "Arial", Font.BOLD, size: 18));
21
22         JButton btnInput = new JButton( text: "1. Input Transaksi");
23         JButton btnLaporan = new JButton( text: "2. Tampil Laporan");
24         JButton btnKeluar = new JButton( text: "3. Keluar");
25
26         add(judul);
27         add(btnInput);
28         add(btnLaporan);
29         add(btnKeluar);
30
31         // ACTION
32         btnInput.addActionListener( ActionEvent e -> new FormInputTransaksi(sistem));
33         btnLaporan.addActionListener( ActionEvent e -> new FormLaporan(sistem));
34         btnKeluar.addActionListener( ActionEvent e -> System.exit( status: 0));
35
36         setVisible(true);
37     }
38
39     public static void main(String[] args) {
40         new MainGUI();
41     }
42 }
```

## Class MainGUI

Class MainGUI merupakan tampilan utama aplikasi berbasis GUI. Class ini menampilkan menu awal yang berisi pilihan untuk melakukan input transaksi, menampilkan laporan, atau keluar dari aplikasi. Class ini berfungsi sebagai penghubung awal antara pengguna dan sistem.

MainGUI tidak menyimpan logika bisnis, melainkan hanya mengatur navigasi antar form dan memanggil class lain sesuai dengan pilihan pengguna.

```
>MainGUI.java  FormInputTransaksi.java
```

```
1 import javax.swing.*;
2 
3 public class FormInputTransaksi extends JFrame { 1 usage
4 
5     public FormInputTransaksi(SistemUrugan sistem) { 1 usage
6         setTitle("Input Transaksi");
7         setSize( width: 400, height: 350);
8         setLocationRelativeTo(null);
9         setLayout(null);
10 
11         JLabel lNoPol = new JLabel( text: "No Polisi:");
12         JTextField tfNoPol = new JTextField();
13 
14         JLabel lSupplier = new JLabel( text: "Supplier:");
15         JTextField tfSupplier = new JTextField();
16 
17         JLabel lP = new JLabel( text: "Panjang:");
18         JTextField tfP = new JTextField();
19 
20         JLabel lL = new JLabel( text: "Lebar:");
21         JTextField tfL = new JTextField();
22 
23         JLabel lT = new JLabel( text: "Tinggi:");
24         JTextField tfT = new JTextField();
25 
26         JLabel lRitase = new JLabel( text: "Ritase:");
27         JTextField tfRitase = new JTextField();
28 
29         JButton btnSimpan = new JButton( text: "Simpan");
30 
31     // POSISI
32         lNoPol.setBounds( x: 20, y: 20, width: 100, height: 25);
33         tfNoPol.setBounds( x: 130, y: 20, width: 200, height: 25);
34 
35         lSupplier.setBounds( x: 20, y: 55, width: 100, height: 25);
36         tfSupplier.setBounds( x: 130, y: 55, width: 200, height: 25);
37 
38         lP.setBounds( x: 20, y: 90, width: 100, height: 25);
39         tfP.setBounds( x: 130, y: 90, width: 200, height: 25);
```

```
MainGUI.java  FormInputTransaksi.java
```

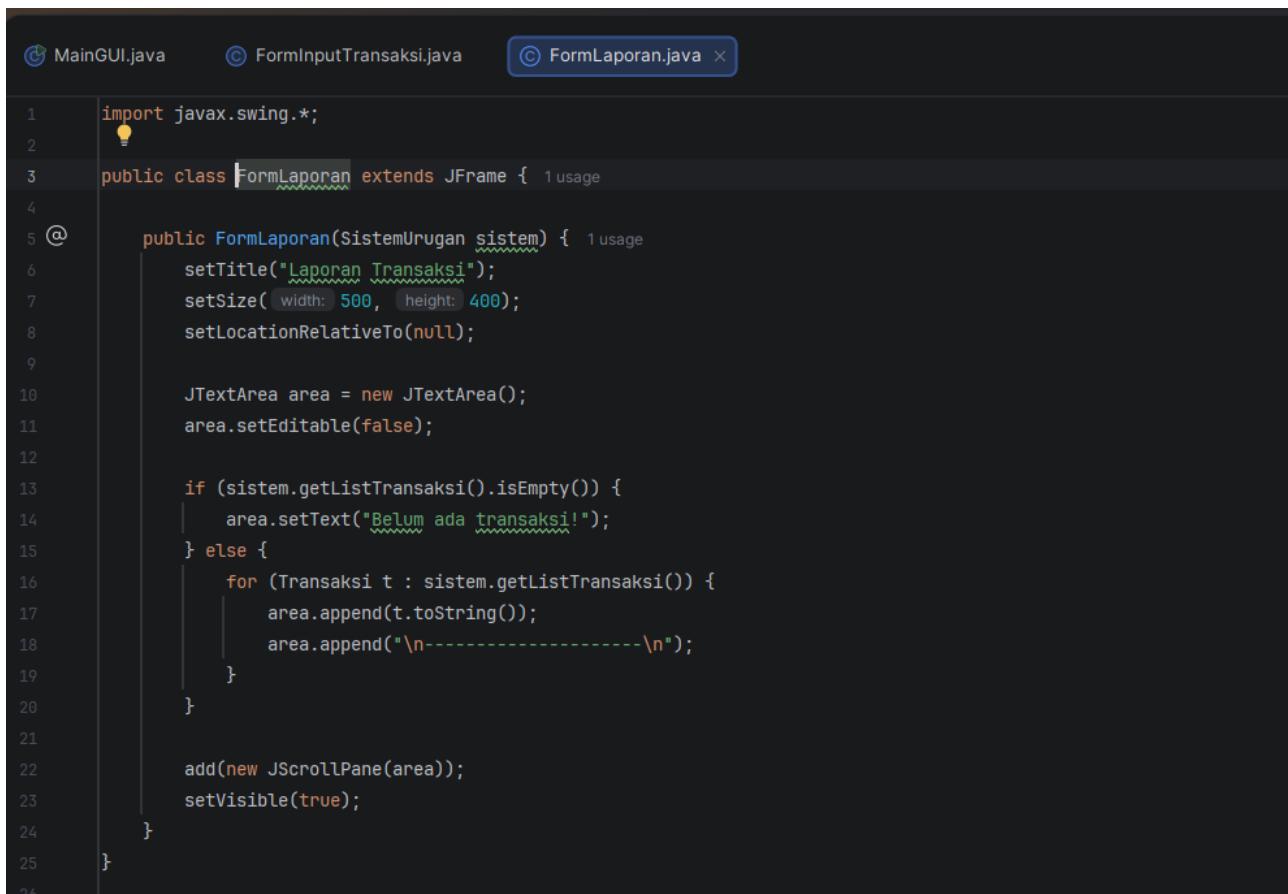
```
1 public class FormInputTransaksi extends JFrame { 1 usage
2     public FormInputTransaksi(SistemUrugan sistem) { 1 usage
3         tfP.setBounds( x: 130, y: 90, width: 200, height: 25);
4 
5         lL.setBounds( x: 20, y: 125, width: 100, height: 25);
6         tfL.setBounds( x: 130, y: 125, width: 200, height: 25);
6 
7         lT.setBounds( x: 20, y: 160, width: 100, height: 25);
8         tfT.setBounds( x: 130, y: 160, width: 200, height: 25);
8 
9         lRitase.setBounds( x: 20, y: 195, width: 100, height: 25);
10        tfRitase.setBounds( x: 130, y: 195, width: 200, height: 25);
10 
11        btnSimpan.setBounds( x: 130, y: 240, width: 100, height: 30);
11 
12        add(lNoPol); add(tfNoPol);
13        add(lSupplier); add(tfSupplier);
14        add(lP); add(tfP);
15        add(lL); add(tfL);
16        add(lT); add(tfT);
17        add(lRitase); add(tfRitase);
18        add(btnSimpan);
18 
19        btnSimpan.addActionListener( ActionEvent e -> {
20            try {
21                String noPol = tfNoPol.getText();
22                String namaSupplier = tfSupplier.getText();
23                double p = Double.parseDouble(tfP.getText());
24                double l = Double.parseDouble(tfL.getText());
25                double t = Double.parseDouble(tfT.getText());
26                int ritase = Integer.parseInt(tfRitase.getText());
26 
27                // Cari supplier berdasarkan nama
28                Supplier supplier = sistem.cariSupplier(namaSupplier);
```

```
3  public class FormInputTransaksi extends JFrame { 1 usage
5      public FormInputTransaksi(SistemUrugan sistem) { 1 usage
6          btnSimpan.addActionListener( ActionEvent e -> {
7
8              Supplier supplier = sistem.cariSupplier(namaSupplier);
9
10             // Jika supplier tidak ada, tanya apakah ingin buat baru
11             if (supplier == null) {
12                 int result = JOptionPane.showConfirmDialog( parentComponent: this,
13                     message: "Supplier '" + namaSupplier + "' tidak ditemukan.\nApakah ingin membuat supplier baru?",
14                     title: "Supplier Baru",
15                     JOptionPane.YES_NO_OPTION);
16
17                 if (result == JOptionPane.YES_OPTION) {
18                     // Buat supplier baru
19                     supplier = new Supplier(namaSupplier);
20                     sistem.tambahSupplier(supplier);
21                 } else {
22                     return;
23                 }
24             }
25
26             // Buat dimensi dan truk
27             Dimensi dimensi = new Dimensi(p, l, t);
28             Truk truk = new Truk(noPol, supplier, dimensi);
29
30             // Buat dan tambah transaksi
31             Transaksi transaksi = new Transaksi(truk, ritase, java.time.LocalDate.now().toString());
32             sistem.tambahTransaksi(transaksi);
33
34             JOptionPane.showMessageDialog( parentComponent: this, message: "Transaksi berhasil disimpan!");
35             dispose();
36         } catch (NumberFormatException ex) {
37             JOptionPane.showMessageDialog( parentComponent: this, message: "Input tidak valid! Periksa kembali.", title: "Error", JOptionPane.ERROR_MESSAGE);
38         }
39     });
40
41     setVisible(true);
42 }
```

## Class FormInputTransaksi

Class FormInputTransaksi digunakan sebagai antarmuka untuk memasukkan data transaksi urugan tanah. Pada form ini, pengguna dapat memasukkan nomor polisi kendaraan, nama supplier, serta dimensi muatan. Data yang dimasukkan kemudian diteruskan ke class SistemUrugan untuk diproses dan disimpan.

Class ini berfokus pada pengambilan input dari pengguna dan tidak melakukan proses perhitungan secara langsung, sehingga menjaga pemisahan antara antarmuka dan logika sistem.



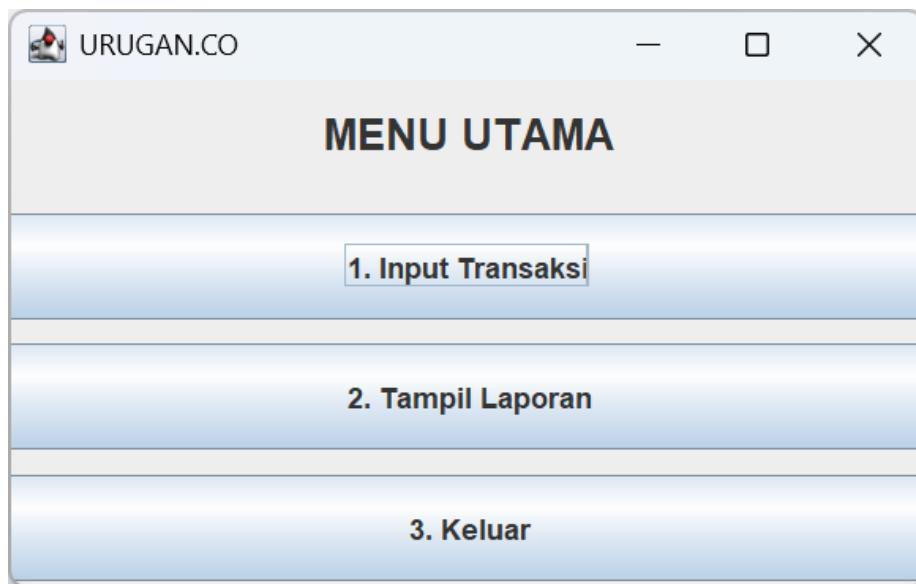
```
1 import javax.swing.*;
2 
3 public class FormLaporan extends JFrame { 1 usage
4 
5     @
6         public FormLaporan(SistemUrugan sistem) { 1 usage
7             setTitle("Laporan Transaksi");
8             setSize( width: 500, height: 400);
9             setLocationRelativeTo(null);
10 
11             JTextArea area = new JTextArea();
12             area.setEditable(false);
13 
14             if (sistem.getListTransaksi().isEmpty()) {
15                 area.setText("Belum ada transaksi!");
16             } else {
17                 for (Transaksi t : sistem.getListTransaksi()) {
18                     area.append(t.toString());
19                     area.append("\n-----\n");
20                 }
21             }
22             add(new JScrollPane(area));
23             setVisible(true);
24         }
25     }
26 }
```

## Class FormLaporan

Class FormLaporan digunakan untuk menampilkan laporan transaksi urugan tanah yang telah tersimpan dalam sistem. Data laporan diambil dari class SistemUrugan dan ditampilkan dalam bentuk teks melalui antarmuka GUI.

Class ini berfungsi sebagai media penyajian informasi kepada pengguna tanpa mengubah atau memproses ulang data transaksi.

## G. Penjelasan Screenshot Tampilan Aplikasi



### Tampilan Menu Utama

Disini user bisa memilih ingin memasukkan transaksi atau melihat laporan transaksi.

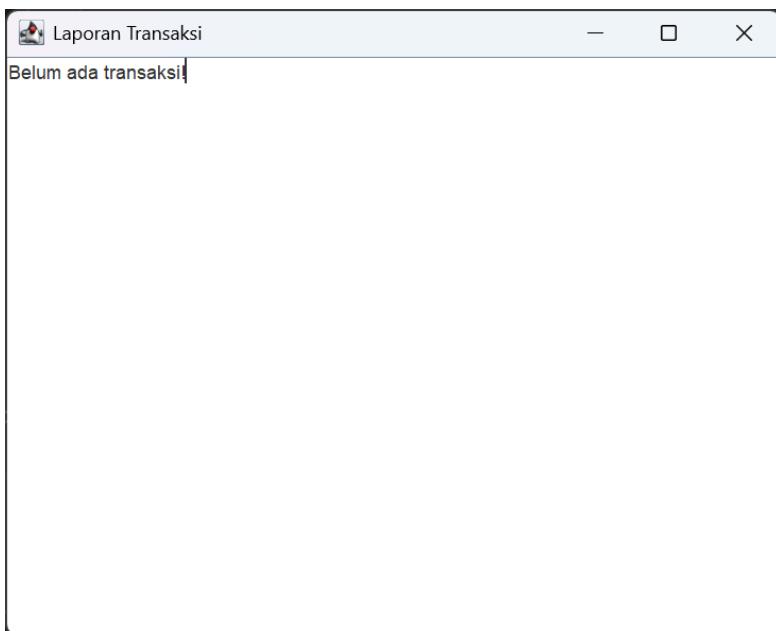
The screenshot shows a window titled "Input Transaksi" with fields for inputting truck dimensions and supplier information. The fields are:

No Polisi:	AB 7890 TA
Supplier:	PT Uruganjaya
Panjang:	90
Lebar:	80
Tinggi:	100
Ritase:	5

At the bottom is a blue "Simpan" button.

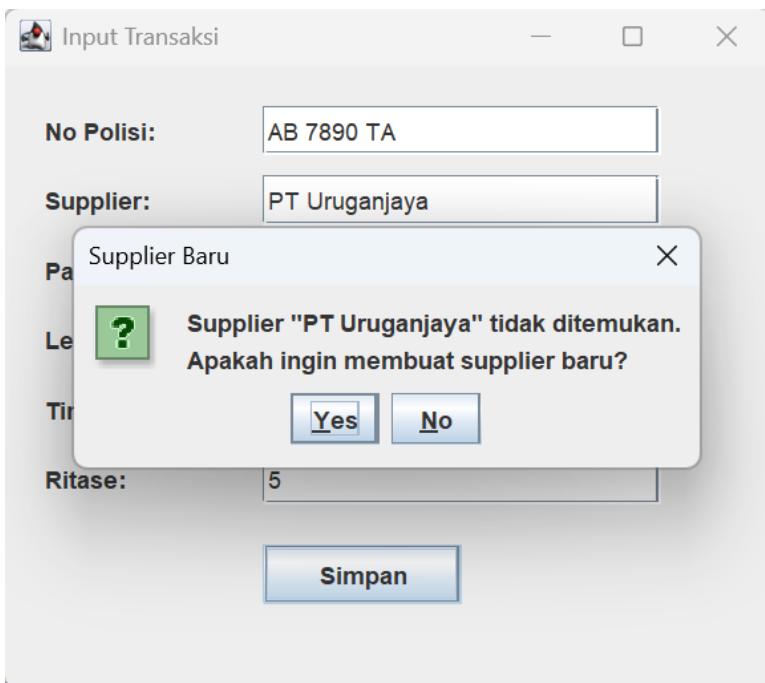
### Tampilan Input

Pada tampilan input terdapat beberapa inputan seperti No Polisi, Nama Supplier, Dimensi dari bak truk, dan ritase dari truk itu sendiri.



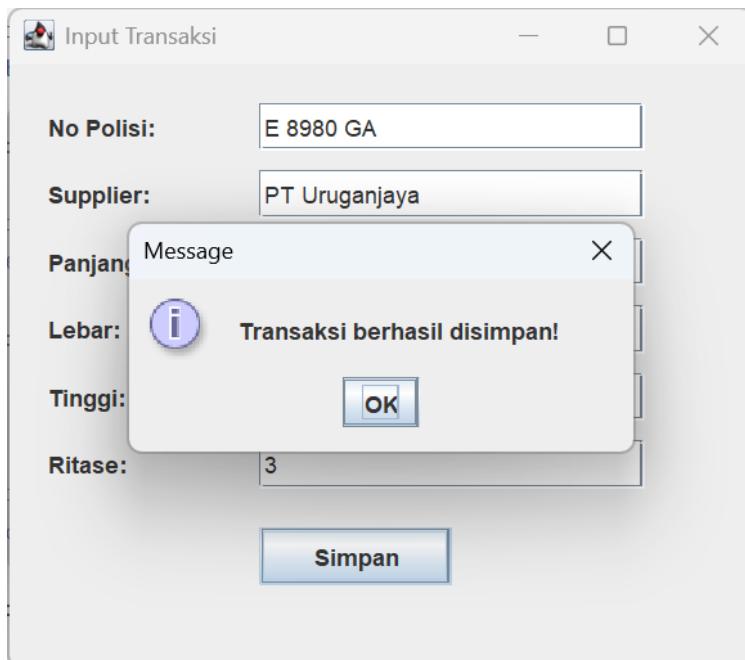
### Tampilan Laporan Transaksi Jika Belum Ada Data

Jika belum ada transaksi yang diinputkan, maka akan menampilkan tulisan "Belum ada transaksi!" seperti diatas



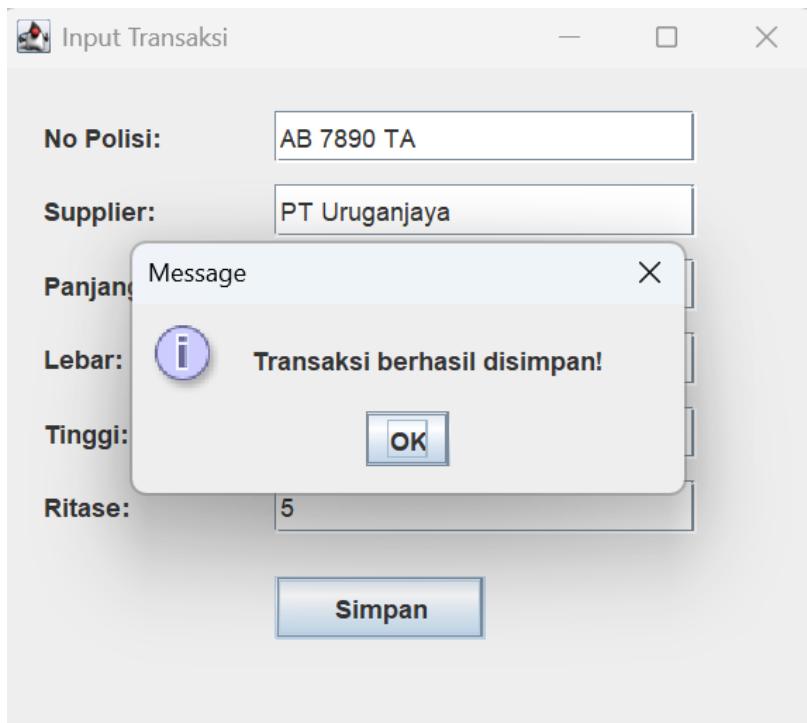
### Tampilan Jika Supplier Belum Ada di Database

Setelah user memasukkan inputan yang tersedia diatas, dan terjadi pengecekan pada database (disini masih menggunakan struktur data bawaan program) jika nama supplier sudah belum tersedia di database, maka akan muncul notifikasi diatas, dan dibuatlah database baru dengan nama yang sudah dimasukkan.



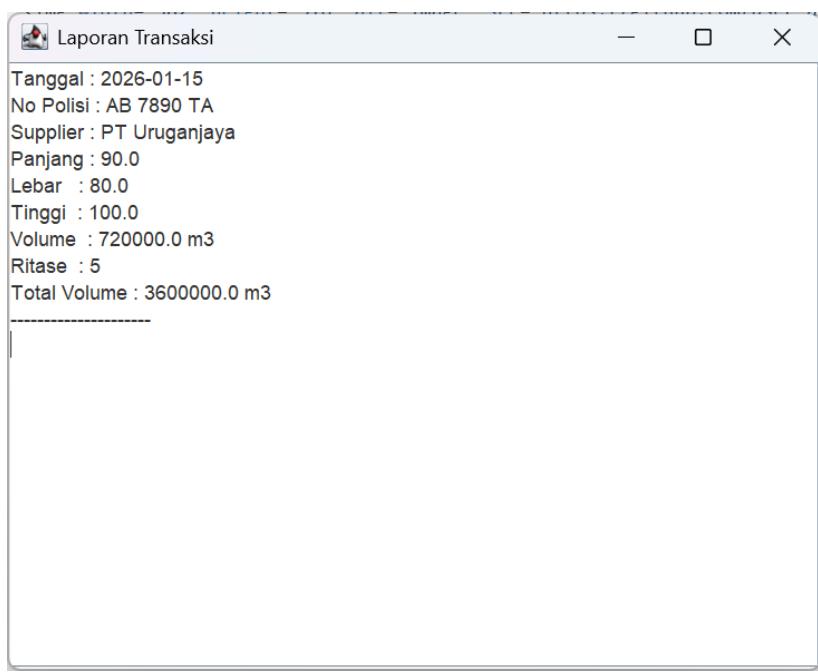
### Tampilan Jika Supplier Sudah Ada di Database

Jika nama supplier sudah ada di database, maka tidak ada pop up input data baru seperti yang sebelumnya, karena sudah ada di database.



### Tampilan Transaksi Berhasil di Simpan

Pop up tampilan transaksi berhasil disimpan, dan data akan secara otomatis masuk ke Laporan Transaksi



### **Tampilan Laporan Transaksi**

Tampilan laporan transaksi sederhana, dan disini untuk tanggalnya di generate secara otomatis sesuai dengan tanggal sekarang, agar user tidak kesusahan dalam menginputkan transaksi.

## H. Penjelasan Screenshot Status Unggah Skrip di Github

Commits on Jan 15, 2026			
Delete Truk.java	(Artmzyyy authored 4 days ago)	Verified	d5de7f3
Delete Transaksi.java	(Artmzyyy authored 4 days ago)	Verified	926e1e1
Delete Supplier.java	(Artmzyyy authored 4 days ago)	Verified	d74e164
Delete SistemUrugan.java	(Artmzyyy authored 4 days ago)	Verified	8354123
Delete MainGUI.java	(Artmzyyy authored 4 days ago)	Verified	b319a0c
Delete FormLaporan.java	(Artmzyyy authored 4 days ago)	Verified	147ca13
Delete FormInputTransaksi.java	(Artmzyyy authored 4 days ago)	Verified	0b4cf47
Delete Dimensi.java	(Artmzyyy authored 4 days ago)	Verified	6448f2b
Add files via upload	(Artmzyyy authored 4 days ago)	Verified	f1fec41
Fix repository link in README	(Artmzyyy authored 4 days ago)	Verified	969dfbd
Update README with new images and project timeline	(Artmzyyy authored 4 days ago)	Verified	322a294
Revise README with project details and specifications	(Artmzyyy authored 4 days ago)	Verified	d88b10a
Add Final Project	(Artmzyyy authored 4 days ago)	Verified	cfe3ee

Screenshot status unggah skrip pada GitHub menunjukkan bahwa seluruh file program telah berhasil diunggah ke dalam repository pada tahap akhir penggerjaan proyek. Proses pengunggahan dilakukan setelah seluruh fitur utama aplikasi selesai diimplementasikan dan diuji, sehingga kode yang diunggah merupakan versi final dari proyek.

Tidak terdapat dokumentasi unggah secara bertahap karena proses pengembangan, penggerjaan dilakukan secara lokal oleh masing-masing anggota kelompok untuk fokus pada penyelesaian fungsi dan struktur program. Setelah aplikasi dinyatakan berjalan dengan baik dan sesuai spesifikasi, seluruh skrip kemudian dikumpulkan dan diunggah ke GitHub sebagai bentuk dokumentasi akhir dan bukti penyelesaian proyek.

Meskipun dokumentasi commit tidak dilakukan secara bertahap, penggunaan GitHub tetap berfungsi sebagai media penyimpanan kode sumber, kolaborasi kelompok, serta arsip proyek yang dapat diakses kembali di kemudian hari.

## **I. Analisis Pengerjaan Proyek**

Pengerjaan proyek dilakukan dalam waktu 3 minggu. Dari sisi waktu, proyek dapat diselesaikan tepat waktu dengan pembagian tugas antar anggota kelompok. Dari sisi ketercapaian spesifikasi, sebagian besar fitur utama berhasil diimplementasikan sesuai perencanaan. Dari sisi biaya, proyek tidak memerlukan biaya tambahan karena menggunakan perangkat lunak gratis. Kendala utama yang dihadapi adalah perancangan struktur class dan penerapan konsep PBO secara tepat. Ke depan, aplikasi ini masih dapat dikembangkan dengan menambahkan fitur database, pencetakan laporan, serta peningkatan tampilan antarmuka.