



MANUAL BOOK

Aplikasi Desktop Sistem Inventaris Toko

Disusun oleh:

Reykal Hizbullah 51422420

Daniela Mendytha R.D.D 50422382

Fikriya Sabila Yusriyani 50422577

Universitas Gunadarma

Mata Praktikum Rekayasa Perangkat Lunak 2

Daftar Isi

1	Pendahuluan	3
1.1	Deskripsi Aplikasi	3
1.2	Teknologi yang Digunakan	3
2	Instalasi dan Konfigurasi	3
2.1	Perangkat Lunak yang Dibutuhkan	3
2.2	Langkah-langkah Instalasi	3
3	Struktur dan Penjelasan Kode	4
3.1	Barang.java (Entity/Model)	4
3.2	BarangRepository.java (Repository)	4
3.3	BarangService.java (Service)	5
3.4	InventarisView.java (View/GUI)	5
3.5	AplikasiInventaris.java (Main Class)	6
4	Panduan Penggunaan	6
4.1	Tampilan Awal (Read)	6
4.2	Menambah Data Barang (Create)	7
4.3	Mengubah Data Barang (Update)	8
4.4	Menghapus Data Barang (Delete)	9

1 Pendahuluan

1.1 Deskripsi Aplikasi

Aplikasi Sistem Inventaris Toko adalah sebuah program desktop yang dirancang untuk membantu pemilik toko dalam mengelola data barang. Aplikasi ini dibangun menggunakan Java Swing untuk antarmuka pengguna (GUI) dan diintegrasikan dengan framework Spring Boot serta ORM Hibernate untuk manajemen data. Fungsionalitas utama aplikasi ini mencakup operasi CRUD (Create, Read, Update, Delete) untuk data barang.

1.2 Teknologi yang Digunakan

- **Bahasa Pemrograman:** Java (JDK 17)
- **Framework:** Spring Boot & Spring Data JPA
- **Antarmuka Pengguna (GUI):** Java Swing
- **Database:** MySQL
- **Build Tool:** Apache Maven
- **IDE:** Apache NetBeans

2 Instalasi dan Konfigurasi

2.1 Perangkat Lunak yang Dibutuhkan

1. Java Development Kit (JDK) versi 17 atau lebih baru.
2. Apache NetBeans IDE 12 atau lebih baru.
3. XAMPP atau server MySQL mandiri.

2.2 Langkah-langkah Instalasi

1. **Impor Database:** Buka phpMyAdmin dan buat database baru. Beri nama database tersebut `proyek_rpl2_inventaris`. Setelah itu, impor file `.sql` yang disediakan.
2. **Buka Proyek di NetBeans:** Ekstrak file ZIP proyek, lalu buka folder proyek melalui menu **File > Open Project...**
3. **Konfigurasi Koneksi (Opsional):** Jika perlu, sesuaikan konfigurasi database di file `application.properties`.

```
# Konfigurasi Koneksi Database
spring.datasource.url=jdbc:mysql://localhost:3306/
    proyek_rpl2_inventaris?useSSL=false&serverTimezone=UTC
spring.datasource.username=root
```

```
spring.datasource.password=
```

Listing 1: Isi file application.properties

4. **Build dan Jalankan Proyek:** Klik kanan pada proyek, pilih **Clean and Build**. Setelah sukses, klik kanan pada file `AplikasiInventaris.java` dan pilih **Run File**.

3 Struktur dan Penjelasan Kode

Program ini dibangun mengikuti arsitektur berlapis yang memisahkan tanggung jawab antara data, logika, dan tampilan.

3.1 Barang.java (Entity/Model)

Class ini adalah **blueprint** untuk data barang. Ia merepresentasikan struktur tabel `barang` di database. Anotasi JPA seperti `@Entity` dan `@Id` digunakan untuk memetakan class ini ke tabel database.

```
@Entity
@Table(name = "barang")
public class Barang {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String namaBarang;
    private int stok;
    private double harga;

    // Constructor, Getters, and Setters...
}
```

Listing 2: Cuplikan Kode Barang.java

3.2 BarangRepository.java (Repository)

Ini adalah **gerbang akses data**. Dengan hanya membuat sebuah interface yang **extends** `JpaRepository`, Spring Data JPA secara otomatis menyediakan implementasi untuk semua method CRUD dasar (`save()`, `findAll()`, `findById()`, `deleteById()`, dll). Kita tidak perlu menulis query SQL sama sekali.

```
@Repository
public interface BarangRepository extends JpaRepository<Barang, Long> {
    // Spring Data JPA akan otomatis menyediakan method CRUD
}
```

Listing 3: Kode BarangRepository.java

3.3 BarangService.java (Service)

Class ini adalah lapisan **logika bisnis**. Ia bertindak sebagai jembatan antara View (tampilan) dan Repository (data). Anotasi `@Service` menandainya sebagai komponen Spring, dan `@Autowired` secara otomatis menyuntikkan `BarangRepository` ke dalamnya.

```
@Service
public class BarangService {
    @Autowired
    private BarangRepository repository;

    public List<Barang> getAllBarang() {
        return repository.findAll();
    }

    public Barang addBarang(Barang barang) {
        return repository.save(barang);
    }
    // ... method lainnya
}
```

Listing 4: Cuplikan Kode BarangService.java

3.4 InventarisView.java (View/GUI)

Ini adalah **antarmuka pengguna** yang dibuat menggunakan Java Swing (`JFrame`). Class ini bertanggung jawab untuk menampilkan data dan menangkap input dari pengguna. Pentingnya, class ini tidak pernah berinteraksi langsung dengan database, melainkan hanya memanggil method dari `BarangService` yang telah disuntikkan melalui constructor.

```
public class InventarisView extends javax.swing.JFrame {
    private final BarangService barangService;

    public InventarisView(BarangService service) {
        this.barangService = service;
        initComponents();
        loadTableData();
    }

    private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {
        // ... logika mengambil data dari form ...
        Barang barang = new Barang(nama, stok, harga);
        barangService.addBarang(barang); // Memanggil service
        loadTableData(); // Refresh tabel
    }
    // ...
}
```

Listing 5: Cuplikan Kode InventarisView.java

3.5 AplikasiInventaris.java (Main Class)

Ini adalah **titik masuk (entry point)** utama aplikasi. Anotasi `@SpringBootApplication` memulai seluruh mesin Spring Boot. Class ini mengimplementasikan `CommandLineRunner`, yang method `run()`-nya akan dieksekusi setelah Spring siap. Di sinilah `InventarisView` dibuat dan ditampilkan.

```
@SpringBootApplication
public class AplikasiInventaris implements CommandLineRunner {

    @Autowired
    private BarangService barangService;

    public static void main(String[] args) {
        new SpringApplicationBuilder(AplikasiInventaris.class)
            .headless(false)
            .run(args);
    }

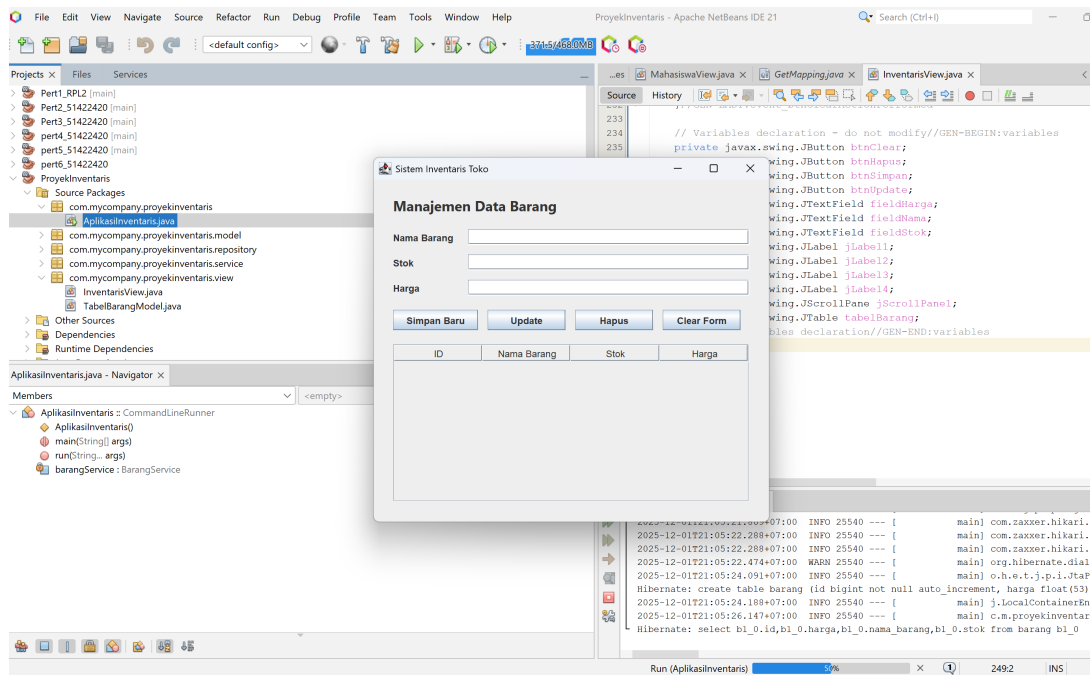
    @Override
    public void run(String... args) throws Exception {
        EventQueue.invokeLater(() -> {
            InventarisView view = new InventarisView(barangService);
            view.setVisible(true);
        });
    }
}
```

Listing 6: Cuplikan Kode AplikasiInventaris.java

4 Panduan Penggunaan

4.1 Tampilan Awal (Read)

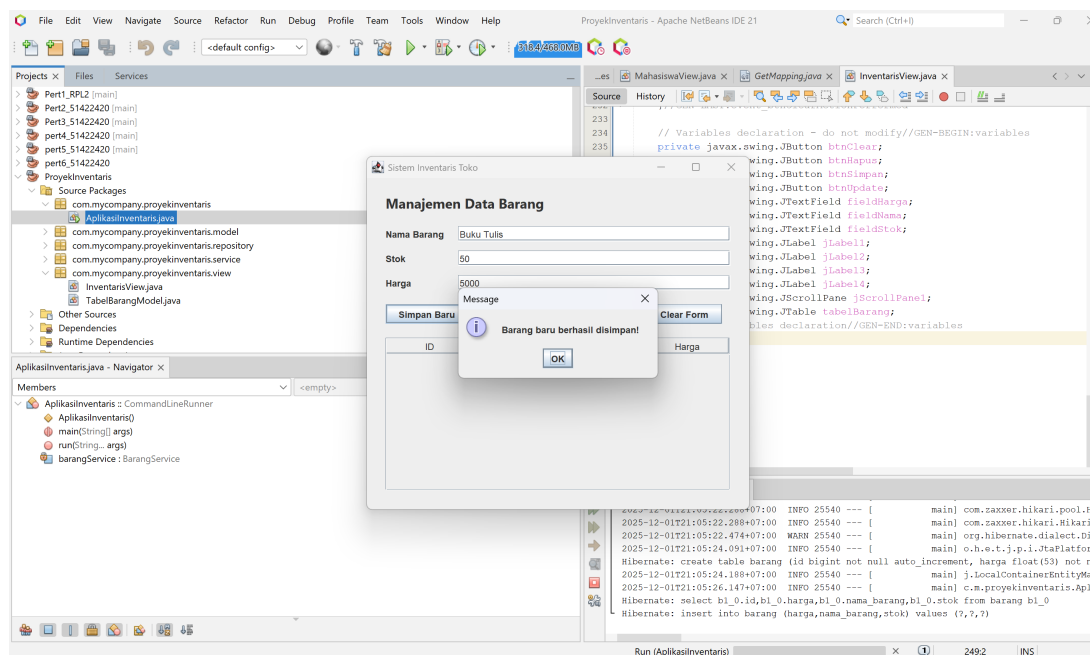
Saat aplikasi pertama kali dijalankan, jendela utama akan muncul dengan tabel data yang masih kosong.



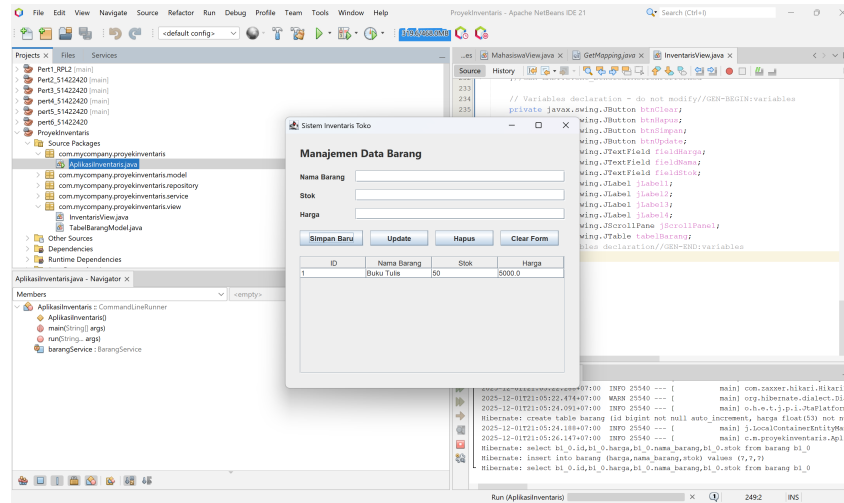
Gambar 1: Tampilan Awal Aplikasi dengan Tabel Kosong

4.2 Menambah Data Barang (Create)

1. Isi form "Nama Barang", "Stok", dan "Harga".
2. Klik tombol **Simpan Baru**.
3. Sebuah pesan konfirmasi akan muncul.
4. Setelah menekan OK, data baru akan tampil di tabel.



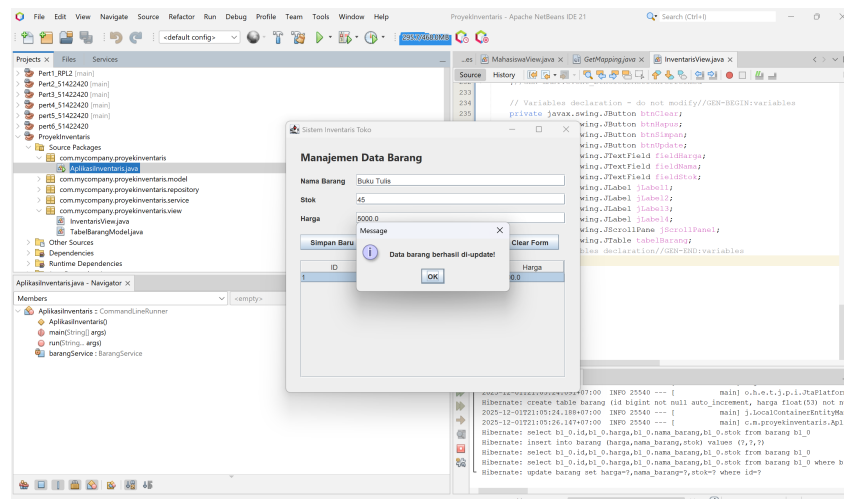
Gambar 2: Mengisi Form untuk Menambah Data Baru



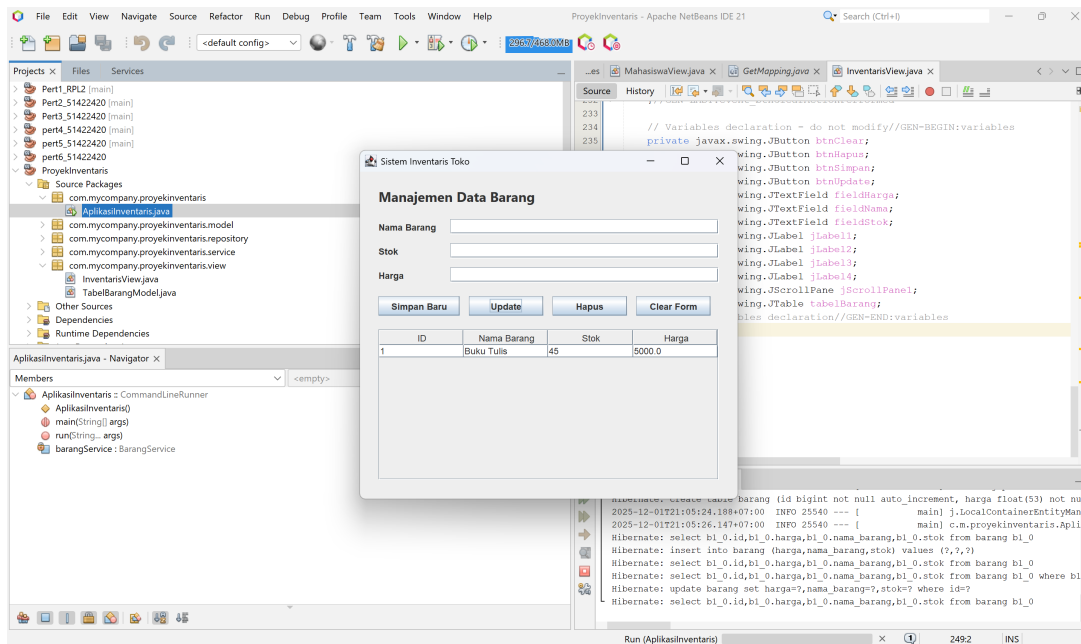
Gambar 3: Pesan Konfirmasi Setelah Data Berhasil Disimpan

4.3 Mengubah Data Barang (Update)

1. Klik pada baris data di tabel yang ingin diubah.
2. Ubah informasi yang diinginkan di form.
3. Klik tombol **Update**.
4. Sebuah pesan konfirmasi akan muncul.
5. Data di tabel akan ter-refresh dengan informasi terbaru.



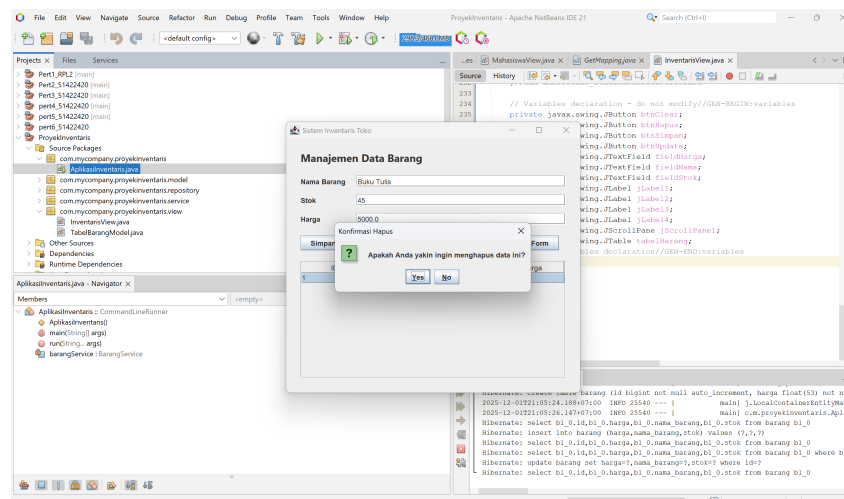
Gambar 4: Pesan Konfirmasi Setelah Data Berhasil Di-update



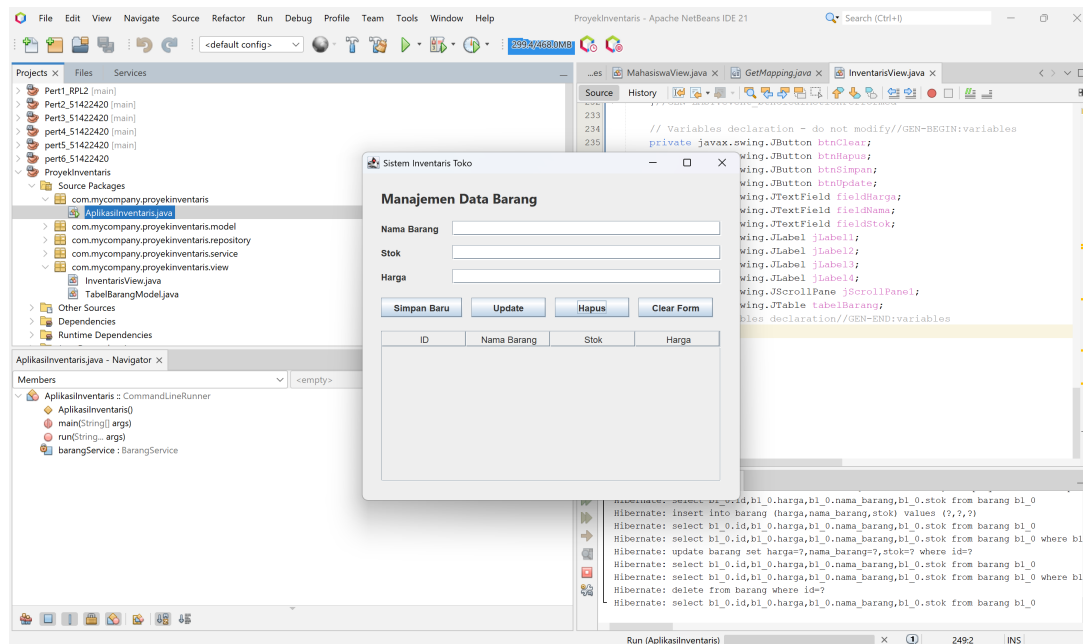
Gambar 5: Tampilan Tabel Setelah Data Berhasil Di-update

4.4 Menghapus Data Barang (Delete)

1. Klik pada baris data di tabel yang ingin dihapus.
2. Klik tombol **Hapus**.
3. Sebuah kotak dialog konfirmasi akan muncul. Klik **Yes** untuk melanjutkan.
4. Data akan terhapus dari database dan hilang dari tampilan tabel.



Gambar 6: Dialog Konfirmasi Sebelum Menghapus Data



Gambar 7: Tampilan Tabel Setelah Data Berhasil Dihapus