

# Perangkat Lunak Manajemen Museum Dokumentasi Teknis

Tim Pengembang

June 9, 2025

## Contents

# 1 Pendahuluan

Dokumen ini menyediakan dokumentasi komprehensif untuk Perangkat Lunak Manajemen Museum, termasuk arsitektur sistem, prosedur pemeliharaan, dan panduan penggunaan. Dokumentasi ini mencakup setiap file dalam proyek dan memberikan informasi rinci tentang tujuan, fungsionalitas, dan penggunaannya.

## 2 Tinjauan Proyek

Perangkat Lunak Manajemen Museum adalah aplikasi desktop berbasis Java yang dibuat menggunakan Swing untuk antarmuka pengguna. Sistem ini dirancang untuk mengelola operasi museum, termasuk otentikasi pengguna, fungsi administratif, dan fitur khusus pengguna.

## 3 Arsitektur Sistem

### 3.1 Arsitektur Frontend

Frontend dibangun menggunakan Java Swing, memberikan pengalaman aplikasi desktop yang kaya.

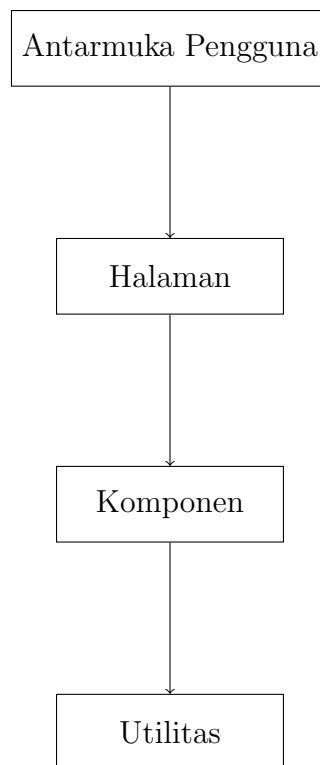


Figure 1: Lapisan Arsitektur Frontend

#### 3.1.1 Kerangka Kerja UI

- Tumpukan Teknologi

- Java Swing untuk komponen UI
- Komponen UI kustom di direktori **Components/**
- Arsitektur berbasis peristiwa (event-driven)
- Implementasi pola MVC

- **Komponen Kunci**

- Kartu kustom untuk tampilan item
- Sistem navigasi
- Komponen formulir
- Widget tampilan data

- **Organisasi UI**

- Antarmuka berbasis peran (Admin/Pengguna)
- Tata letak responsif
- Gaya yang konsisten
- Fitur aksesibilitas

### 3.1.2 Lapisan Antarmuka Pengguna

- **Lapisan Presentasi**

- Direktori **Pages/** berisi semua frame UI
- Direktori **Components/** untuk elemen UI yang dapat digunakan kembali
- Gaya dan tema kustom
- Penangan dan pendengar peristiwa (event handlers and listeners)

- **Model Tampilan (View Models)**

- Implementasi pengikatan data (data binding)
- Manajemen status (state management)
- Mekanisme pembaruan UI
- Logika validasi

## 3.2 Arsitektur Middleware

Lapisan middleware menangani logika bisnis, pemrosesan data, dan komunikasi antara frontend dan backend.



Figure 2: Alur Komunikasi Middleware

### 3.2.1 Lapisan Logika Bisnis

- **Komponen Inti**
  - Components/Utilities/ untuk logika bisnis
  - Validasi dan pemrosesan data
  - Implementasi aturan bisnis
  - Manajemen status
- **Lapisan Layanan (Service Layer)**
  - Layanan otentikasi
  - Layanan pemrosesan data
  - Layanan penanganan gambar
  - Layanan pencarian dan filter
- **Lapisan Akses Data (Data Access Layer)**
  - Manajemen koneksi basis data
  - Eksekusi kueri
  - Penanganan transaksi
  - Pemetaan data

### 3.2.2 Lapisan Integrasi

- **Transformasi Data**
  - Pemetaan objek-relasional
  - Konversi format data
  - Aturan validasi
  - Penanganan kesalahan
- **Komunikasi**
  - Komunikasi frontend-backend
  - Propagasi peristiwa
  - Sinkronisasi status
  - Propagasi kesalahan

## 3.3 Arsitektur Backend

Backend dibangun di atas basis data MySQL dengan utilitas pemrosesan data kustom.

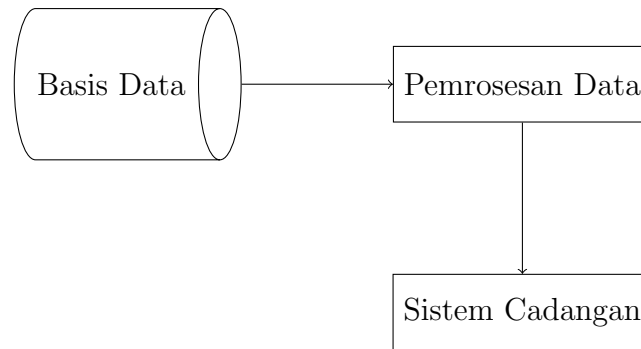


Figure 3: Komponen Arsitektur Backend

### 3.3.1 Lapisan Basis Data

- **Desain Basis Data**

- Skema basis data MySQL
- Hubungan antar tabel
- Optimalisasi indeks
- Manajemen batasan (constraint)

- **Manajemen Data**

- Operasi CRUD
- Manajemen transaksi
- Pencadangan dan pemulihan
- Integritas data

- **Optimalisasi Kueri**

- Kueri terindeks
- Prosedur tersimpan (stored procedures)
- Caching kueri
- Pemantauan kinerja

### 3.3.2 Pemrosesan Data

- **Alat Data**

- `get_data.py` untuk pemrosesan data
- Skrip validasi data
- Utilitas impor/ekspor
- Alat transformasi data

- **Sistem Cadangan**

- Pencadangan otomatis

- Pemulihan data
- Kontrol versi
- Manajemen arsip

### 3.3.3 Lapisan Keamanan

- **Otentikasi**

- Otentikasi pengguna
- Kontrol akses berbasis peran
- Manajemen sesi
- Keamanan kata sandi

- **Perlindungan Data**

- Enkripsi data
- Komunikasi aman
- Pencatatan akses (logging)
- Pemantauan keamanan

## 4 Sistem Koneksi dan Alur Data

### 4.1 Arsitektur Koneksi Basis Data

Aplikasi ini mengimplementasikan sistem koneksi basis data terpusat melalui kelas utilitas `DatabaseConnection`.



Figure 4: Arsitektur Koneksi Basis Data

### 4.2 Contoh Alur Data

#### 4.2.1 Alur Tampilan Koleksi



Figure 5: Alur Data Tampilan Koleksi

1. `MuseumCollection.java` memulai permintaan data
2. Memanggil `ItemQueries.getCollectionItems()`
3. `DatabaseConnection` mengeksekusi kueri
4. `ResultSet` dikembalikan ke `MuseumCollection`
5. Data diproses oleh `ContainerPopulator`
6. Ditampilkan melalui komponen `ItemCard`

#### 4.2.2 Alur Manajemen Item

1. `AddItem.java` mengumpulkan data item
2. Memvalidasi melalui `validateItemInput()`
3. Memanggil `ItemQueries.insertItem()`
4. `DatabaseConnection` mengeksekusi insert
5. Transaksi di-commit
6. UI diperbarui melalui `ContainerPopulator`

#### 4.2.3 Implementasi Pencarian

1. `SearchBar.java` menangkap input pengguna
2. Memicu `ItemQueries.searchItems()`
3. `DatabaseConnection` mengeksekusi kueri pencarian
4. Hasil diproses oleh `ContainerPopulator`
5. Ditampilkan di tampilan `MuseumCollection`

### 4.3 Manajemen Koneksi



Figure 6: Siklus Hidup Koneksi

#### 4.3.1 Penanganan Sumber Daya

- **Siklus Hidup Koneksi**

- Koneksi diperoleh dari pool
- Digunakan untuk eksekusi kueri
- Dikembalikan ke pool setelah digunakan
- Pembersihan otomatis saat aplikasi keluar

- **Pemulihan Kesalahan**

- Penanganan waktu habis koneksi (timeout)
- Upaya koneksi ulang otomatis
- Pencatatan dan pelaporan kesalahan
- Sistem notifikasi pengguna

#### 4.3.2 Optimalisasi Kinerja

- **Connection Pooling**

- Manajemen ukuran pool tetap
- Penggunaan kembali koneksi
- Pembersihan koneksi idle
- Pemantauan pool

- **Optimalisasi Kueri**

- Caching prepared statement
- Dukungan operasi batch
- Streaming ResultSet
- Pengaturan waktu habis koneksi

### 4.4 Pola Akses Data

#### 4.4.1 Operasi Baca

*// Contoh: Mengambil item koleksi*

```
public ResultSet getCollectionItems() {  
    String query = ItemQueries.SELECT_ALL_ITEMS;  
    return DatabaseConnection.executeQuery(query);  
}
```

*// Contoh: Mencari item*

```
public ResultSet searchItems(String searchTerm) {  
    String query = ItemQueries.SEARCH_ITEMS;  
    PreparedStatement stmt = DatabaseConnection.prepareStatement(query);  
    stmt.setString(1, "%" + searchTerm + "%");  
    return stmt.executeQuery();  
}
```



#### 4.4.2 Operasi Tulis

```
// Contoh: Menambahkan item baru
public boolean addItem(Item item) {
    String query = ItemQueries.INSERT_ITEM;
    PreparedStatement stmt = DatabaseConnection.prepareStatement(query);
    stmt.setString(1, item.getName());
    stmt.setString(2, item.getDescription());
    // ... atur parameter lain
    return stmt.executeUpdate() > 0;
}

// Contoh: Memperbarui item
public boolean updateItem(Item item) {
    String query = ItemQueries.UPDATE_ITEM;
    PreparedStatement stmt = DatabaseConnection.prepareStatement(query);
    stmt.setString(1, item.getName());
    stmt.setString(2, item.getDescription());
    // ... atur parameter lain
    stmt.setInt(7, item.getId());
    return stmt.executeUpdate() > 0;
}
```

#### 4.5 Manajemen Transaksi

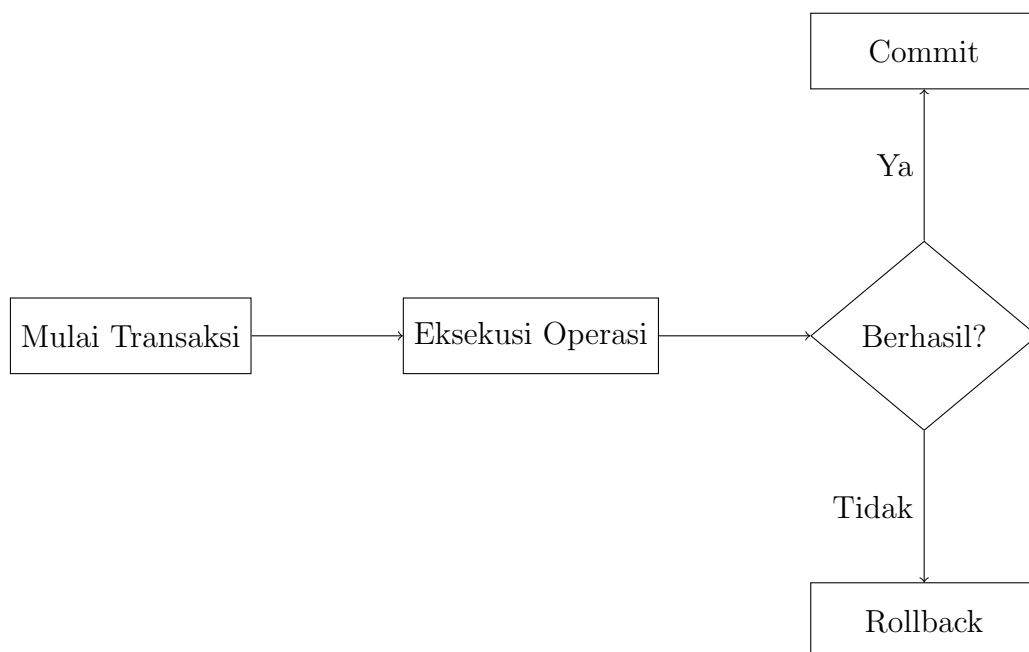


Figure 7: Alur Transaksi

##### 4.5.1 Pola Transaksi

- Transaksi Sederhana

- Commit operasi tunggal
- Rollback otomatis saat gagal
- Manajemen status koneksi

- **Transaksi Kompleks**

- Batch operasi ganda
- Kontrol commit manual
- Manajemen savepoint
- Penanganan rollback

#### 4.5.2 Penanganan Kesalahan

- **Kesalahan Basis Data**

- Penanganan eksepsi SQL
- Pemulihan kegagalan koneksi
- Rollback transaksi
- Notifikasi pengguna

- **Kesalahan Aplikasi**

- Kesalahan validasi data
- Pelanggaran aturan bisnis
- Pembersihan sumber daya
- Pencatatan kesalahan

## 5 Struktur Proyek

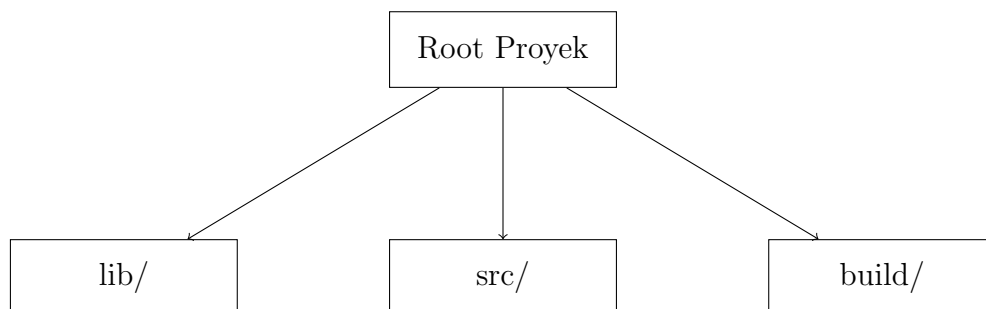


Figure 8: Struktur Direktori Proyek

Proyek ini mengikuti struktur direktori yang terorganisir dengan baik:

## 5.1 Isi Direktori Root

- `src/` - Direktori kode sumber
- `build/` - Direktori output build
- `lib/` - Pustaka eksternal
- `dataBackup/` - Direktori cadangan basis data
- `.idea/` - Konfigurasi IntelliJ IDEA
- `out/` - Direktori output terkompilasi
- `build.xml` - Konfigurasi build Ant
- `README.md` - Tinjauan proyek
- `.gitignore` - Aturan abaikan Git

## 6 Organisasi Kode Sumber

### 6.1 Aplikasi Utama

#### 6.1.1 Main.java

Lokasi: `src/com/app/Main.java` Ukuran: 1.6KB Baris: 56

**Tujuan** Titik masuk utama aplikasi yang menginisialisasi sistem dan mengelola navigasi jendela.

#### Komponen Kunci

- Manajemen tumpukan jendela
- Inisialisasi aplikasi
- Kontrol navigasi

#### Metode

- `main(String[] args)` - Titik masuk aplikasi
- `showWindow(JFrame newWindow)` - Menampilkan jendela baru
- `goBack()` - Menangani navigasi kembali
- `clearStack()` - Membersihkan tumpukan jendela

### 6.2 Direktori Komponen

Terletak di `src/Components/`, berisi komponen UI dan utilitas yang dapat digunakan kembali.

### 6.2.1 Utilitas

**DatabaseConnection.java** Lokasi: `src/Components/Utilities/DatabaseConnection.java`  
Ukuran: 1.7KB Baris: 52

**Tujuan** Mengelola koneksi basis data dan menyediakan metode akses basis data.

#### Fitur Kunci

- Connection pooling
- Eksekusi kueri
- Manajemen koneksi
- Penanganan kesalahan

#### Metode

- `getConnection()` - Membuat koneksi basis data
- `executeQuery(String query)` - Mengeksekusi kueri SQL
- `closeConnection()` - Menutup koneksi basis data

**Config.java** Lokasi: `src/Components/Utilities/Config.java` Ukuran: 1.5KB Baris: 55

**Tujuan** Mengelola konfigurasi dan pengaturan aplikasi.

#### Fitur Kunci

- Pemuatan konfigurasi
- Manajemen pengaturan
- Variabel lingkungan

**ImageResizer.java** Lokasi: `src/Components/Utilities/ImageResizer.java` Ukuran: 4.8KB Baris: 107

**Tujuan** Menangani pemrosesan dan pengubahan ukuran gambar untuk item museum.

#### Fitur Kunci

- Pengubahan ukuran gambar
- Konversi format
- Optimalisasi kualitas
- Pembuatan thumbnail

## Metode

- `resizeImage(Image original, int width, int height)`
- `createThumbnail(Image original)`
- `optimizeImage(Image image)`

**ItemQueries.java** Lokasi: `src/Components/Utilities/ItemQueries.java` Ukuran: 3.3KB Baris: 82

**Tujuan** Berisi kueri SQL untuk manajemen item.

## Fitur Kunci

- Operasi CRUD
- Kueri pencarian
- Kueri filter
- Operasi join

**ContainerPopulator.java** Lokasi: `src/Components/Utilities/ContainerPopulator.java` Ukuran: 5.8KB Baris: 130

**Tujuan** Mengelola populasi konten dinamis di kontainer UI.

## Fitur Kunci

- Pemuatan konten dinamis
- Manajemen kontainer
- Penanganan tata letak
- Pengikatan peristiwa (event binding)

**StatsCounter.java** Lokasi: `src/Components/Utilities/StatsCounter.java` Ukuran: 2.2KB Baris: 67

**Tujuan** Melacak dan menampilkan statistik sistem.

## Fitur Kunci

- Manajemen penghitung
- Perhitungan statistik
- Agregasi data
- Pemformatan tampilan

**CustomScrollBar.java** Lokasi: `src/Components/Utilities/CustomScrollBar.java` Ukuran: 1.5KB Baris: 43

**Tujuan** Implementasi scrollbar kustom untuk UI yang konsisten.

#### **Fitur Kunci**

- Gaya kustom
- Scrolling halus
- Penanganan peristiwa
- Manajemen ukuran

### **6.2.2 Kartu (Cards)**

**NavigationBar.java** Lokasi: `src/Components/Cards/NavigationBar.java` Ukuran: 14KB Baris: 277

**Tujuan** Komponen navigasi utama untuk aplikasi.

#### **Fitur Kunci**

- Manajemen menu
- Penanganan navigasi
- Adaptasi peran pengguna
- Pembaruan dinamis

**SearchBar.java** Lokasi: `src/Components/Cards/SearchBar.java` Ukuran: 2.9KB Baris: 76

**Tujuan** Implementasi fungsionalitas pencarian.

#### **Fitur Kunci**

- Pencarian real-time
- Opsi filter
- Riwayat pencarian
- Tampilan hasil

### **6.2.3 Kartu Koleksi**

**ItemCard.java** Lokasi: `src/Components/Cards/Collection/ItemCard.java` Ukuran: 7.9KB Baris: 190

**Tujuan** Menampilkan item museum individu dalam koleksi.

**Fitur Kunci**

- Tampilan item
- Penanganan gambar
- Tata letak informasi
- Penanganan interaksi

**RecentlyAddedCard.java** Lokasi: `src/Components/Cards/Collection/RecentlyAddedCard.java`  
Ukuran: 9.1KB Baris: 201

**Tujuan** Menampilkan item museum yang baru ditambahkan.

**Fitur Kunci**

- Tampilan item terbaru
- Pengurutan berbasis waktu
- Akses cepat
- Penanganan pembaruan

**RecentlyAddedContainer.java** Lokasi: `src/Components/Cards/Collection/RecentlyAddedContainer.java`  
Ukuran: 6.8KB Baris: 181

**Tujuan** Kontainer untuk item yang baru ditambahkan.

**Fitur Kunci**

- Manajemen tata letak
- Organisasi kartu
- Penanganan scroll
- Manajemen pembaruan

**MuseumCollection.java** Lokasi: `src/Components/Cards/Collection/MuseumCollection.java`  
Ukuran: 12KB Baris: 323

**Tujuan** Komponen tampilan koleksi utama.

## Fitur Kunci

- Tampilan koleksi
- Pemfilteran
- Pengurutan
- Paginasi

## 7 Direktori Halaman (Pages)

Terletak di `src/Pages/`, berisi semua halaman aplikasi yang diatur berdasarkan peran pengguna.

### 7.1 Halaman Login

#### 7.1.1 `LoginFrame.java`

Lokasi: `src/Pages/Login/LoginFrame.java` Ukuran: 9.4KB Baris: 236

**Tujuan** Antarmuka login utama untuk otentikasi pengguna.

## Fitur Kunci

- Otentikasi pengguna
- Validasi kata sandi
- Manajemen sesi
- Penanganan kesalahan

## Metode

- `authenticateUser()`
- `validateInput()`
- `initializeSession()`
- `handleLoginError()`

#### 7.1.2 `RegisterFrame.java`

Lokasi: `src/Pages/Login/RegisterFrame.java` Ukuran: 10KB Baris: 252

**Tujuan** Antarmuka pendaftaran pengguna baru.



## **Fitur Kunci**

- Pendaftaran pengguna
- Validasi input
- Pembuatan akun
- Penanganan kesalahan

## **Metode**

- `validateRegistration()`
- `createUserAccount()`
- `handleRegistrationError()`
- `checkUsernameAvailability()`

## **7.2 Halaman Admin**

### **7.2.1 Dashboard.java**

Lokasi: `src/Pages/Admin/Dashboard.java` Ukuran: 24KB Baris: 444

**Tujuan** Panel kontrol admin utama.

## **Fitur Kunci**

- Statistik sistem
- Fungsi akses cepat
- Pemantauan aktivitas
- Manajemen peringatan

## **Metode**

- `loadStatistics()`
- `updateDashboard()`
- `handleAlerts()`
- `manageQuickAccess()`

### **7.2.2 Inventory.java**

Lokasi: `src/Pages/Admin/Inventory.java` Ukuran: 9.0KB Baris: 235

**Tujuan** Antarmuka manajemen inventaris.

### **Fitur Kunci**

- Daftar item
- Fungsionalitas pencarian
- Manajemen filter
- Operasi item

### **Metode**

- loadInventory()
- searchItems()
- filterItems()
- updateInventory()

#### **7.2.3 AddItem.java**

Lokasi: `src/Pages/Admin/AddItem.java` Ukuran: 8.2KB Baris: 222

**Tujuan** Antarmuka untuk menambahkan item baru ke koleksi.

### **Fitur Kunci**

- Formulir input item
- Unggah gambar
- Pemilihan kategori
- Penugasan lokasi

### **Metode**

- validateItemInput()
- handleImageUpload()
- saveNewItem()
- assignLocation()

#### **7.2.4 EditItem.java**

Lokasi: `src/Pages/Admin/EditItem.java` Ukuran: 8.1KB Baris: 225

**Tujuan** Antarmuka untuk memodifikasi item yang ada.

## **Fitur Kunci**

- Pengeditan item
- Manajemen gambar
- Pembaruan status
- Pengeditan metadata

## **Metode**

- `loadItemData()`
- `updateItem()`
- `handleImageUpdate()`
- `validateChanges()`

### **7.2.5 Suggestions.java**

Lokasi: `src/Pages/Admin/Suggestions.java` Ukuran: 12KB Baris: 283

**Tujuan** Antarmuka manajemen saran pengguna.

## **Fitur Kunci**

- Tinjauan saran
- Manajemen status
- Penanganan respons
- Opsi pemfilteran

## **Metode**

- `loadSuggestions()`
- `updateStatus()`
- `sendResponse()`
- `filterSuggestions()`

## **7.3 Halaman Pengguna**

### **7.3.1 HomePage.java**

Lokasi: `src/Pages/User/HomePage.java` Ukuran: 33KB Baris: 784

**Tujuan** Antarmuka pengguna utama.

## **Fitur Kunci**

- Item unggulan
- Navigasi
- Konten yang dipersonalisasi
- Akses cepat

## **Metode**

- `loadFeaturedItems()`
- `updatePersonalizedContent()`
- `handleNavigation()`
- `initializeQuickAccess()`

### **7.3.2 MuseumCollection.java**

Lokasi: `src/Pages/User/MuseumCollection.java` Ukuran: 22KB Baris: 489

**Tujuan** Antarmuka penjelajahan koleksi.

## **Fitur Kunci**

- Tampilan koleksi
- Fungsionalitas pencarian
- Pemfilteran kategori
- Paginasi

## **Metode**

- `loadCollection()`
- `handleSearch()`
- `applyFilters()`
- `managePagination()`

### **7.3.3 ItemDisplay.java**

Lokasi: `src/Pages/User/ItemDisplay.java` Ukuran: 14KB Baris: 350

**Tujuan** Antarmuka tampilan detail item.

### **Fitur Kunci**

- Detail item
- Galeri gambar
- Item terkait
- Tampilan informasi

### **Metode**

- `loadItemDetails()`
- `displayImages()`
- `findRelatedItems()`
- `formatInformation()`

#### **7.3.4 ItemProfile.java**

Lokasi: `src/Pages/User/ItemProfile.java` Ukuran: 13KB Baris: 297

**Tujuan** Tampilan informasi item yang komprehensif.

### **Fitur Kunci**

- Informasi historis
- Status konservasi
- Riwayat pameran
- Metadata terperinci

### **Metode**

- `loadHistory()`
- `updateConservationStatus()`
- `displayExhibitionHistory()`
- `formatMetadata()`

#### **7.3.5 SuggestionForm.java**

Lokasi: `src/Pages/User/SuggestionForm.java` Ukuran: 9.1KB Baris: 256

**Tujuan** Antarmuka pengiriman umpan balik pengguna.

## **Fitur Kunci**

- Input formulir
- Pemilihan kategori
- Penetapan prioritas
- Penanganan pengiriman

## **Metode**

- `validateForm()`
- `handleSubmission()`
- `assignPriority()`
- `selectCategory()`

### **7.3.6 SuggestionTable.java**

Lokasi: `src/Pages/User/SuggestionTable.java` Ukuran: 6.5KB Baris: 184

**Tujuan** Tampilan riwayat saran pengguna.

## **Fitur Kunci**

- Daftar saran
- Tampilan status
- Melihat respons
- Opsi pemfilteran

## **Metode**

- `loadSuggestions()`
- `displayStatus()`
- `showResponses()`
- `applyFilters()`

## **8 Sistem Build**

### **8.1 build.xml**

Lokasi: `build.xml` Ukuran: 1.9KB Baris: 54

**Tujuan** File konfigurasi build Ant.

### **Fitur Kunci**

- Target build
- Manajemen dependensi
- Pengaturan kompilasi
- Konfigurasi deployment

### **Target Build**

- `clean` - Menghapus artefak build
- `compile` - Mengompilasi file sumber
- `jar` - Membuat JAR yang dapat dieksekusi
- `run` - Mengompilasi dan menjalankan aplikasi

## **9 Basis Data**

### **9.1 `museum_insert_queries.sql`**

Lokasi: `src/tools/museum_insert_queries.sql` Ukuran: 541KB Baris: 8477

**Tujuan** Skrip inisialisasi basis data dan penyisipan data.

### **Fitur Kunci**

- Pembuatan tabel
- Penyisipan data
- Pembuatan indeks
- Definisi batasan (constraint)

### **9.2 `get_data.py`**

Lokasi: `src/tools/get_data.py` Ukuran: 8.5KB Baris: 180

**Tujuan** Utilitas pemrosesan dan ekstraksi data.

### **Fitur Kunci**

- Ekstraksi data
- Konversi format
- Validasi data
- Fungsionalitas ekspor

## **10 Pemeliharaan dan Pembaruan**

### **10.1 Pemeliharaan Kode**

1. Ikuti struktur paket yang ada
2. Pertahankan pemisahan kepentingan (separation of concerns)
3. Gunakan tumpukan jendela untuk navigasi
4. Simpan komponen UI di direktori Komponen
5. Ikuti konvensi pengkodean Java
6. Dokumentasikan semua fitur baru
7. Uji secara menyeluruh sebelum deployment

### **10.2 Pemeliharaan Basis Data**

1. Pencadangan rutin menggunakan fungsionalitas pencadangan
2. Pantau kinerja basis data
3. Perbarui skema basis data menggunakan skrip SQL yang disediakan
4. Jaga integritas data
5. Optimalkan kueri
6. Pantau penggunaan penyimpanan

## **11 Panduan Penggunaan**

### **11.1 Instalasi**

1. Pastikan Java Runtime Environment terpasang
2. Pasang MySQL Server
3. Jalankan skrip inisialisasi basis data
4. Build proyek menggunakan Ant
5. Konfigurasi koneksi basis data
6. Siapkan direktori cadangan



## 11.2 Menjalankan Aplikasi

1. Jalankan file JAR atau gunakan target run Ant
2. Login dengan kredensial yang sesuai
3. Navigasi melalui aplikasi menggunakan antarmuka yang disediakan
4. Ikuti panduan khusus peran
5. Gunakan sumber daya bantuan yang disediakan

## 12 Pemecahan Masalah

Masalah umum dan solusinya:

- Masalah koneksi basis data - Periksa layanan dan kredensial MySQL
- Masalah rendering UI - Verifikasi kompatibilitas versi Java
- Kegagalan build - Pastikan semua dependensi ada
- Masalah kinerja - Periksa sumber daya sistem
- Inkonsistensi data - Verifikasi integritas basis data

## 13 Panduan Pengembangan

- Ikuti konvensi pengkodean Java
- Dokumentasikan fitur dan perubahan baru
- Uji secara menyeluruh sebelum deployment
- Pertahankan cadangan data penting
- Gunakan kontrol versi secara efektif
- Ikuti praktik terbaik keamanan

## 14 Kesimpulan

Dokumentasi ini memberikan panduan komprehensif untuk Perangkat Lunak Manajemen Museum. Untuk dukungan atau klarifikasi tambahan, silakan hubungi tim pengembang.

## 15 Klasifikasi Lapisan Arsitektur

### 15.1 Lapisan Frontend

Lapisan frontend terdiri dari semua komponen dan halaman yang terkait dengan UI:

### 15.1.1 Halaman

- src/Pages/Login/
  - LoginFrame.java - Antarmuka login
  - RegisterFrame.java - Antarmuka pendaftaran
- src/Pages/Admin/
  - Dashboard.java - Panel kontrol admin
  - Inventory.java - Manajemen inventaris
  - AddItem.java - Antarmuka penambahan item
  - EditItem.java - Antarmuka pengeditan item
  - Suggestions.java - Manajemen saran
- src/Pages/User/
  - HomePage.java - Antarmuka beranda pengguna
  - MuseumCollection.java - Penjelajahan koleksi
  - ItemDisplay.java - Tampilan detail item
  - ItemProfile.java - Tampilan profil item
  - SuggestionForm.java - Pengiriman saran
  - SuggestionTable.java - Riwayat saran

### 15.1.2 Komponen UI

- src/Components/Cards/
  - NavigationBar.java - Navigasi utama
  - SearchBar.java - Fungsionalitas pencarian
  - Collection/ItemCard.java - Kartu tampilan item
  - Collection/RecentlyAddedCard.java - Kartu item terbaru
  - Collection/RecentlyAddedContainer.java - Kontainer item terbaru
  - Collection/MuseumCollection.java - Tampilan koleksi
- src/Components/Utilities/
  - CustomScrollBar.java - Scrollbar UI kustom
  - ContainerPopulator.java - Populasi konten dinamis

## 15.2 Lapisan Middleware

Lapisan middleware menangani logika bisnis dan pemrosesan data:

- `src/Components/Utilities/`
  - `ImageResizer.java` - Pemrosesan gambar
  - `StatsCounter.java` - Manajemen statistik
  - `Config.java` - Manajemen konfigurasi
- `src/com/app/`
  - `Main.java` - Titik masuk dan navigasi aplikasi

## 15.3 Lapisan Backend

Lapisan backend mengelola penyimpanan data dan operasi basis data:

- `src/Components/Utilities/`
  - `DatabaseConnection.java` - Manajemen koneksi basis data
  - `ItemQueries.java` - Kueri basis data
- `src/tools/`
  - `museum_insert_queries.sql` - Skema dan data basis data
  - `get_data.py` - Utilitas pemrosesan data
- `dataBackup/` - Direktori cadangan basis data

## 15.4 File Konfigurasi

- `build.xml` - Konfigurasi build
- `README.md` - Dokumentasi proyek
- `.gitignore` - Konfigurasi kontrol versi
- `Museum Management Software - Project Files.iml` - Konfigurasi proyek