

**IMPLEMENTASI *PERN STACK* DALAM PENGEMBANGAN SISTEM  
DIGITALISASI DAN MANAJEMEN KOLEKSI MUSEUM LAMPUNG**

**(Skripsi)**

**Oleh**

**YOGI APRIO**

**2115031023**



**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
2025**

## ABSTRAK

### IMPLEMENTASI *PERN STACK* DALAM PENGEMBANGAN SISTEM DIGITALISASI DAN MANAJEMEN KOLEKSI MUSEUM LAMPUNG

Oleh  
YOGI APRIO

Museum memiliki peran penting dalam melestarikan warisan budaya sekaligus menyebarluaskan informasi kepada masyarakat. Namun, pengelolaan koleksi secara manual di Museum Lampung masih menghadapi sejumlah kendala, antara lain keterbatasan akses publik, risiko kehilangan data, dan pencatatan yang kurang terstruktur. Untuk menjawab permasalahan tersebut, penelitian ini mengembangkan sistem digitalisasi dan manajemen koleksi berbasis *PERN Stack* (*PostgreSQL*, *Express.js*, *React.js*, *Node.js*) sebagai solusi berbasis teknologi modern. Sistem yang dikembangkan menghadirkan fitur utama berupa operasi *CRUD* koleksi, unggah foto dan video 360°, serta pembuatan *QR Code* yang terhubung langsung ke basis data. Fitur tersebut tidak hanya memudahkan pengelola dalam mendokumentasikan koleksi, tetapi juga memungkinkan pengunjung mengakses informasi koleksi secara interaktif melalui perangkat digital. Dari sisi keamanan, sistem dilengkapi *JSON Web Token* (JWT) untuk autentikasi, *hashing password* menggunakan *Bcrypt*, serta *Role-Based Access Control* (RBAC) untuk membedakan hak akses antara admin dan superadmin. Hasil pengujian memperlihatkan bahwa *backend* memiliki stabilitas tinggi dengan cakupan unit test mencapai 100%. Pada *frontend*, pengujian performa melalui Lighthouse mencatat skor 95% untuk *desktop* dan 86% untuk *mobile*, dengan capaian 100% pada aksesibilitas dan praktik terbaik. Proses unggah media menunjukkan waktu 6,17 detik untuk foto dan 44,91 detik untuk video 360°, sehingga ukuran file terbukti sangat berpengaruh terhadap kecepatan unggah. Secara keseluruhan, sistem ini mendukung digitalisasi koleksi museum, memperkuat pengelolaan data, serta meningkatkan kualitas pengalaman pengunjung melalui akses digital yang lebih luas, interaktif, dan berkelanjutan.

Kata Kunci: Aksesibilitas, Digitalisasi, Museum, Manajemen Koleksi, *PERN Stack*.

## **ABSTRACT**

### **IMPLEMENTATION OF PERN STACK IN THE DEVELOPMENT OF A DIGITALIZATION AND COLLECTION MANAGEMENT SYSTEM FOR THE LAMPUNG MUSEUM**

**By**

**YOGI APRIO**

Museums play a crucial role in preserving cultural heritage and disseminating knowledge to the public. However, manual collection management at the Lampung Museum still faces several issues, including limited accessibility, risk of data loss, and unstructured documentation. To address these challenges, this study developed a digitalization and collection management system based on the PERN Stack (PostgreSQL, Express.js, React.js, Node.js) as a modern technological approach. The system integrates key features such as collection CRUD operations, photo and 360° video uploads, and QR Code generation directly linked to the database. These features not only assist administrators in documenting collections more systematically but also enable visitors to access collection information interactively through digital devices. In terms of security, the system employs JSON Web Token (JWT) for authentication, password hashing with Bcrypt, and Role-Based Access Control (RBAC) to distinguish access privileges between administrators and super administrators. Testing results demonstrated strong backend reliability, with unit test coverage reaching 100%. On the frontend, performance testing using Lighthouse achieved 95% for desktop and 86% for mobile, with 100% in accessibility and best practices. Media upload performance recorded 6.17 seconds for photos and 44.91 seconds for 360° videos, indicating that file size has a substantial impact on upload duration. In conclusion, the system supports museum collection digitalization, strengthens data management, and enhances visitor experiences through broader, interactive, and sustainable digital access.

**Keywords :** Accessibility, Digitalization, Collection Management, Museum, PERN Stack,.

**IMPLEMENTASI *PERN STACK* DALAM PENGEMBANGAN SISTEM  
DIGITALISASI DAN MANAJEMEN KOLEKSI MUSEUM LAMPUNG**

**Oleh**

**YOGI APRIO**

**Skripsi**

**Sebagai salah satu syarat untuk mendapat gelar**

**SARJANA TEKNIK**

**Pada**

**Jurusan Teknik Elektro**

**Fakultas Teknik**

**Universitas Lampung**



**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS LAMPUNG  
2025**

Judul : IMPLEMENTASI *PERN STACK*  
DALAM PENGEMBANGAN  
SISTEM DIGITALISASI DAN  
MANAJEMEN KOLEKSI  
MUSEUM LAMPUNG

Nama Mahasiswa : Yogi Aprio

Nomor Pokok Mahasiswa : 2115031023

Jurusan : Teknik Elektro

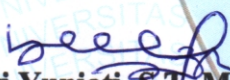
Fakultas : Teknik

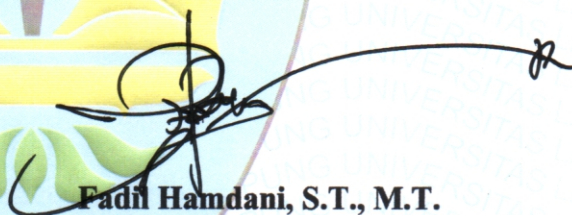
### MENYETUJUI

#### 1. Komisi Pembimbing

Pembimbing Utama

Pembimbing Pendamping

  
**Yetti Yuniati, S.T., M.T.**  
NIP. 198001132009122002

  
**Fadi Hamdani, S.T., M.T.**  
NIP. 198707262022031003

#### 2. Mengetahui

Ketua Jurusan Teknik Elektro

Ketua Program studi Teknik Elektro

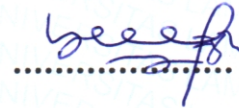
  
**Herlinawati, S.T., M.T.**  
NIP. 197103141999032001

  
**Sumadi, S.T., M.T.**  
NIP. 197311042000031001

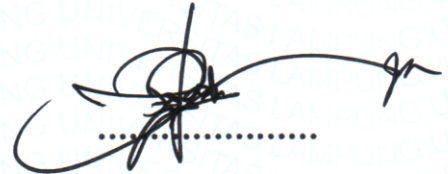
## MENGESAHKAN

1. Tim Penguji

Ketua : Yetti Yuniati, S.T, M.T



Sekretaris : Fadil Hamdani, S.T., M.T.



Penguji : Ing. Ardian Ulvan, S.T., M.Sc., Ph.D. ....



2. Dekan Fakultas Teknik



Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc. )

NIP. 19750928 200112 1 002



Tanggal Lulus Ujian Skripsi : 20 Agustus 2025

## SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi ini tidak terdapat karya yang pernah dilakukan orang lain dan, sepanjang sepengetahuan saya tidak terdapat atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini sebagaimana yang disebutkan dalam daftar pustaka. Selain itu, saya menyatakan pula bahwa skripsi ini dibuat oleh saya sendiri.

Apabila pernyataan saya tidak benar, maka saya bersedia dikenai sanksi sesuai dengan hukum yang berlaku.

Bandar Lampung, 20 Agustus 2025



Yogi Aprio

NPM. 2115031023

## RIWAYAT HIDUP



Penulis lahir di Astra Ksetra pada tanggal 30 April 2003. Penulis merupakan putra kedua dari empat bersaudara, anak dari pasangan Bahtiar dan Yuniar. Penulis mengawali pendidikan di SD Abadi Perkasa dari tahun 2009 hingga 2015, kemudian melanjutkan ke SMP Abadi Perkasa pada tahun 2015 hingga 2018. Pendidikan menengah atas ditempuh di SMA Negeri 1 Tumijajar, jurusan Matematika dan Ilmu Pengetahuan Alam (MIPA), dari tahun 2018 hingga 2021.

Pada tahun 2021, penulis diterima sebagai mahasiswa Universitas Lampung, Program Studi Teknik Elektro, melalui jalur SNMPTN. Selama masa studi, penulis aktif dalam kegiatan akademik dan organisasi, antara lain menjadi Anggota Departemen Kaderisasi dan Pengembangan Organisasi Himpunan Mahasiswa Teknik Elektro selama dua periode. Penulis memiliki beberapa pencapaian, di antaranya lolos pendanaan Program Kreativitas Mahasiswa (PKM) 2023 dengan judul "Ajar Tanam Mentimun Berbasis Soilless Culture System untuk Pengkayaan Eko-agrowisata Kampung Liman Benawi, Lampung Tengah", serta lolos pendanaan PPK ORMAWA 2023 dengan judul "Pengembangan Desa Gisting Permai sebagai Desa Agropolitan dengan Sistem Greenhouse Berbasis Intelligent Control System". Selanjutnya, pada tahun 2024 penulis mengikuti program MBKM Studi Independen di Dicoding dengan fokus pada pengembangan Front-end dan Back-end web.

## PERSEMBAHAN



**Alhamdulillah, Atas Berkat Rahmat Allah yang Maha Kuasa**

**KUPERSEMBAHKAN KARYA INI UNTUK**

*Bapak dan Ibu Tercinta*

*Bahtiar dan Yuniar*

*Saudara Tercinta*

*Yusri Arifin, Ilham Ramadan, Rahmad Andika*

Keluarga Besar yang selalu menjadi sumber semangat, Dosen  
yang telah membimbing dengan tulus,  
serta seluruh teman yang setia kebersamai setiap langkah perjalanan ini

## MOTTO

“Sesungguhnya bersama kesulitan ada kemudahan.”

*(QS. Al-Insyirah [94]: 6)*

“Kegagalan hanyalah koma, bukan penutup cerita; sebab hidup adalah kalimat panjang yang terus ditulis, dan setiap jeda hanyalah ruang untuk mengatur napas sebelum melangkah lebih jauh.”

(Penulis)

## SANWACANA

Alhamdulillah segala puji bagi Allah subhanahu wa ta'ala atas limpahan rahmat dan karunia-Nya lah penulis dapat menyelesaikan skripsi yang berjudul “Implementasi *PERN Stack* Dalam Pengembangan Sistem Digitalisasi Dan Manajemen Koleksi Museum Lampung” sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung. Shalawat serta salam senantiasa penulis haturkan kepada Nabi Muhammad SAW, semoga kita termasuk golongan yang mendapatkan syafaatnya di Yaumul Qiyamah kelak, Aamiin.

Penyusunan skripsi ini tidak terlepas dari bantuan, bimbingan, kritk, saran, dan doa dari berbagai pihak. Oleh karena itu, pada kesempatan ini sebagai wujud rasa hormat, penulis mengucapkan banyak terima kasih kepada:

1. Kedua orang tua tercinta, atas doa, dukungan, dan pengorbanan yang tiada henti dalam memberikan motivasi dan semangat kepada penulis.
2. Ibu Yetti Yuniati, S.T., M.T., selaku dosen pembimbing utama, atas arahan, bimbingan, serta motivasi yang senantiasa diberikan.
3. Bapak Fadil Hamdani, S.T., M.T., selaku dosen pembimbing pendamping, atas masukan, saran, dan dukungan yang membantu dalam penyusunan skripsi ini.
4. Bapak Ing. Ardian Ulvan, S.T., M.Sc., Ph.D., selaku dosen penguji, atas kritik, saran, dan masukan yang membangun dalam penyempurnaan penelitian.
5. Ibu Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., I.P.M., ASEAN Eng., selaku Rektor Universitas Lampung.
6. Bapak Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung.
7. Ibu Herlinawati, S.T., M.T., selaku Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.
8. Bapak Sumadi, S.T., M.T., selaku Kepala Program Studi Teknik Elektro Fakultas Teknik Universitas Lampung.
9. Bapak Dr. Eng. Lukmanul Hakim, S.T., M.Sc., selaku dosen pembimbing akademik, atas arahan dan bimbingan yang diberikan selama masa studi.

10. Seluruh dosen Jurusan Teknik Elektro, Fakultas Teknik Universitas Lampung, atas ilmu, wawasan, dan pengalaman berharga yang telah diberikan kepada penulis.
11. Mbak Nurul beserta staf Jurusan Teknik Elektro dan Fakultas Teknik, atas bantuan dan pelayanan dalam urusan administrasi maupun kebutuhan lainnya.
12. Rekan-rekan Kaderisasi dan Pengembangan Organisasi angkatan 2021, atas dukungan, kebersamaan, dan motivasi yang diberikan selama proses perkuliahan.
13. Keluarga besar EXCALTO 2021, yang telah menjadi bagian penting dalam perjalanan akademik dan kebersamaan di lingkungan kampus.
14. Keluarga besar HIMATRO, khususnya kakak-kakak angkatan 2018, 2019, dan 2020 atas bimbingan serta teladan yang diberikan, serta adik-adik angkatan 2022, 2023, dan 2024 atas motivasi yang senantiasa menginspirasi penulis.
15. Keluarga besar konsentrasi TELTI, yang telah menjadi wadah bagi penulis untuk memperdalam pengetahuan dan mengembangkan keterampilan akademik.
16. Seluruh pihak yang tidak dapat disebutkan satu per satu, yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penulisan skripsi ini masih terdapat banyak kekurangan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang bersifat membangun guna menyempurnakan skripsi ini di masa mendatang. Semoga skripsi ini dapat memberikan manfaat bagi penulis secara khusus dan masyarakat pada umumnya.

Bandar Lampung, 20 Agustus 2025  
Penulis,

Yogi Aprio  
NPM. 2115031023

## DAFTAR ISI

|                                   | Halaman     |
|-----------------------------------|-------------|
| <b>ABSTRAK.....</b>               | <b>ii</b>   |
| <b>HALAMAN JUDUL .....</b>        | <b>iv</b>   |
| <b>LEMBAR PERSETUJUAN .....</b>   | <b>v</b>    |
| <b>LEMBAR PENGESAHAN.....</b>     | <b>vi</b>   |
| <b>SURAT PERNYATAAN .....</b>     | <b>vii</b>  |
| <b>RIWAYAT HIDUP .....</b>        | <b>viii</b> |
| <b>PERSEMBAHAN.....</b>           | <b>ix</b>   |
| <b>MOTTO.....</b>                 | <b>x</b>    |
| <b>SANWACANA.....</b>             | <b>xi</b>   |
| <b>DAFTAR ISI.....</b>            | <b>xii</b>  |
| <b>DAFTAR GAMBAR .....</b>        | <b>xvi</b>  |
| <b>DAFTAR TABEL.....</b>          | <b>xix</b>  |
| <b>I. PENDAHULUAN .....</b>       | <b>1</b>    |
| 1.1 Latar Belakang .....          | 1           |
| 1.2 Rumusan Masalah.....          | 2           |
| 1.3 Batasan Masalah .....         | 2           |
| 1.4 Tujuan Penelitian .....       | 3           |
| 1.5 Manfaat Penelitian .....      | 3           |
| 1.6 Sistematika Penulisan .....   | 4           |
| <b>II. TINJAUAN PUSTAKA.....</b>  | <b>5</b>    |
| 2.1 Penelitian Terdahulu .....    | 5           |
| 2.2 Digitalisasi dan Museum ..... | 13          |
| 2.2.1 Digitalisasi .....          | 13          |
| 2.2.2 Manajemen Museum .....      | 13          |
| 2.2.3 Museum .....                | 14          |
| 2.2.4 Koleksi Museum.....         | 14          |

|                                                           |           |
|-----------------------------------------------------------|-----------|
| 2.3 Teknologi dalam Pengembangan Sistem .....             | 14        |
| 2.3.1 <i>SDLC (Software Development Life Cycle)</i> ..... | 15        |
| 2.3.2 <i>Fullstack Development</i> .....                  | 15        |
| 2.4 Teknologi dan Alat Pendukung .....                    | 15        |
| 2.4.1 <i>Visual Studio Code</i> .....                     | 16        |
| 2.4.2 <i>Postman</i> .....                                | 16        |
| 2.4.3 <i>Browser</i> .....                                | 16        |
| 2.5 Komponen Sistem <i>PERN Stack</i> .....               | 17        |
| 2.5.1 <i>PostgreSQL</i> .....                             | 17        |
| 2.5.2 <i>Express.js</i> .....                             | 17        |
| 2.5.3 <i>React.js</i> .....                               | 18        |
| 2.5.4 <i>Node.js</i> .....                                | 18        |
| 2.6 <i>Database</i> .....                                 | 18        |
| 2.7 <i>REST API</i> .....                                 | 19        |
| 2.8 Kamera .....                                          | 19        |
| 2.9 <i>Multer</i> .....                                   | 20        |
| 2.10 Otentikasi .....                                     | 20        |
| 2.11 Keamanan Data dan Enkripsi <i>Password</i> .....     | 21        |
| <b>III. METODOLOGI PENELITIAN .....</b>                   | <b>22</b> |
| 3.1 Waktu dan Tempat Penelitian .....                     | 22        |
| 3.2 Alat dan Bahan .....                                  | 22        |
| 3.3 Diagram Alir Penelitian .....                         | 24        |
| 3.3.1 Studi Literatur .....                               | 27        |
| 3.3.2 Identifikasi Masalah .....                          | 27        |
| 3.3.3 Analisis Kebutuhan .....                            | 27        |
| 3.3.4 Perancangan Sistem .....                            | 28        |
| 3.3.4.1 Arsitektur Sistem .....                           | 28        |
| 3.3.4.2 Perancangan Fungsional Sistem .....               | 30        |
| 3.3.4.3 Perancangan Tabel <i>Database</i> .....           | 45        |
| 3.3.4.4 Perancangan Antarmuka Pengguna .....              | 47        |
| 3.3.5 Pengembangan Sistem .....                           | 60        |
| 3.3.5.1 <i>Backend</i> .....                              | 60        |

|                                                                                  |           |
|----------------------------------------------------------------------------------|-----------|
| 3.3.5.2 <i>Frontend</i> .....                                                    | 63        |
| 3.3.5.3 Integrasi Sistem.....                                                    | 65        |
| 3.3.6 Pengujian Sistem .....                                                     | 65        |
| 3.3.7 <i>Deployment Website</i> .....                                            | 67        |
| 3.3.8 Analisis Capaian .....                                                     | 68        |
| 3.3.9 Laporan Penelitian .....                                                   | 68        |
| <b>IV. HASIL DAN PEMBAHASAN .....</b>                                            | <b>70</b> |
| 4.1 Implementasi Sistem.....                                                     | 70        |
| 4.1.1 Implementasi <i>Backend</i> .....                                          | 70        |
| 4.1.1.1 Struktur Proyek <i>Backend</i> .....                                     | 70        |
| 4.1.1.2 Implementasi Model Data.....                                             | 71        |
| 4.1.1.3 Autentikasi dan Keamanan Pengguna .....                                  | 72        |
| 4.1.1.4 Operasi <i>CRUD</i> dan Validasi Data .....                              | 74        |
| 4.1.1.5 Pengunggahan Media dan Pembuatan Kode <i>QR</i> ....                     | 75        |
| 4.1.2 Implementasi <i>Frontend</i> .....                                         | 77        |
| 4.1.2.1 Struktur Folder <i>Frontend</i> .....                                    | 77        |
| 4.1.2.2 Manajemen <i>State</i> Aplikasi.....                                     | 78        |
| 4.1.2.3 Desain Antarmuka Responsif dan Aksesibel.....                            | 79        |
| 4.1.2.4 Implementasi Halaman Publik.....                                         | 79        |
| 4.1.2.5 Implementasi <i>Dashboard</i> Admin Koleksi dan<br>Admin Kegiatan.....   | 87        |
| 4.1.2.6 Implementasi <i>Dashboard</i> Superadmin .....                           | 91        |
| 4.1.2.7 Implementasi Viewer Video 360 Derajat .....                              | 94        |
| 4.2 Pengujian Sistem.....                                                        | 94        |
| 4.2.1 Pengujian Fungsionalitas ( <i>Black-Box Testing</i> ) .....                | 95        |
| 4.2.2 Pengujian <i>API</i> ( <i>Postman Testing</i> ) .....                      | 96        |
| 4.2.3 Pengujian Unit ( <i>Jest Testing</i> ) .....                               | 99        |
| 4.3 Hasil <i>Deployment</i> dan Analisis Capaian.....                            | 101       |
| 4.3.1 Proses <i>Deployment Website</i> .....                                     | 101       |
| 4.3.2 Analisis Performa Berdasarkan Pengujian <i>Desktop</i> dan<br>Mobile ..... | 103       |
| 4.3.3 Analisis Waktu Respons <i>Upload</i> Gambar dan Video .....                | 105       |
| 4.3.4 Analisis Fungsionalitas Sistem .....                                       | 106       |

|                                                           |     |
|-----------------------------------------------------------|-----|
| 4.3.5 Analisis Keamanan Sistem.....                       | 107 |
| 4.3.6 Analisis Aksesibilitas dan Antarmuka Pengguna ..... | 108 |
| V. KESIMPULAN DAN SARAN .....                             | 110 |
| 5.1 Kesimpulan .....                                      | 110 |
| 5.2 Saran .....                                           | 111 |
| DAFTAR PUSTAKA.....                                       | 113 |

## DAFTAR GAMBAR

|                                                                     | Halaman |
|---------------------------------------------------------------------|---------|
| 2.1 <i>Roadmap</i> Penelitian .....                                 | 12      |
| 3.1 Diagram Alir Penelitian .....                                   | 25      |
| 3.2 Diagram Arsitektur Sistem.....                                  | 30      |
| 3.3 <i>Use Case</i> Diagram .....                                   | 33      |
| 3.4 <i>Sequence</i> Diagram <i>Login</i> Admin .....                | 34      |
| 3.5 <i>Sequence</i> Diagram <i>Login</i> Superadmin.....            | 35      |
| 3.6 <i>Sequence</i> Diagram Tambah Admin .....                      | 36      |
| 3.7 <i>Sequence</i> Diagram Edit Admin.....                         | 37      |
| 3.8 <i>Sequence</i> Diagram Hapus Admin .....                       | 38      |
| 3.9 <i>Sequence</i> Diagram Tambah Koleksi (Admin Koleksi).....     | 39      |
| 3.10 <i>Sequence</i> Diagram Edit Koleksi (Admin Koleksi).....      | 40      |
| 3.11 <i>Sequence</i> Diagram Hapus Koleksi (Admin Koleksi) .....    | 41      |
| 3.12 <i>Sequence</i> Diagram Tambah Kegiatan (Admin Kegiatan) ..... | 42      |
| 3.13 <i>Sequence</i> Diagram Edit Kegiatan (Admin Kegiatan).....    | 43      |
| 3.14 <i>Sequence</i> Diagram Hapus Kegiatan (Admin Kegiatan) .....  | 44      |
| 3.15 Halaman <i>Landing Page (Home)</i> .....                       | 48      |
| 3.16 Halaman <i>Login</i> .....                                     | 49      |
| 3.17 Halaman Koleksi .....                                          | 50      |
| 3.18 Halaman Detail Koleksi.....                                    | 51      |
| 3.19 Halaman Daftar Kegiatan .....                                  | 52      |
| 3.20 Halaman Detail Kegiatan.....                                   | 53      |
| 3.21 Halaman Statistik <i>Dashboard</i> Admin.....                  | 54      |
| 3.22 Halaman Tambah Koleksi .....                                   | 54      |
| 3.23 Halaman Daftar Koleksi .....                                   | 55      |
| 3.24 Halaman Edit Koleksi.....                                      | 55      |

|                                                          |    |
|----------------------------------------------------------|----|
| 3.25 Halaman Tambah Kegiatan .....                       | 56 |
| 3.26 Halaman Daftar Kegiatan .....                       | 56 |
| 3.27 Halaman Edit Kegiatan.....                          | 57 |
| 3.28 Halaman Statistik <i>Dashboard</i> Superadmin ..... | 58 |
| 3.29 Halaman Tambah Admin.....                           | 58 |
| 3.30 Halaman Daftar Admin.....                           | 58 |
| 3.31 Halaman Edit Admin .....                            | 59 |
| 3.32 Halaman Daftar Koleksi (Superadmin) .....           | 59 |
| 3.33 Halaman Daftar Kegiatan (Superadmin) .....          | 59 |
| 4.1 Struktur Folder <i>Backend</i> .....                 | 71 |
| 4.2 Tabel Collections.....                               | 72 |
| 4.3 Tabel Users.....                                     | 72 |
| 4.4 Tabel Events .....                                   | 72 |
| 4.5 Tabel Visitors .....                                 | 72 |
| 4.6 Tabel <i>QR Code</i> .....                           | 72 |
| 4.7 Kode Autentikasi .....                               | 73 |
| 4.8 Kode <i>Hashing</i> Password .....                   | 74 |
| 4.9 Kode Rute <i>API</i> Koleksi .....                   | 74 |
| 4.10 Kode Validasi Koleksi.....                          | 74 |
| 4.11 Kode Upload Koleksi ke Cloudinary.....              | 75 |
| 4.12 Kode Pembuatan <i>QR Code</i> .....                 | 76 |
| 4.13 Struktur Folder <i>Frontend</i> .....               | 74 |
| 4.14 Halaman Beranda.....                                | 80 |
| 4.15 Halaman Daftar Koleksi .....                        | 82 |
| 4.16 Halaman Detail Koleksi.....                         | 83 |
| 4.17 Halaman Daftar Kegiatan .....                       | 84 |
| 4.18 Halaman Detail Kegiatan.....                        | 85 |
| 4.19 Halaman <i>Login</i> .....                          | 86 |
| 4.20 Halaman Tentang.....                                | 87 |
| 4.21 Halaman Statistik <i>Dashboard</i> Admin.....       | 88 |
| 4.22 Halaman Tambah Koleksi .....                        | 88 |
| 4.23 Halaman Daftar Koleksi .....                        | 89 |

|                                                                             |     |
|-----------------------------------------------------------------------------|-----|
| 4.24 Halaman Edit Koleksi.....                                              | 89  |
| 4.25 Halaman Tambah Kegiatan .....                                          | 90  |
| 4.26 Halaman Daftar Kegiatan .....                                          | 90  |
| 4.27 Halaman Edit Kegiatan.....                                             | 91  |
| 4.28 Halaman Statistik <i>Dashboard</i> Superadmin .....                    | 92  |
| 4.29 Halaman Tambah Admin.....                                              | 92  |
| 4.30 Halaman Daftar admin.....                                              | 93  |
| 4.31 Halaman Edit Admin .....                                               | 93  |
| 4.32 Halaman Daftar Koleksi (Superadmin) .....                              | 93  |
| 4.33 Halaman Daftar Kegiatan .....                                          | 94  |
| 4.34 Contoh Pengujian <i>Endpoint GET /api/collections</i> di Postman ..... | 99  |
| 4.35 Hasil Pengujian Performa <i>Desktop</i> .....                          | 104 |
| 4.36 Hasil Pengujian Performa Mobile .....                                  | 104 |
| 4.37 Hasil Pengujian Upload Gambar .....                                    | 105 |
| 4.38 Hasil Pengujian Upload Video .....                                     | 106 |

## DAFTAR TABEL

|                                                                                    | Halaman |
|------------------------------------------------------------------------------------|---------|
| Tabel 2.1 Penelitian Terdahulu.....                                                | 8       |
| Tabel 2.2 Jenis-Jenis Kamera.....                                                  | 19      |
| Tabel 3.1 Alat dan Bahan.....                                                      | 22      |
| Tabel 3.2 Analisis Kebutuhan .....                                                 | 28      |
| Tabel 3.3 Struktur Tabel Users .....                                               | 45      |
| Tabel 3.4 Struktur Tabel Collections .....                                         | 46      |
| Tabel 3.5 Struktur Tabel QRCode .....                                              | 46      |
| Tabel 3.6 Struktur Tabel Visitors.....                                             | 47      |
| Tabel 3.7 Struktur Tabel Events.....                                               | 47      |
| Tabel 3.8 Struktur Folder <i>Backend</i> .....                                     | 61      |
| Tabel 3.9 Struktur Folder <i>Frontend</i> .....                                    | 64      |
| Tabel 4.1 Hasil Pengujian Fungsionalitas Sistem ( <i>Black-Box Testing</i> ) ..... | 95      |
| Tabel 4.2 Hasil Pengujian <i>Api</i> ( <i>Postman Testing</i> ) .....              | 97      |
| Tabel 4.3 Hasil Pengujian Unit Sistem ( <i>Jest Testing</i> ).....                 | 100     |
| Tabel 4.4 Perbandingan Nilai <i>Desktop</i> dan <i>Mobile</i> .....                | 103     |

## I. PENDAHULUAN

### 1.1 Latar Belakang

Digitalisasi pengelolaan koleksi museum menjadi aspek krusial dalam upaya pelestarian budaya dan sejarah. Seiring dengan perkembangan teknologi, museum dituntut untuk beradaptasi dengan menggunakan sistem digital yang memungkinkan pengelolaan informasi secara efisien. Digitalisasi tidak hanya meningkatkan aksesibilitas koleksi bagi publik, tetapi juga membantu dalam pelestarian artefak melalui dokumentasi yang lebih baik dan aman [1]. Dengan memanfaatkan teknologi informasi, museum dapat menjangkau pengunjung yang lebih luas dan meningkatkan interaksi dengan pengunjung.

Museum Lampung, sebagai salah satu lembaga budaya di Indonesia, menghadapi tantangan dalam pengelolaan koleksi yang masih bersifat manual. Pengelolaan yang tidak terintegrasi berisiko menyebabkan kehilangan informasi penting tentang koleksi yang ada. Oleh karena itu, penerapan sistem berbasis *web* untuk digitalisasi koleksi sangat diperlukan. Sistem ini tidak hanya akan mempermudah pencarian dan penyimpanan data, tetapi juga mempercepat proses pemeliharaan dan aksesibilitas bagi peneliti dan masyarakat umum [2].

Dengan adanya perkembangan teknologi berbasis *web*, penerapan sistem digital di museum dapat menjadi solusi yang efektif. Digitalisasi koleksi museum tidak hanya memungkinkan pencatatan data yang lebih sistematis, tetapi juga meningkatkan aksesibilitas bagi masyarakat luas [3]. Teknologi berbasis *fullstack PERN* (*PostgreSQL*, *Express.js*, *React.js*, *Node.js*) menawarkan fleksibilitas dan performa tinggi dalam pengelolaan informasi koleksi museum, sehingga dapat mengatasi berbagai permasalahan dalam sistem konvensional.

Teknologi modern seperti video 360<sup>0</sup> dan *QR Code* juga dapat dimanfaatkan untuk meningkatkan pengalaman pengunjung. Dengan mengintegrasikan teknologi ini ke dalam sistem digital museum, pengunjung dapat mendapatkan informasi tambahan tentang koleksi melalui perangkat *mobile* seperti *smartphone*, tablet, dan perangkat lain yang mendukung akses *web* [4]. Ini tidak hanya membuat pengalaman berkunjung menjadi lebih interaktif tetapi juga menumbuhkan minat dan kesadaran sejarah di kalangan masyarakat. Berdasarkan permasalahan yang telah diidentifikasi, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem berbasis *PERN Stack* guna mendukung digitalisasi koleksi Museum Lampung.

## 1.2 Rumusan Masalah

Adapun rumusan masalah pada penelitian yang dibahas adalah sebagai berikut :

1. Bagaimana sistem digital berbasis *PERN Stack* dapat diterapkan untuk mendukung pengelolaan koleksi di Museum Lampung?
2. Bagaimana pengembangan sistem berbasis *PostgreSQL* mampu mendukung penyimpanan dan manajemen data koleksi museum secara terstruktur?
3. Bagaimana mekanisme otentikasi pengguna dapat diimplementasikan untuk membedakan akses antara berbagai peran admin dan memastikan akses publik bagi pengunjung dalam sistem informasi museum?

## 1.3 Batasan Masalah

Adapun batasan pada penelitian ini adalah sebagai berikut:

1. Penelitian ini hanya mencakup digitalisasi koleksi di Museum Lampung.
2. Penelitian ini berfokus pada fitur digitalisasi koleksi berbasis foto, deskripsi teks, video 360<sup>0</sup>, dan *QR code*.
3. Sistem yang dikembangkan fokus pada fitur dokumentasi koleksi museum dan tidak mencakup aspek transaksi atau manajemen keuangan museum.
4. Jumlah koleksi yang akan didigitalisasi dalam penelitian ini dibatasi

sebanyak 20 koleksi.

5. *Backend* aplikasi di-*deploy* menggunakan platform cloud Railway dan *frontend* di-*deploy* menggunakan platform Vercel.
6. Penelitian ini tidak membahas analisis perangkat keras atau infrastruktur *deployment*, fokus hanya pada implementasi dan pengembangan sistem digitalisasi dan manajemen koleksi museum.

#### 1.4 Tujuan Penelitian

Adapun tujuan dari penelitian adalah sebagai berikut:

1. Merancang dan mengimplementasikan sistem digital berbasis *PERN Stack* (*PostgreSQL*, *Express.js*, *React.js*, dan *Node.js*) untuk pengelolaan koleksi Museum Lampung.
2. Menerapkan dan mengembangkan sistem penyimpanan data berbasis *PostgreSQL* untuk mendukung manajemen data koleksi museum secara terstruktur dan terintegrasi dengan sistem digital yang dibangun.
3. Menerapkan sistem otentikasi untuk membedakan peran pengguna (admin) dalam pengelolaan akses terhadap data koleksi, serta memastikan akses publik bagi pengunjung tanpa memerlukan otentikasi.

#### 1.5 Manfaat Penelitian

Adapun manfaat dari penelitian adalah sebagai berikut:

1. Memberikan solusi bagi Museum Lampung dalam mendokumentasikan koleksi secara digital, sehingga mengurangi risiko kehilangan atau kerusakan data koleksi.
2. Mempermudah akses informasi koleksi museum bagi masyarakat, peneliti, dan pengelola museum melalui sistem berbasis web yang interaktif.
3. Mendukung perkembangan metode digitalisasi koleksi museum dengan integrasi teknologi seperti *QR Code* dan video 360° untuk meningkatkan aksesibilitas informasi.

## **1.6 Sistematika Penulisan**

Adapun sistematika penulisan pada penelitian ini adalah sebagai berikut:

### **I. PENDAHULUAN**

Bab ini menjelaskan latar belakang masalah, tujuan penelitian, rumusan masalah, batasan masalah, manfaat penelitian serta sistematika penulisan pada penelitian ini.

### **II. TINJAUAN PUSTAKA**

Bab ini menjelaskan mengenai penelitian sebelumnya dan juga teori pendukung yang menjadi pengantar pemahaman dan berkaitan dengan materi penelitian yang didapat dari berbagai sumber ilmiah.

### **III. METODOLOGI PENELITIAN**

Bab ini menjelaskan metodologi penelitian berupa waktu dan tempat penelitian, alat dan bahan yang digunakan dalam penelitian, serta metode dan alur penelitian yang digunakan dalam pelaksanaan penelitian.

### **IV. HASIL DAN PEMBAHASAN**

Bab ini berisi tentang proses pengambilan data, hasil yang didapatkan dari penelitian dan analisis data dari hasil penelitian.

### **V. KESIMPULAN DAN SARAN**

Bab ini berisi tentang kesimpulan dari penelitian yang dilakukan dan saran yang didasarkan pada hasil daya mengenai perbaikan dan pengembangan lebih lanjut agar didapatkan hasil yang lebih baik.

### **DAFTAR PUSTAKA**

Bab ini berisi referensi yang digunakan dalam penulisan dan pelaksanaan penelitian.

## II. TINJAUAN PUSTAKA

### 2.1 Penelitian Terdahulu

Digitalisasi museum telah menjadi fokus utama dalam pelestarian dan penyebaran budaya serta sejarah di era modern. Penelitian terkait digitalisasi museum menunjukkan bagaimana teknologi inovatif dapat diterapkan untuk meningkatkan aksesibilitas, efisiensi pengelolaan, dan keterlibatan pengunjung. Berikut adalah ringkasan dari berbagai penelitian yang telah dilakukan.

Penelitian berjudul "*A Rapid and Cost-Effective Pipeline for Digitization of Museum Specimens with 3D Photogrammetry*" dilakukan oleh Medina pada tahun 2020, yang bertujuan untuk mendigitalisasi spesimen museum dengan metode fotogrametri 3D. Penelitian ini menggunakan kamera, perangkat lunak, dan *turntable* otomatis untuk menghasilkan model 3D berkualitas tinggi dengan biaya rendah. Hasil penelitian menunjukkan bahwa metode ini dapat meningkatkan efisiensi dan aksesibilitas koleksi museum [6].

Penelitian berjudul "*Designing Mobile Application for Museum Enthusiasts*" dilakukan oleh Puspasari pada tahun 2020, yang bertujuan untuk meningkatkan keterlibatan publik dengan museum melalui pengembangan aplikasi *mobile*. Aplikasi ini dirancang menggunakan diagram *use case* berdasarkan kebutuhan pengguna yang diperoleh melalui kuesioner terbuka. Hasil penelitian menunjukkan bahwa fitur *augmented reality* dan *proximity* sensor pada aplikasi dapat meningkatkan interaksi antara pengunjung dan koleksi museum [7].

Penelitian berjudul "*Creation of a Virtual Museum for the Dissemination of 3D Models of Historical Clothing*" dilakukan oleh Meier et al. pada tahun 2021, yang berfokus pada pembuatan museum virtual untuk pakaian abad ke-16 di Spanyol. Museum ini menggunakan perangkat lunak Blender dan Unity untuk menciptakan pengalaman interaktif. Hasil penelitian menunjukkan bahwa museum virtual ini

meningkatkan aksesibilitas koleksi secara global [8].

Penelitian berjudul "Perancangan Sistem Informasi Manajemen Koleksi Museum Berbasis *Web* (Studi Kasus Museum Lampung)" dilakukan oleh Istiawan dan Nuralia pada tahun 2021. Penelitian ini mengembangkan sistem berbasis *web* dengan metode *UML* untuk meningkatkan pengelolaan koleksi di Museum Lampung. Hasil pengujian menggunakan metode *Black Box* menunjukkan sistem ini bebas dari kesalahan sintaks dan mampu meningkatkan efisiensi pengelolaan koleksi [9].

Penelitian berjudul "*Digitization of Museum Collections and the Hellenic Museum Digitization Project*" dilakukan pada tahun 2021, yang mendigitalisasi koleksi Hellenic Museum menggunakan fotogrametri berbasis struktur dari gerakan (SfM). Hasil penelitian menunjukkan bahwa koleksi digital 3D ini mendukung penelitian lebih lanjut terhadap artefak budaya [10].

Penelitian berjudul "Rancang Bangun Aplikasi Museum Digital Berbasis Android" dilakukan oleh Bunari pada tahun 2022, yang bertujuan untuk mengembangkan aplikasi digital untuk Museum Sang Nila Utama. Aplikasi ini mencakup 10 klasifikasi koleksi dengan total lebih dari 300 koleksi, serta diuji menggunakan metode *usability testing*. Hasilnya menunjukkan aplikasi memenuhi lima aspek *usability*, seperti *Learnability* dan *Efficiency* [11].

Penelitian berjudul "*Digitization and Virtual Experience of Museum Collections The Virtual Tour Of The Civic Art Gallery Of Ancona*" dilakukan oleh Angeloni pada tahun 2022, yang mengeksplorasi tur virtual berbasis gambar untuk koleksi galeri seni. Penelitian ini menunjukkan bahwa tur virtual dapat meningkatkan pengalaman edukasi dan aksesibilitas pengunjung [12].

Penelitian berjudul "*A Blockchain-Based Cryptographic Interaction Method of Digital Museum Collections*" dilakukan oleh Zhao pada tahun 2023, yang mengembangkan protokol berbasis *blockchain* untuk pertukaran koleksi digital. Penelitian ini menunjukkan bahwa protokol tersebut meningkatkan keamanan dan transparansi, serta melindungi hak cipta dalam pertukaran koleksi digital [13].

Penelitian berjudul "*Digitization and the Collection Sustainability: Report on the Grand Egyptian Museum Project*" dilakukan oleh Mohammed dan Metwaly pada tahun 2023, yang bertujuan untuk mengkaji transformasi digital koleksi Grand Egyptian Museum (GEM) sebagai upaya mendukung keberlanjutan koleksi. Penelitian ini menggunakan pendekatan kuantitatif dengan pengamatan sistematis dan ulasan literatur untuk menilai strategi digitalisasi. Hasil penelitian menunjukkan bahwa digitalisasi tidak hanya meningkatkan aksesibilitas koleksi tetapi juga memungkinkan pelestarian budaya secara berkelanjutan melalui kolaborasi lintas sektor [14].

Penelitian berjudul "*VR-Driven Museum Opportunities: Digitized Archives in the Age of the Metaverse*" dilakukan oleh Gao pada tahun 2023, yang menyoroti penggunaan teknologi VR dalam arsip digital museum berbasis metaverse. Hasil penelitian menunjukkan bahwa VR dapat memberikan pengalaman interaktif yang mendalam dan meningkatkan aksesibilitas [15].

Penelitian berjudul "*Increasing the Impact of Vertebrate Scientific Collections through 3D Imaging*" dilakukan oleh Blackburn. pada tahun 2024, yang membahas pengembangan jaringan koleksi tematik *openVertebrate* (*oVert*) untuk mendigitalkan spesimen vertebrata. Penelitian ini menggunakan pencitraan CT dan fotogrametri untuk menciptakan representasi digital berkualitas tinggi dari spesimen ilmiah. Hasil penelitian menunjukkan bahwa digitalisasi 3D meningkatkan aksesibilitas koleksi untuk berbagai keperluan, termasuk penelitian, pendidikan, dan seni [16].

Penelitian berjudul "*Website-based Collection Inventory Information System Design at Wayang*" dilakukan oleh Prasuteja pada tahun 2024, yang bertujuan untuk meningkatkan efisiensi pengelolaan koleksi di Museum Wayang Sendang Mas Banyumas. Sistem ini dikembangkan menggunakan metode prototipe dengan *framework CodeIgniter* dan diuji menggunakan metode *black-box* dan *white-box* testing. Hasil pengujian menunjukkan peningkatan efisiensi dalam pelaporan data koleksi dengan tingkat kepuasan pengguna mencapai 100% [17].

Penelitian terdahulu secara keseluruhan dapat dilihat pada Tabel 2.1.

Tabel 2.1 Penelitian Terdahulu

| No. | Judul                                                                                                  | Penulis dan Tahun                                     | Metode                                                                                                                       | Hasil                                                                                                                                    |
|-----|--------------------------------------------------------------------------------------------------------|-------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| 1.  | <i>A Rapid and Cost-Effective Pipeline for Digitization of Museum Specimens with 3D Photogrammetry</i> | Joshua J. Medina, James M. Maley, dkk. (2020) [6].    | Eksperimen menggunakan fotogrametri 3D dengan perangkat keras sederhana (kamera, perangkat lunak, <i>turntable</i> otomatis) | Metode ini menghasilkan model 3D berkualitas tinggi dengan biaya rendah, meningkatkan efisiensi digitalisasi koleksi.                    |
| 2.  | <i>Designing Mobile Application for Museum Enthusiasts</i>                                             | Dewi Puspasari (2020) [7].                            | Pengembangan berbasis kebutuhan pengguna menggunakan diagram <i>use case</i> dan kuesioner terbuka.                          | Aplikasi dengan fitur <i>augmented reality</i> meningkatkan interaksi pengguna dengan museum serta daya tarik museum bagi generasi muda. |
| 3.  | <i>Creation of a Virtual Museum for the Dissemination of 3D Models of Historical Clothing</i>          | Cecile Meier, Isabel Sanchez Berriel, dkk.(2021) [8]. | Studi desain menggunakan perangkat lunak Blender, Unity, dan Marvelous Designer untuk menciptakan museum virtual.            | Museum virtual memungkinkan pengguna mengakses koleksi pakaian abad ke-16 dalam format interaktif dan terjangkau.                        |
| 4.  | Perancangan Sistem Informasi Manajemen Koleksi Museum Berbasis <i>Web</i> (Studi Kasus Museum Lampung) | Nanang Istiawan dan Nuralia (2021) [9].               | Pengembangan sistem berbasis <i>web</i> menggunakan <i>UML</i> dan diuji dengan metode <i>Black Box</i> .                    | Sistem ini meningkatkan pengelolaan koleksi museum dengan aksesibilitas yang lebih baik dan pengelolaan data yang lebih efisien.         |

Tabel 2.1 Penelitian Terdahulu (Lanjutan)

| No. | Judul                                                                                                                 | Penulis dan Tahun                         | Metode                                                                                                                                                                  | Hasil                                                                                                                                            |
|-----|-----------------------------------------------------------------------------------------------------------------------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 5.  | <i>Digitization of Museum Collections and the Hellenic Museum Digitization Project</i>                                | Tom Keep (2021) [10].                     | Digitasi menggunakan teknik <i>Structure from Motion (SfM)</i> untuk menghasilkan model digital 3D                                                                      | Koleksi digital 3D mendukung penelitian dan interaksi yang lebih besar dengan artefak museum.                                                    |
| 6.  | Rancang Bangun Aplikasi Museum Digital Berbasis Android                                                               | Bunari, Yuliska, Asyul Fikri (2022) [11]. | Pengembangan aplikasi menggunakan metode waterfall dan usability testing                                                                                                | Aplikasi ini memenuhi lima aspek <i>usability</i> , seperti <i>Learnability</i> dan Efisiensi, serta meningkatkan pengalaman digital pengunjung. |
| 7.  | <i>Digitization and Virtual Experience of Museum Collections. The Virtual Tour Of The Civic Art Gallery Of Ancona</i> | Renato Angeloni (2022) [12].              | Pendekatan berbasis gambar untuk rekonstruksi tur virtual                                                                                                               | Tur virtual museum meningkatkan pengalaman edukasi pengunjung serta memperluas aksesibilitas koleksi.                                            |
| 8.  | <i>A Blockchain-Based Cryptographic Interaction Method of Digital Museum Collections</i>                              | Liutao Zhao, dkk (2023) [13].             | Studi teknis menggunakan <i>blockchain</i> dengan algoritma SM2 <i>elliptic curve</i> dan <i>Diffie-Hellman</i> untuk mengembangkan protokol pertukaran koleksi digital | Sistem ini meningkatkan keamanan dan transparansi dalam pertukaran koleksi digital museum.                                                       |

Tabel 2.1 Penelitian Terdahulu (Lanjutan)

| No. | Judul                                                                                                   | Penulis dan Tahun                                       | Metode                                                                                                                                                                      | Hasil                                                                                                                                                  |
|-----|---------------------------------------------------------------------------------------------------------|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| 9.  | <i>Digitization and the Collection Sustainability: Report on the Grand Egyptian Museum Project</i>      | Said Nasser Mohammed & Heba Khairy Metwaly (2023) [14]. | Pendekatan kuantitatif untuk mengkaji digitalisasi koleksi, dengan penerapan teknologi seperti <i>AR</i> , <i>VR</i> , aplikasi <i>mobile</i> , dan basis data terintegrasi | Digitalisasi mendukung keberlanjutan koleksi museum dan meningkatkan keterlibatan publik.                                                              |
| 10. | <i>VR-driven Museum Opportunities: Digitized Archives in the Age of the Metaverse</i>                   | Ze Gao, Tristan Braud, dkk. (2023) [15].                | Kajian literatur sistematis dengan PRISMA, membahas peluang penggunaan teknologi <i>VR</i> dalam arsip digital dan pengalaman berbasis metaverse                            | Teknologi <i>VR</i> dan metaverse memungkinkan pengalaman arsip digital yang lebih imersif dan meningkatkan keterlibatan pengunjung secara signifikan. |
| 11. | <i>Website-based Collection Inventory Information System Design at Wayang</i>                           | Soni Gunawan Prasuteja, Adhi Wibowo, dkk. (2024) [16].  | Metode prototipe dengan alat bantu <i>CodeIgniter</i> , diuji menggunakan metode <i>black-box</i> dan <i>white-box testing</i>                                              | Sistem berbasis <i>web</i> ini meningkatkan efisiensi pengelolaan koleksi museum dan tingkat kepuasan pengguna mencapai 100%.                          |
| 12. | <i>Increasing the Impact of Vertebrate Scientific Collections through 3D Imaging: The oVert Network</i> | David C. Blackburn, dkk (2024) [17].                    | Kolaborasi antar-lembaga menggunakan pencitraan CT dan fotogrametri untuk menciptakan representasi digital spesimen vertebrata                                              | Koleksi digital mendukung penelitian evolusi, pendidikan, seni, dan pelestarian dengan aksesibilitas lebih luas.                                       |

Penelitian terdahulu menunjukkan bahwa digitalisasi museum menjadi tren penting dalam pelestarian dan pengelolaan koleksi. Studi-studi sebelumnya telah mengeksplorasi berbagai teknologi seperti fotogrametri 3D, aplikasi *mobile*, *blockchain*, *virtual reality (VR)*, dan sistem berbasis web untuk meningkatkan aksesibilitas dan efisiensi dalam pengelolaan koleksi museum. Perbedaan utama dengan penelitian yang dilakukan oleh penulis adalah pendekatan teknologi yang digunakan serta fokus pengembangan sistem. Penelitian ini menggunakan *PERN Stack (PostgreSQL, Express.js, React.js, dan Node.js)* untuk membangun sistem berbasis web yang mendukung digitalisasi koleksi Museum Lampung. Fitur tambahan seperti integrasi *QR Code* dan video 360° juga menjadi pembeda utama dari penelitian sebelumnya yang dapat dilihat pada *roadmap* dari penelitian yang akan dilakukan pada Gambar 2.1.



Gambar 2.1 Roadmap Penelitian

## **2.2 Digitalisasi dan Museum**

Dalam era transformasi digital, digitalisasi menjadi salah satu langkah strategis yang dilakukan oleh berbagai institusi, termasuk museum. Digitalisasi tidak hanya mengubah cara penyimpanan dan penyebaran informasi, tetapi juga membuka akses yang lebih luas terhadap koleksi budaya yang sebelumnya terbatas secara fisik. Oleh karena itu, pemahaman terhadap konsep digitalisasi dan peran museum dalam pelestarian budaya menjadi dasar penting dalam pengembangan sistem digitalisasi koleksi museum.

### **2.2.1 Digitalisasi**

Digitalisasi merupakan proses konversi informasi dari format analog ke format digital. Proses ini melibatkan pengubahan berbagai jenis data, termasuk teks, gambar, audio, dan video, menjadi format yang dapat diproses, disimpan, dan ditransmisikan oleh sistem komputer. Digitalisasi telah menjadi bagian integral dari transformasi teknologi di berbagai sektor, termasuk bisnis, pendidikan, dan pelestarian budaya [18].

### **2.2.2 Manajemen Museum**

Manajemen museum adalah disiplin yang mencakup perencanaan, pengorganisasian, pengarahan, dan pengendalian sumber daya museum untuk mencapai tujuan pelestarian, penelitian, dan pendidikan. Ini melibatkan berbagai aspek operasional, termasuk pengelolaan koleksi, pengembangan pameran, manajemen sumber daya manusia, pemasaran, dan penggalangan dana.

Salah satu aspek kunci dalam manajemen museum modern adalah pengembangan strategi digital. Ini mencakup implementasi teknologi untuk meningkatkan pengalaman pengunjung, memperluas jangkauan museum melalui platform *online*, dan meningkatkan efisiensi operasional. Strategi digital juga melibatkan penggunaan media sosial dan platform digital lainnya untuk menjangkau pengunjung yang lebih luas dan melibatkan mereka dalam dialog tentang koleksi dan program museum [19].

### 2.2.3 Museum

Museum adalah institusi permanen, nirlaba yang melayani masyarakat dan perkembangannya, terbuka untuk umum, yang mengakuisisi, melestarikan, meneliti, mengkomunikasikan, dan memamerkan warisan berwujud dan tak berwujud kemanusiaan dan lingkungannya untuk tujuan pendidikan, studi, dan kesenangan. Museum memainkan peran penting dalam melestarikan dan mempresentasikan warisan budaya, sejarah, dan ilmu pengetahuan kepada publik. Fungsi utama museum meliputi pengumpulan, pelestarian, penelitian, dan pameran. Museum mengakuisisi dan merawat objek-objek yang memiliki nilai budaya, historis, atau ilmiah. Dalam era digital, museum menghadapi tantangan dan peluang baru. Digitalisasi koleksi memungkinkan akses yang lebih luas dan pelestarian jangka panjang. Museum juga mengadopsi teknologi interaktif seperti realitas virtual dan *augmented* untuk meningkatkan pengalaman pengunjung. Selain itu, platform *online* memungkinkan museum untuk menjangkau pengunjung global, menawarkan tur virtual, dan melibatkan pengunjung melalui media sosial [20].

### 2.2.4 Koleksi Museum

Koleksi museum merupakan inti dari institusi museum, terdiri dari objek-objek yang memiliki nilai budaya, historis, artistik, atau ilmiah. Koleksi ini dapat mencakup berbagai jenis artefak, spesimen, dokumen, dan karya seni yang dikumpulkan, dipelihara, dan dipelajari oleh museum. Koleksi museum berfungsi sebagai sumber daya penting untuk penelitian, pendidikan, dan pameran, memungkinkan museum untuk memenuhi misi mereka dalam melestarikan dan menginterpretasikan warisan budaya dan alam [21].

## 2.3 Teknologi dalam Pengembangan Sistem

Pengembangan sistem digitalisasi museum memerlukan pendekatan yang terstruktur dan sistematis agar dapat mengakomodasi kompleksitas data dan kebutuhan pengguna. Oleh karena itu, kerangka kerja seperti *SDLC* dan pendekatan *fullstack* menjadi landasan penting dalam perancangan dan implementasi solusi digital yang efektif.

### 2.3.1 *SDLC (Software Development Life Cycle)*

*Software Development Life Cycle (SDLC)* merupakan kerangka kerja sistematis yang membagi proses pembangunan perangkat lunak ke dalam sejumlah tahapan yang berurutan. Fase-fase tersebut meliputi perencanaan, analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Setiap fase dalam *SDLC* dirancang untuk meningkatkan efisiensi serta kualitas perangkat lunak yang dihasilkan, sekaligus meminimalisasi risiko kegagalan proyek. Penerapan model *SDLC* seperti *Waterfall*, *Agile*, dan *Iterative* memungkinkan pengembang memilih pendekatan sesuai kompleksitas dan kebutuhan sistem. Model ini memberikan kejelasan dalam perencanaan proyek, manajemen waktu, serta dokumentasi sistematis yang sangat berguna untuk pengembangan skala besar [22].

### 2.3.2 *Fullstack Development*

*Fullstack development* merupakan pendekatan menyeluruh dalam rekayasa perangkat lunak yang mengintegrasikan seluruh lapisan aplikasi mulai dari *front-end* sebagai antarmuka pengguna, *back-end* yang menangani logika dan proses bisnis, hingga manajemen basis data dalam satu alur kerja terpadu. Pada lapisan *front-end*, framework seperti *React.js*, *Angular*, dan *Vue.js* menghadirkan arsitektur berbasis komponen yang dinamis; di sisi *back-end*, kerangka kerja seperti *Express.js (Node.js)*, *Django (Python)*, dan *Ruby on Rails* memfasilitasi *routing*, *middleware*, dan *ORM*; sedangkan untuk penyimpanan data, pilihan umum meliputi *PostgreSQL*, *MongoDB*, dan *MySQL* untuk kebutuhan relasional maupun *NoSQL*. Kombinasi teknologi ini sering kali dirangkum dalam susunan populer seperti *MEAN (MongoDB, Express.js, Angular, Node.js)*, *MERN (MongoDB, Express.js, React.js, Node.js)*, dan *PERN (PostgreSQL, Express.js, React.js, Node.js)*, sehingga *fullstack developer* dapat mengelola integrasi antarlapisan, otomasi *deployment*, serta optimasi performa secara *end-to-end* [23].

## 2.4 Teknologi dan Alat Pendukung

Dalam membangun sistem digitalisasi koleksi museum yang efektif, dibutuhkan berbagai alat bantu pengembangan yang mampu mempercepat proses pengkodean, pengujian, dan visualisasi.

#### 2.4.1 *Visual Studio Code*

*Visual Studio Code (VS Code)* adalah editor kode sumber yang ringan namun kuat, yang dikembangkan oleh *Microsoft*. *VS Code* mendukung berbagai bahasa pemrograman dan menawarkan fitur-fitur canggih seperti penyelesaian kode otomatis, *debugging* terintegrasi, dan kontrol versi *Git*. Dalam pengembangan *web* dan aplikasi, *VS Code* menjadi pilihan populer karena integrasinya yang mulus dengan teknologi *frontend* dan *backend*. Dukungannya untuk *JavaScript*, *TypeScript*, dan berbagai kerangka kerja *web* modern, serta kemampuannya untuk menangani operasi *backend* seperti manajemen *database* dan *API*, menjadikannya alat yang serbaguna untuk pengembangan *fullstack* [24].

#### 2.4.2 *Postman*

*Postman* adalah platform kolaborasi untuk pengembangan *API* yang memungkinkan pengembang untuk merancang, membangun, menguji, dan mendokumentasikan *API* dengan efisien. Awalnya dikembangkan sebagai ekstensi *browser* sederhana, *Postman* telah berkembang menjadi sebuah alat yang komprehensif yang mendukung seluruh siklus hidup pengembangan *API*. Salah satu fitur utama *Postman* adalah kemampuannya untuk membuat dan mengirim permintaan *HTTP* ke *endpoint API* dengan mudah. Ini memungkinkan pengembang untuk menguji *API* secara manual dan otomatis, dengan dukungan untuk berbagai metode *HTTP*, *header*, dan *payload*. *Postman* juga menawarkan lingkungan variabel yang memudahkan pengujian di berbagai konfigurasi dan tahap pengembangan [25].

#### 2.4.3 *Browser*

*Browser*, atau peramban *web*, adalah perangkat lunak aplikasi yang digunakan untuk mengakses dan menampilkan informasi dari *World Wide Web*. *Browser* berfungsi sebagai antarmuka antara pengguna dan konten *web*, menerjemahkan kode *HTML*, *CSS*, dan *JavaScript* menjadi halaman *web* yang dapat dibaca dan berinteraksi dengan pengguna. Perkembangan *browser* telah mengubah cara kita mengakses dan berinteraksi dengan informasi *online* [26].

## 2.5 Komponen Sistem *PERN Stack*

*PERN Stack* (*PostgreSQL*, *Express.js*, *React.js*, dan *Node.js*) merupakan kombinasi teknologi *open-source* yang saling terintegrasi untuk membangun aplikasi *web* dinamis dan responsif. *Stack* ini dipilih karena kemampuannya dalam menyediakan alur kerja yang efisien, skalabilitas tinggi, serta fleksibilitas dalam pengelolaan data dan antarmuka pengguna.

### 2.5.1 *PostgreSQL*

*PostgreSQL* merupakan sistem manajemen basis data relasional *open-source* yang mendukung pendekatan *object-relational database management system* (*ORDBMS*). *PostgreSQL* menawarkan keunggulan dalam hal integritas data, skalabilitas, serta kepatuhan terhadap standar *SQL*. Sistem ini memiliki fitur-fitur canggih seperti dukungan terhadap tipe data kompleks, kemampuan *indexing* beragam (*B-tree*, *GiST*, *GIN*), hingga pemrograman prosedural menggunakan *PL/pgSQL*. *PostgreSQL* juga mendukung transaksi *ACID*, replikasi data, serta pengolahan data *JSON*, menjadikannya sangat relevan untuk aplikasi berskala besar, termasuk dalam pengelolaan sistem informasi berbasis *web* [27].

### 2.5.2 *Express.js*

*Express.js* adalah kerangka kerja aplikasi *web* minimalis dan fleksibel untuk *Node.js* yang menyediakan serangkaian fitur robust untuk membangun aplikasi *web* dan *mobile*. Arsitektur *Express.js* dibangun di atas konsep *middleware*, yang merupakan fungsi-fungsi yang memiliki akses ke objek permintaan (*req*), objek respons (*res*), dan fungsi *middleware* berikutnya dalam siklus permintaan-*respons* aplikasi. *Middleware* dapat melakukan berbagai tugas seperti mengeksekusi kode, membuat perubahan pada objek permintaan dan *respons*, mengakhiri siklus permintaan-*respons*, atau memanggil *middleware* berikutnya dalam tumpukan. Pendekatan ini memberikan fleksibilitas yang luar biasa dalam menangani permintaan *HTTP* dan memungkinkan pengembang untuk dengan mudah menambahkan fungsionalitas ke aplikasi mereka [28].

### 2.5.3 *React.js*

*React.js*, atau lebih dikenal sebagai *React.js*, adalah sebuah library JavaScript *open-source* yang dikembangkan oleh Facebook untuk membangun antarmuka pengguna (UI) yang interaktif dan efisien. Pendekatan berbasis komponen dan konsep *Virtual DOM* yang diusung oleh *React.js* memungkinkan pengembang untuk menciptakan aplikasi web yang kompleks dengan performa tinggi dan mudah dipelihara [29].

### 2.5.4 *Node.js*

*Node.js* merupakan lingkungan *runtime JavaScript* yang bersifat *open-source* dan *cross-platform*, dirancang untuk mengeksekusi kode *JavaScript* di luar lingkungan browser. Arsitektur *Node.js* yang berbasis *event-driven* dan *non-blocking I/O* memungkinkan pengembang untuk membangun aplikasi yang *scalable* dan efisien dalam penggunaan sumber daya [27].

## 2.6 Database

*Database*, atau basis data, adalah kumpulan terorganisir dari data yang disimpan dan diakses secara elektronik. Basis data dirancang untuk mengelola, menyimpan, dan mengambil informasi secara efisien, dan merupakan komponen fundamental dari sebagian besar sistem informasi modern. Basis data dapat bervariasi dalam kompleksitas dan ukuran, mulai dari daftar sederhana hingga struktur yang sangat kompleks dengan jutaan catatan. Sistem Manajemen Basis Data (DBMS) adalah perangkat lunak yang mengelola basis data, menyediakan antarmuka antara data dan pengguna atau aplikasi. DBMS menangani tugas-tugas seperti penyimpanan data, pengambilan, pembaruan, dan penghapusan, serta menjaga integritas dan keamanan data. Jenis DBMS yang umum termasuk basis data relasional (seperti *MySQL*, *PostgreSQL*), basis data *NoSQL* (seperti *MongoDB*, *Cassandra*), dan basis data berorientasi grafik (seperti *Neo4j*) [30].

## 2.7 REST API

*REST (Representational State Transfer) API* adalah pendekatan arsitektur layanan *web* yang memungkinkan interaksi antara klien dan server melalui protokol *HTTP*. *REST API* memanfaatkan metode standar *HTTP* seperti *Get*, *Post*, *Put*, dan *Delete*, serta menerapkan prinsip *stateless* yang berarti setiap permintaan bersifat mandiri tanpa menyimpan informasi sesi di sisi server. *REST API* dikenal ringan dan fleksibel, menjadikannya pilihan utama dalam pengembangan aplikasi modern termasuk sistem berbasis *microservices*. Pendekatan ini juga memungkinkan integrasi antar sistem yang lebih efisien serta pengembangan *frontend-backend* yang terpisah namun sinkron [28].

## 2.8 Kamera

Kamera merupakan perangkat yang digunakan untuk merekam gambar atau video dari dunia nyata. Dalam era digital, kamera telah mengalami perkembangan signifikan dari bentuk analog menjadi digital, memungkinkan penyimpanan dan pemrosesan gambar secara elektronik. Kamera digital modern menggunakan sensor gambar elektronik untuk menangkap cahaya dan mengonversinya menjadi sinyal digital yang dapat disimpan dan dimanipulasi. Kamera 360<sup>0</sup> dan teknologi *light field* juga membuka kemungkinan baru dalam pengambilan dan pemrosesan gambar, memungkinkan manipulasi pasca-pengambilan yang lebih fleksibel [31]. Jenis-jenis kamera dan karakteristiknya dapat dilihat pada tabel 2.2.

Tabel 2.2 Jenis-Jenis Kamera

| No. | Jenis Kamera           | Karakteristik                               | Keterangan Penggunaan                  |
|-----|------------------------|---------------------------------------------|----------------------------------------|
| 1.  | <i>DSLR</i>            | Sensor besar, lensa yang dapat diganti      | Fotografi profesional                  |
| 2.  | <i>Mirrorless</i>      | Ukuran kompak, kualitas setara <i>DSLR</i>  | Fotografi profesional dan enthusiast   |
| 3.  | <i>Point-and-shoot</i> | Ukuran kecil, mudah digunakan               | Penggunaan sehari-hari                 |
| 4.  | <i>Smartphone</i>      | Terintegrasi dengan perangkat <i>mobile</i> | Fotografi kasual, sosial media         |
| 5.  | <i>Action Camera</i>   | Tahan air, tahan guncangan                  | Olahraga ekstrem, <i>underwater</i>    |
| 6.  | <i>360-degree</i>      | Merekam seluruh area sekitar                | <i>Virtual tour</i> , realitas virtual |

## 2.9 Multer

*Multer* adalah *middleware Node.js* yang dirancang khusus untuk menangani data *multipart/form-data*, yang umumnya digunakan untuk mengunggah *file* dalam aplikasi *web*. Dikembangkan sebagai solusi yang efisien dan mudah digunakan untuk mengelola unggahan *file* dalam lingkungan *Express.js*, *Multer* telah menjadi komponen penting dalam pengembangan aplikasi *web* modern yang memerlukan fungsionalitas unggah *file*. *Multer* bekerja dengan memarsing *form* data yang masuk dan menyimpan *file* yang diunggah ke lokasi yang ditentukan, sambil menyediakan informasi tentang *file* tersebut kepada aplikasi [32].

## 2.10 Otentikasi

Otentikasi merupakan proses verifikasi identitas pengguna sebelum diberikan akses ke sumber daya sistem. Dalam pengembangan aplikasi *web* modern, metode otentikasi berbasis token seperti *JSON Web Token (JWT)* banyak digunakan karena mendukung skema otentikasi tanpa status (*stateless authentication*). Berikut langkah-langkah proses otentikasi berbasis token:

1. Pengguna mengirimkan permintaan *login* beserta kredensial (*username* dan *password*) melalui antarmuka aplikasi.
2. Server menerima permintaan dan memverifikasi kredensial dengan mencocokkannya terhadap data yang tersimpan di basis data.
3. Jika kredensial valid, server menghasilkan token otentikasi (*JWT*) yang memuat informasi identitas dan hak akses pengguna, lalu mengenkripsi token tersebut.
4. Token dikirimkan kembali ke pengguna dan disimpan di sisi klien (misal pada *local storage* atau *session storage*).
5. Pada setiap permintaan selanjutnya yang memerlukan otorisasi, pengguna menyisipkan token *JWT* pada *header* permintaan (biasanya di *Authorization header*).
6. Server memverifikasi validitas token pada setiap permintaan. Jika token valid dan belum kedaluwarsa, server memberikan akses ke sumber daya sesuai hak akses pengguna.

Penerapan otentikasi yang kuat sangat penting untuk melindungi data sensitif serta

membatasi hak akses berdasarkan peran pengguna dalam sistem [33].

### 2.11 Keamanan Data dan Enkripsi *Password*

Keamanan data, khususnya *password*, sangat krusial dalam sistem informasi modern. Salah satu metode yang umum digunakan untuk mengamankan *password* adalah enkripsi satu arah melalui algoritma *hashing bcrypt*. Berikut adalah langkah-langkah proses *hashing password* menggunakan *bcrypt*:

1. Pengguna melakukan registrasi dan memasukkan *password* pada antarmuka aplikasi.
2. Sistem secara otomatis menghasilkan *salt*, yaitu data acak yang digabungkan dengan *password* sebelum proses *hashing*.
3. *Password* dan *salt* diproses menggunakan algoritma *hashing bcrypt* untuk menghasilkan *hash password*.
4. *Hash password* yang dihasilkan (bukan *password* asli) disimpan di basis data.
5. Saat *login*, *password* yang dimasukkan pengguna digabungkan dengan *salt* yang sama dan diproses kembali menggunakan algoritma *bcrypt*.
6. Sistem membandingkan *hash* hasil *input login* dengan *hash* yang tersimpan di basis data. Jika cocok, otentikasi berhasil.

Penggunaan *bcrypt* memberikan keunggulan berupa penambahan *salt* secara otomatis dan dukungan pengaturan kompleksitas melalui *cost factor*, sehingga hasil akhir menjadi unik walaupun *password*-nya sama [33].

III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan di Laboratorium Terpadu Teknik Elektro, Fakultas Teknik, Universitas Lampung dan Museum Lampung, dengan fokus pada pengembangan dan pengujian sistem informasi berbasis *web* untuk koleksi museum. Penelitian berlangsung selama Desember 2024 hingga Juli 2025.

3.2 Alat dan Bahan

Adapun alat dan bahan yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.1.

Tabel 3.1 Alat dan Bahan

| No | Perangkat                 | Spesifikasi                                    | Kegunaan                                              | Keterangan                                         |
|----|---------------------------|------------------------------------------------|-------------------------------------------------------|----------------------------------------------------|
| 1. | Laptop                    | Prosesor AMD Ryzen 5 5500U, RAM 8GB, SSD 256GB | Untuk pengembangan <i>frontend</i> dan <i>backend</i> | Digunakan sebagai perangkat utama pengembangan     |
| 2. | <i>Visual Studio Code</i> | Versi 1.96.4                                   | Penulisan dan <i>debugging</i> kode                   | Mendukung integrasi pengembangan sistem            |
| 3. | <i>PostgreSQL</i>         | Versi 17.4-1                                   | Menyimpan data koleksi museum                         | <i>Database</i> untuk mendigitalisasi koleksi      |
| 4. | <i>Postman</i>            | Versi 11.23.3                                  | Menguji rute <i>API</i> pada backend                  | Memastikan <i>API</i> berjalan sesuai fungsi       |
| 5. | Kamera                    | Insta360 ONE RS                                | Mendokumentasikan koleksi untuk basis data museum     | Mendukung digitalisasi visual koleksi              |
| 6. | Browser                   | Chrome Versi 131.0.6778.266                    | Menguji antarmuka <i>frontend</i>                     | Digunakan untuk memastikan kompatibilitas aplikasi |

Tabel 3.1 Alat dan Bahan (Lanjutan)

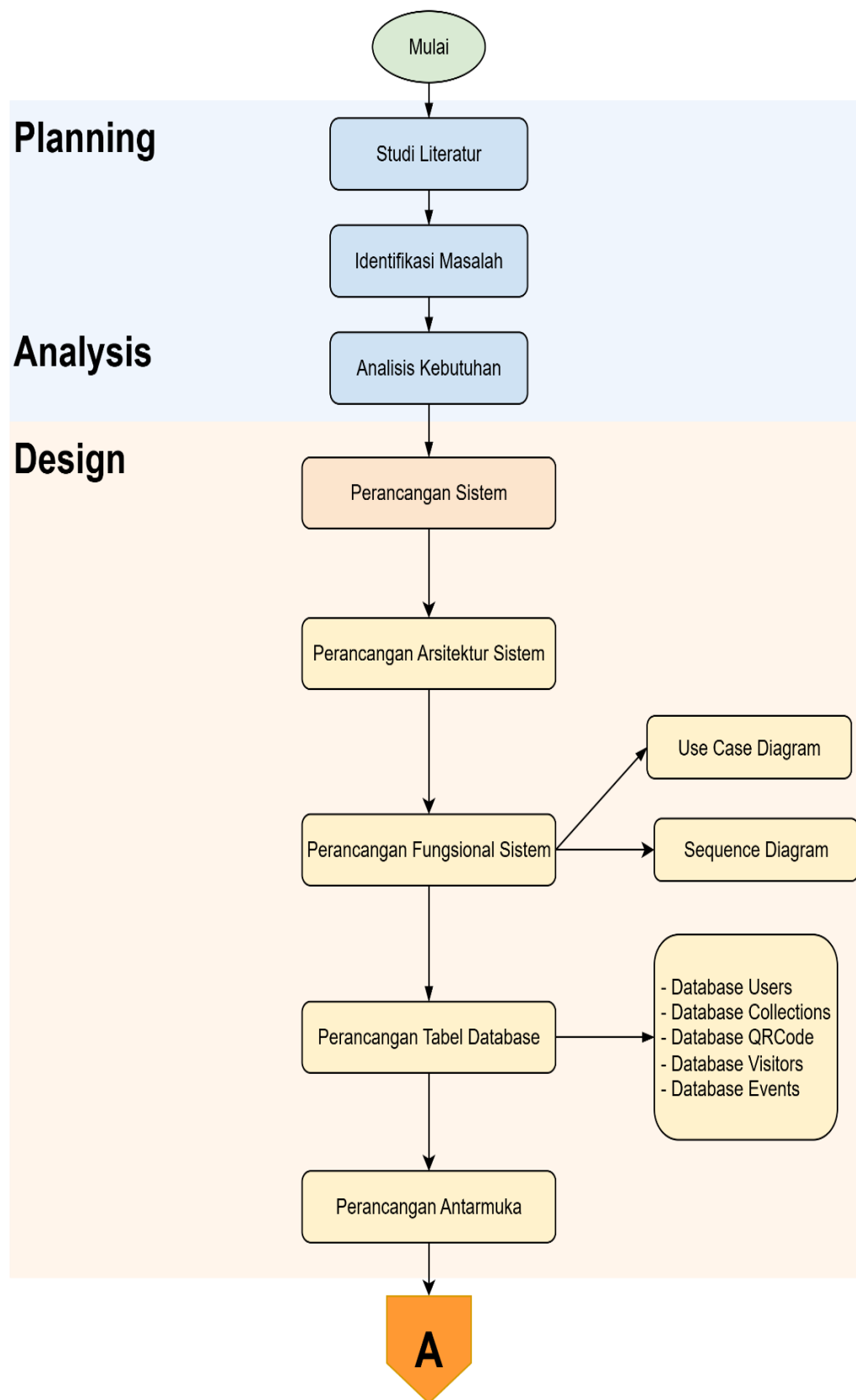
| No  | Perangkat             | Spesifikasi                     | Kegunaan                                                    | Keterangan                                                                     |
|-----|-----------------------|---------------------------------|-------------------------------------------------------------|--------------------------------------------------------------------------------|
| 7.  | <i>Node.js</i>        | <i>Node.js</i> versi 18.20.3    | Sebagai runtime untuk pengembangan <i>backend</i>           | Mendukung operasi <i>server-side</i>                                           |
| 8.  | <i>React.js</i>       | Versi 18.3.1                    | Membangun antarmuka pengguna                                | Digunakan untuk <i>frontend</i> dinamis                                        |
| 9.  | <i>Express.js</i>     | Versi 4.21.1                    | Mengelola rute <i>backend</i>                               | Membantu komunikasi <i>frontend-backend</i>                                    |
| 10. | <i>Multer</i>         | Versi 1.4.5                     | Mengunggah data multimedia                                  | Mempermudah unggahan gambar koleksi                                            |
| 11. | Jest                  | Versi 30.0.4                    | Pengujian unit dan integrasi kode <i>JavaScript</i>         | Memudahkan pengujian otomatis dalam pengembangan software                      |
| 12. | Cloudinary            | Platform layanan cloud          | Menyimpan dan mengelola media (gambar/video) berbasis cloud | Menyediakan penyimpanan media dengan optimasi dan <i>CDN</i> untuk akses cepat |
| 13. | <i>REST API</i>       | Representational State Transfer | Standar komunikasi data <i>client-server</i>                | Mendefinisikan pertukaran data <i>HTTP</i> terstruktur                         |
| 14. | <i>Bcrypt</i>         | Versi 6.0.0                     | Enkripsi (hash) kata sandi                                  | Mengamankan penyimpanan kata sandi pengguna                                    |
| 15. | <i>Dotenv</i>         | Versi 17.2.0                    | Memuat variabel lingkungan                                  | Menyimpan konfigurasi rahasia (misal kunci <i>API</i> ) dari <i>file .env</i>  |
| 16. | <i>Cors</i>           | Versi 2.8.5                     | Mengaktifkan permintaan lintas domain                       | Memungkinkan frontend mengakses <i>API</i> secara aman                         |
| 17. | <i>Json Web Token</i> | Versi 9.0.2                     | Pembuatan dan verifikasi token JWT                          | Mendukung autentikasi dan otorisasi pengguna                                   |
| 18. | <i>Sequelize</i>      | Versi 6.37.7                    | <i>ORM</i> untuk database relasional SQL                    | Menyederhanakan akses dan manipulasi database relasional                       |
| 19. | <i>Supertest</i>      | Versi 7.1.3                     | Pengujian rute API HTTP                                     | Memfasilitasi pengujian <i>endpoint</i> server dalam kode                      |

Tabel 3.1 Alat dan Bahan (Lanjutan)

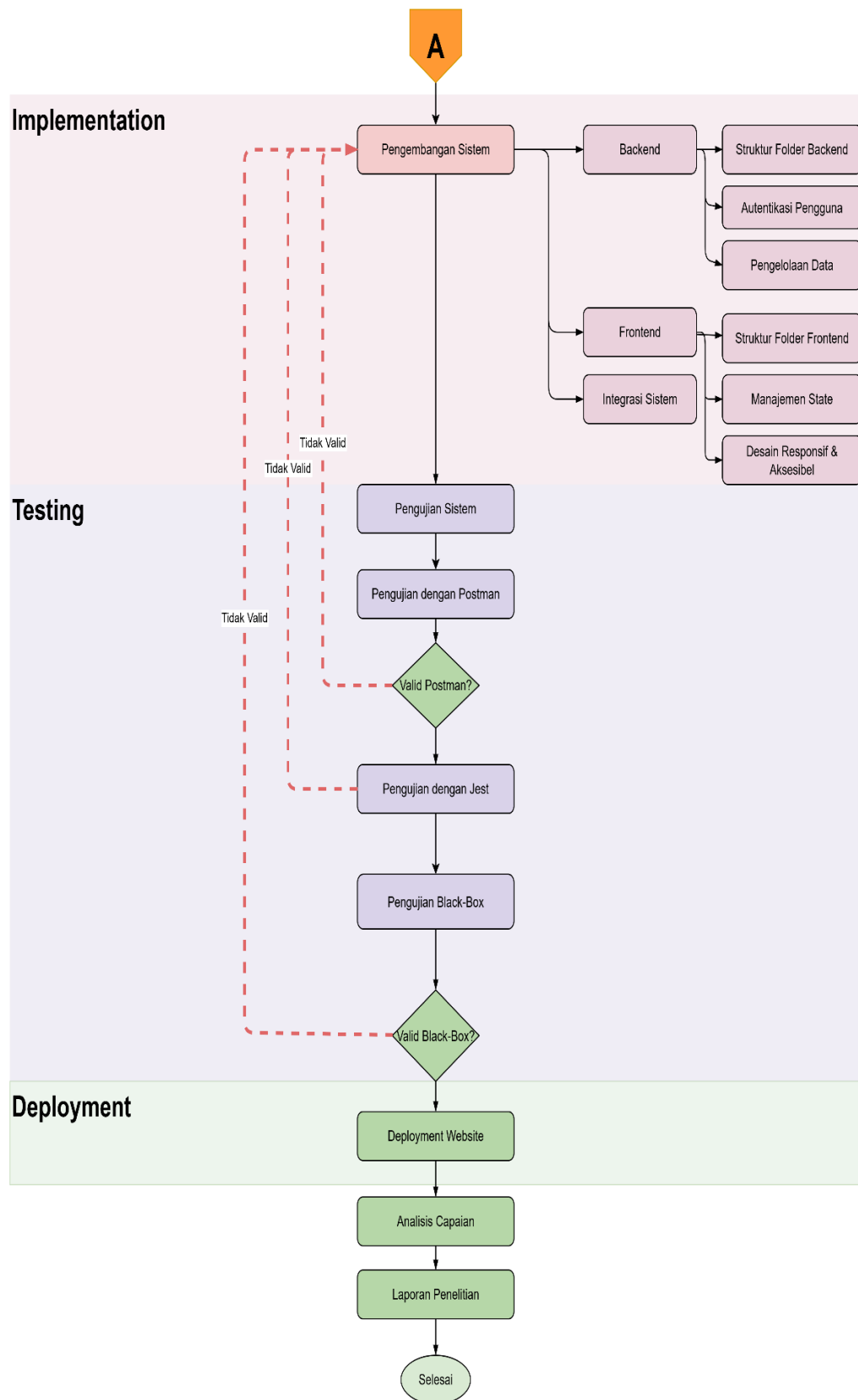
| No  | Perangkat | Spesifikasi                               | Kegunaan                                           | Keterangan                                                                                   |
|-----|-----------|-------------------------------------------|----------------------------------------------------|----------------------------------------------------------------------------------------------|
| 20. | Railway   | Platform <i>deployment</i> berbasis cloud | Menyimpan dan menjalankan aplikasi <i>backend</i>  | Menyediakan hosting <i>Node.js</i> dan <i>PostgreSQL</i> secara otomatis dengan <i>CI/CD</i> |
| 21  | Vercel    | Platform serverless hosting               | Menyimpan dan menjalankan aplikasi <i>frontend</i> | Melakukan <i>build</i> otomatis dan menyajikan <i>frontend React</i> dengan optimal          |

### 3.3 Diagram Alir Penelitian

Penelitian ini menggunakan metode *SDLC* untuk mendukung pengembangan sistem secara bertahap. Adapun tahap-tahap yang dilakukan untuk menyusun penelitian ini dapat dilihat pada Gambar 3.1



Gambar 3.1 Diagram Alir Penelitian



Gambar 3.1 Diagram Alir Penelitian (lanjutan)

### 3.3.1 Studi Literatur

Studi literatur diawali dengan mengumpulkan dan mengkaji referensi dari jurnal ilmiah, buku, dan laporan penelitian terdahulu yang relevan dengan digitalisasi museum dan pengembangan perangkat lunak berbasis teknologi *fullstack*. Studi ini dilakukan untuk memahami tren teknologi terkini, teknologi yang digunakan dalam proyek serupa, serta sebagai dasar teoritis dalam membangun sistem yang akan dikembangkan. Selain itu, studi literatur juga menjadi referensi dalam pemilihan arsitektur sistem dan teknologi pendukung seperti *PostgreSQL*, *React.js*, dan *Node.js*.

### 3.3.2 Identifikasi Masalah

Identifikasi dilakukan dengan menelaah kondisi aktual pengelolaan koleksi di Museum Lampung berdasarkan data umum dari publikasi dan dokumentasi instansi. Museum Lampung diketahui belum memiliki sistem digital yang memadai untuk mengelola koleksi secara daring, sehingga proses pendataan koleksi masih dilakukan secara manual dengan catatan fisik atau *spreadsheet* sederhana. Kondisi ini berpotensi menyebabkan kehilangan data, kesulitan dalam pencarian koleksi, serta keterbatasan akses informasi bagi publik. Permasalahan ini kemudian dianalisis lebih lanjut untuk merumuskan kebutuhan sistem digital yang dapat menjawab tantangan tersebut. Identifikasi masalah ini menjadi landasan dalam menentukan ruang lingkup dan kebutuhan sistem yang akan dibangun.

### 3.3.3 Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan untuk mengidentifikasi permasalahan dalam sistem pengelolaan koleksi di Museum Lampung serta merumuskan solusi berbasis teknologi yang sesuai dengan kebutuhan pengguna. Analisis ini menjadi dasar dalam proses perancangan dan pengembangan sistem digitalisasi koleksi museum yang dapat dilihat pada Tabel 3.2.

Tabel 3.2 Analisis Kebutuhan

| No. | Permasalahan yang Ditemukan                                                                       | Dampak                                                  | Solusi yang Diberikan                                                                                 |
|-----|---------------------------------------------------------------------------------------------------|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| 1   | Pendataan koleksi masih dilakukan secara manual menggunakan catatan fisik atau <i>spreadsheet</i> | Risiko kehilangan atau kerusakan data                   | Membangun sistem basis data terstruktur menggunakan <i>PostgreSQL</i>                                 |
| 2   | Informasi koleksi tidak tersedia secara digital dan belum dapat diakses oleh publik secara daring | Akses terhadap data koleksi terbatas                    | Mengembangkan aplikasi <i>web</i> berbasis <i>React.js</i> dan <i>Node.js</i> untuk publikasi koleksi |
| 3   | Koleksi belum terdokumentasi secara visual dalam bentuk gambar atau video                         | Kurangnya daya tarik dan informasi visual koleksi       | Menyediakan fitur upload gambar dan video 360° menggunakan <i>Multer</i>                              |
| 4   | Pengunjung kesulitan menemukan koleksi tertentu secara cepat                                      | Menurunnya pengalaman pengguna dalam eksplorasi koleksi | Menyediakan fitur pencarian dan filter berdasarkan nama, tahun, dan kategori                          |
| 5   | Pengelolaan data koleksi seperti pembaruan atau penghapusan dilakukan secara manual               | Sulit melakukan pelacakan histori atau perubahan data   | Menyediakan fitur <i>CRUD</i> koleksi yang dapat diakses oleh admin melalui dashboard                 |

### 3.3.4 Perancangan Sistem

Perancangan sistem dilakukan setelah kebutuhan teridentifikasi, dengan tujuan merancang struktur teknis, alur interaksi pengguna, dan basis data. Proses ini mencakup perancangan fungsional sistem, struktur *database*, arsitektur sistem, dan antarmuka pengguna.

#### 3.3.4.1 Arsitektur Sistem

Arsitektur sistem yang dikembangkan dalam penelitian ini menggunakan pendekatan *client-server* berbasis *PERN Stack* (*PostgreSQL*, *Express.js*, *React.js*, dan *Node.js*). Sistem ini dirancang secara modular untuk mendukung digitalisasi dan manajemen koleksi Museum Lampung. Arsitektur ini mengintegrasikan lima komponen utama yang saling berinteraksi secara dua arah untuk menjamin keandalan sistem.

##### 1. Antarmuka Pengguna (*Frontend*)

*Frontend* dibangun menggunakan *React.js* dan berfungsi sebagai antarmuka interaktif antara sistem dan pengguna, baik pengunjung umum, admin, maupun superadmin. Melalui antarmuka ini, pengguna dapat melihat daftar koleksi,

melakukan pencarian, menggunakan *filter* berdasarkan tahun atau kategori, serta mengakses halaman detail koleksi. Selain itu, admin dan superadmin memiliki akses ke dashboard pengelolaan untuk manajemen koleksi dan akun pengguna.

## 2. *REST API*

*REST API* yang dibangun menggunakan *Express.js* menjadi pintu utama komunikasi antara *frontend* dan *backend*. Seluruh permintaan dari *frontend* dikirim ke *REST API* menggunakan protokol *HTTP* dengan metode seperti *GET*, *POST*, *PUT*, dan *DELETE*. *REST API* ini bertugas meneruskan permintaan ke *backend* dan mengembalikan hasil pemrosesan dalam format *JSON*. Selain itu, *REST API* juga menangani proses otorisasi berdasarkan peran pengguna.

## 3. *Backend* (Server Aplikasi)

*Backend* dikembangkan dengan *Node.js* dan berperan penting dalam menangani seluruh proses logika aplikasi. Setelah menerima permintaan dari *REST API*, *backend* melakukan validasi data, mengelola autentikasi menggunakan *JSON Web Token* (JWT), serta menjalankan operasi terhadap data koleksi dan pengguna. *Backend* juga bertanggung jawab atas pengelolaan unggah *file* media ke layanan penyimpanan eksternal seperti Cloudinary, serta memastikan data siap ditransfer ke basis data.

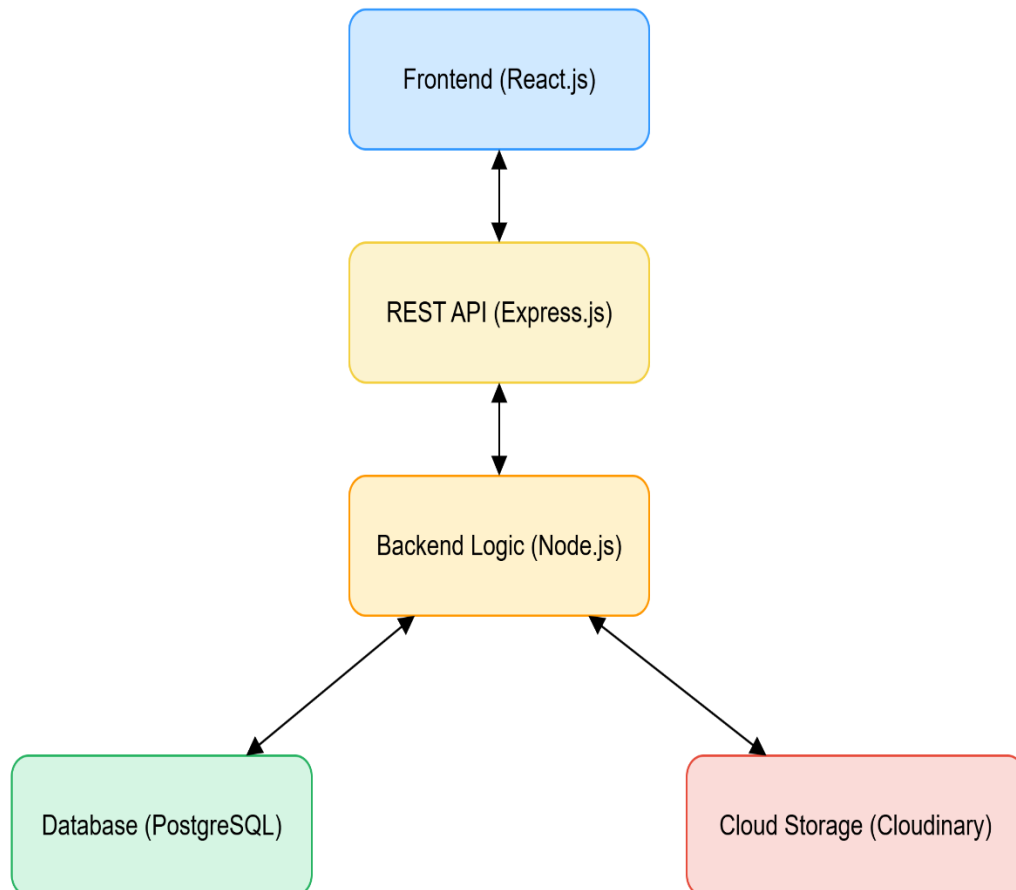
## 4. Basis Data (*Database*)

Sistem menggunakan *PostgreSQL* sebagai basis data relasional untuk menyimpan data terstruktur, termasuk data koleksi, informasi pengguna, kategori, tahun, serta *URL file* media dan *QR Code*. Relasi antartabel dirancang dengan prinsip normalisasi untuk menjaga integritas data, efisiensi akses, dan fleksibilitas pengembangan.

## 5. Penyimpanan Media Berbasis *Cloud* (Cloudinary)

Seluruh *file* media seperti gambar koleksi, video 360°, dan *QR Code* disimpan di layanan Cloudinary. *Backend* bertugas mengunggah *file* ke Cloudinary dan menyimpan *URL file* tersebut ke dalam *database*. Pendekatan ini dipilih karena memberikan keuntungan dari sisi skalabilitas, kecepatan akses, serta efisiensi penyimpanan dibanding menyimpan *file* langsung di server.

Kelima komponen tersebut berkomunikasi secara dua arah, dimulai dari *frontend* yang mengirimkan permintaan ke *REST API*, diteruskan ke *backend*, dan selanjutnya diproses melalui akses ke basis data atau layanan penyimpanan media. Respons dari *backend* akan kembali ke *frontend* dalam bentuk informasi yang ditampilkan kepada pengguna secara *real-time*. Struktur lengkap dari arsitektur sistem ini ditampilkan pada Gambar 3.2.



Gambar 3.2 Diagram Arsitektur Sistem

### 3.3.4.2 Perancangan Fungsional Sistem

#### A. Use Case Diagram

*Use case* diagram digunakan untuk menggambarkan hubungan antara pengguna sistem (aktor) dan fitur-fitur yang dapat mereka akses. Diagram ini membantu dalam memahami bagaimana sistem merespons berbagai interaksi berdasarkan peran pengguna yang berbeda. Dalam sistem digitalisasi dan manajemen koleksi

Museum Lampung, terdapat empat jenis aktor utama, yaitu: superadmin, admin koleksi, admin kegiatan, dan pengunjung.

1. Pengunjung merupakan pengguna umum yang tidak memiliki hak akses untuk mengubah data. Fitur-fitur yang tersedia untuk pengunjung antara lain:
  - a. Melihat daftar koleksi museum,
  - b. Melihat detail informasi koleksi,
  - c. Melakukan pencarian koleksi berdasarkan nama,
  - d. Melakukan penyaringan (*filter*) koleksi berdasarkan nama, kategori, atau tahun,
  - e. Melihat daftar kegiatan (*event*) yang tersedia,
  - f. Melihat detail informasi kegiatan.
2. Admin Koleksi bertanggung jawab dalam pengelolaan data koleksi. Setelah berhasil login, admin koleksi dapat melakukan hal-hal berikut:
  - a. Menambahkan koleksi baru ke dalam sistem,
  - b. Mengubah informasi koleksi yang sudah ada,
  - c. Menghapus koleksi yang tidak relevan,
  - d. Melihat statistik koleksi pada *dashboard*.
3. Admin Kegiatan memiliki tanggung jawab dalam pengelolaan kegiatan atau event museum. Setelah *login*, admin kegiatan memiliki hak untuk:
  - a. Menambahkan kegiatan baru ke sistem,
  - b. Mengubah informasi kegiatan yang telah terdaftar,
  - c. Menghapus kegiatan yang sudah tidak relevan,
  - d. Melihat daftar kegiatan yang tersedia.
4. Superadmin memiliki peran tertinggi dalam sistem. Selain dapat mengakses *dashboard* dan *login*, superadmin memiliki hak istimewa untuk:

- a. Melihat daftar akun admin (baik admin koleksi maupun admin kegiatan),
- b. Menambahkan akun admin baru,
- c. Menghapus akun admin yang tidak aktif atau tidak diperlukan,
- d. Mengubah informasi akun admin,
- e. Melihat data koleksi dan kegiatan tanpa hak untuk mengubah.

Dengan pembagian peran ini, sistem menerapkan *Role-Based Access Control (RBAC)* untuk memastikan keamanan dan efisiensi dalam pengelolaan data. Setiap aktor hanya dapat mengakses fitur yang sesuai dengan hak aksesnya. Interaksi antara seluruh aktor dengan fitur sistem dapat dilihat secara visual pada Gambar 3.3, yang menggambarkan hubungan antara setiap peran dengan fitur-fitur utama yang tersedia di sistem digitalisasi Museum Lampung.



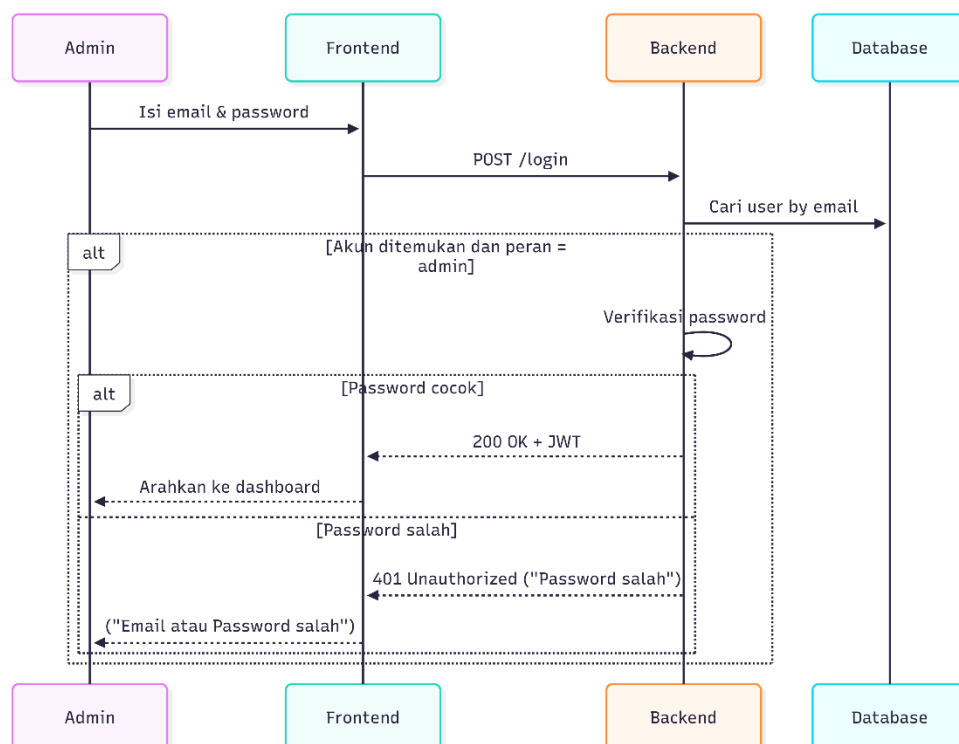
Gambar 3.3 Use Case Diagram

## B. Sequence Diagram

*Sequence* diagram merupakan representasi visual dari alur komunikasi antar komponen dalam sistem berdasarkan urutan waktu. Diagram ini menggambarkan bagaimana aktor (pengguna) seperti admin dan superadmin berinteraksi dengan sistem dalam menjalankan berbagai fungsi penting, seperti *login*, pengelolaan akun admin, dan manajemen koleksi. Dengan menggunakan *sequence* diagram, pengembang dan pihak terkait dapat memahami bagaimana data berpindah dari *frontend* ke *backend*, serta bagaimana sistem merespons setiap permintaan, baik dalam kondisi berhasil maupun gagal.

## 1. Sequence Diagram Login Admin

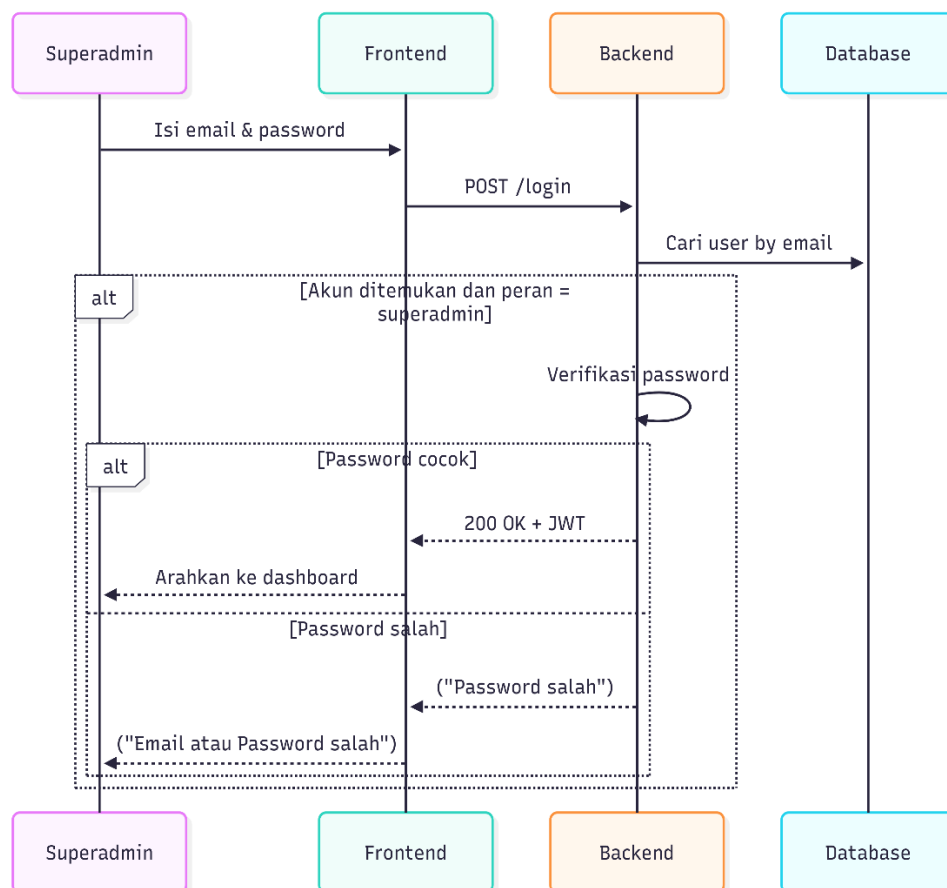
Proses *login* untuk admin diawali ketika pengguna memasukkan alamat email dan kata sandi pada halaman *login* admin. Setelah tombol “Login” ditekan, data dikirimkan oleh *frontend* ke *backend* menggunakan metode *POST*. *Backend* kemudian memeriksa apakah akun dengan email tersebut ada di *database*. Jika ditemukan, *backend* melakukan verifikasi password yang diinput dengan *password* yang tersimpan dalam bentuk *hash*. Bila *password* cocok dan peran pengguna adalah “admin”, *backend* akan menghasilkan token JWT (*JSON Web Token*) sebagai bukti autentikasi, lalu mengirimkannya kembali ke *frontend*. *Frontend* menyimpan token ini, kemudian mengarahkan pengguna ke halaman *dashboard* admin. Namun, apabila terjadi kesalahan seperti email atau password tidak sesuai, maka *backend* akan mengembalikan respons error, “Email atau Password salah”. Pesan ini akan ditampilkan oleh *frontend*, sehingga pengguna mengetahui bahwa proses *login* gagal. Sequence diagram *login* Admin dalam website Museum Lampung dapat dilihat pada Gambar 3.4.



Gambar 3.4 Sequence Diagram Login Admin

## 2. Sequence Diagram Login Superadmin

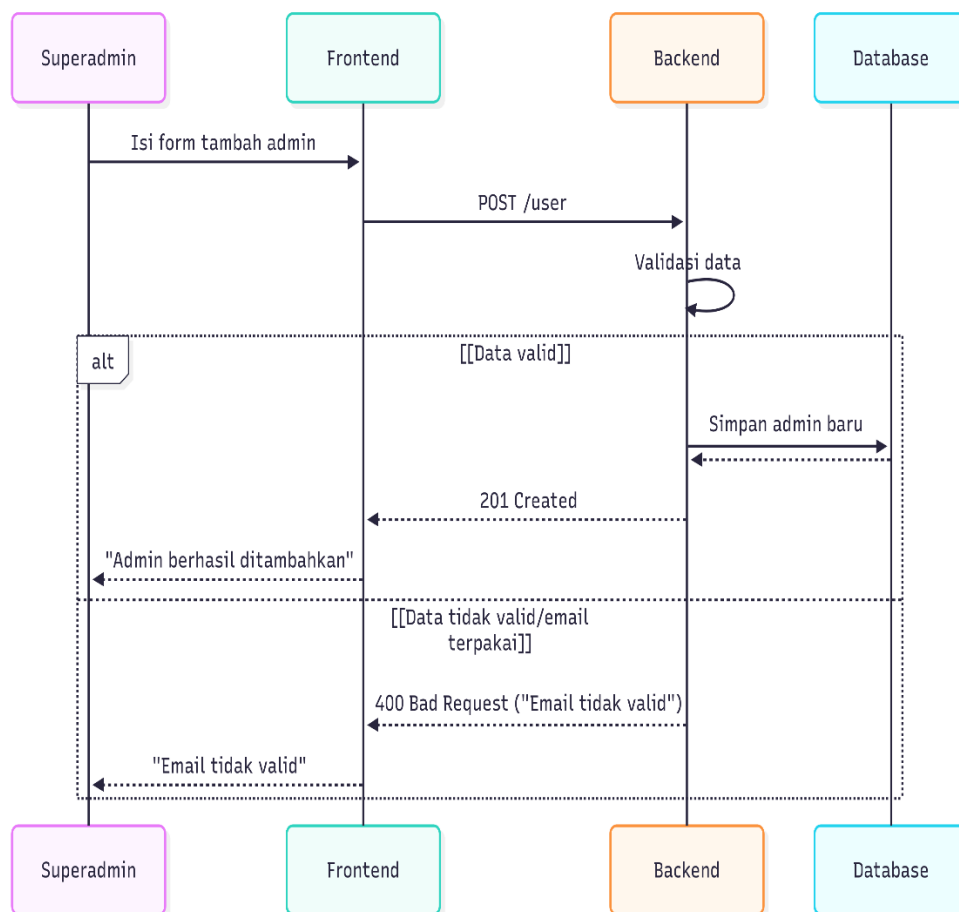
Skenario *login* superadmin serupa dengan admin, namun dengan hak akses yang lebih tinggi. Pengguna dengan peran superadmin mengisi *form login* dengan email dan *password*. Data tersebut dikirim ke *backend*, dan *backend* melakukan validasi terhadap email dan *password* yang diberikan. Jika akun ditemukan dan memiliki peran “superadmin”, *backend* akan membuat token JWT dan mengirimkannya ke *frontend* sebagai tanda autentikasi berhasil. *Frontend* kemudian menyimpan token tersebut dan mengarahkan superadmin ke *dashboard* khusus untuk mengelola akun admin. Jika validasi gagal, misalnya karena email tidak ditemukan, atau *password* salah, *backend* mengembalikan pesan error “Email atau Password salah”. *Frontend* akan menampilkan notifikasi kesalahan yang sesuai agar pengguna dapat memperbaiki *input*. Sequence diagram *login* Superadmint dilihat pada Gambar 3.5.



Gambar 3.5 Sequence Diagram Login Superadmin

### 3. Sequence Diagram Tambah Admin (Superadmin)

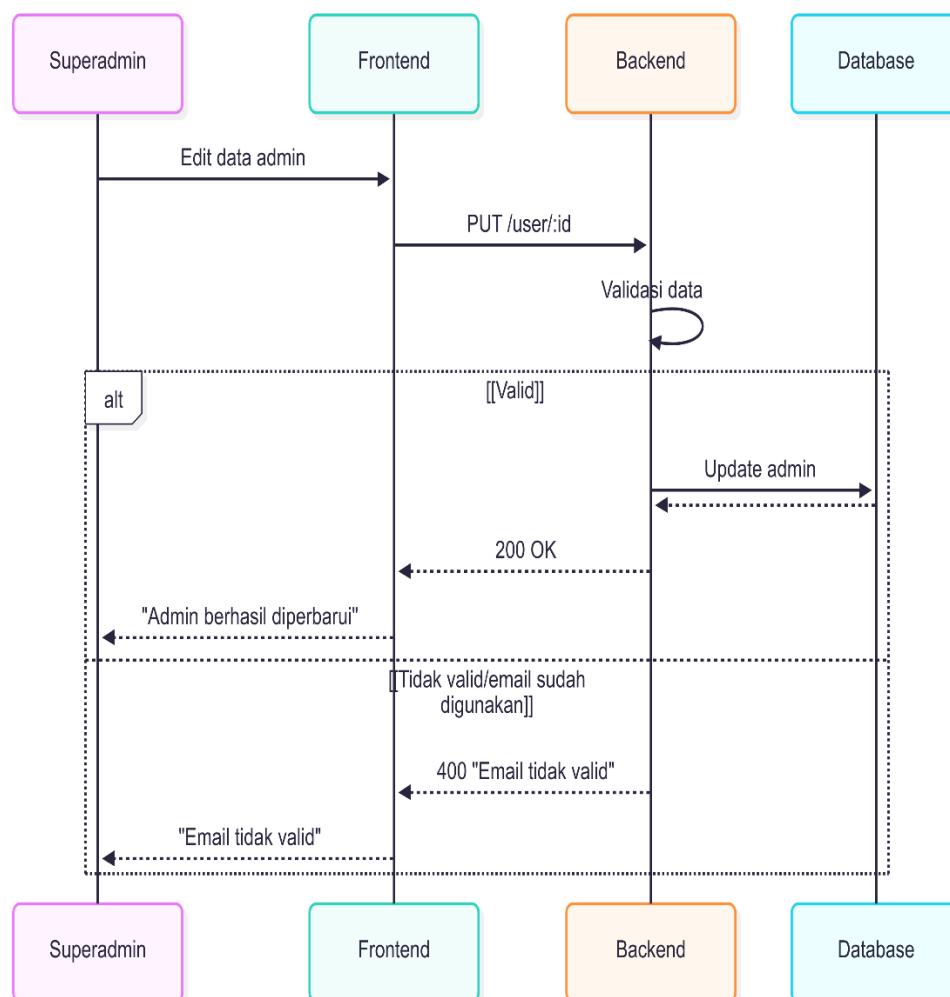
Superadmin memiliki akses untuk menambahkan akun admin baru. Proses ini dimulai saat superadmin mengisi formulir tambah admin yang mencakup nama, email, *password*. Setelah data dikirimkan ke *backend* melalui metode *POST*, sistem akan melakukan validasi terhadap setiap *field*, termasuk format email dan panjang *password*. *Backend* juga memeriksa apakah email sudah digunakan. Jika semua validasi berhasil, data akan disimpan ke dalam basis data. Setelah berhasil ditambahkan, *backend* mengirimkan respons sukses dan *frontend* akan menampilkan notifikasi “Admin berhasil ditambahkan”. Namun, jika terjadi kesalahan seperti email tidak valid atau sudah digunakan, *backend* akan mengembalikan pesan error seperti “Email tidak valid” atau “Email sudah terdaftar”. Pesan ini ditampilkan ke superadmin melalui notifikasi yang jelas di halaman frontend. *Sequence* diagram tambah admin dapat dilihat pada gambar 3.6.



Gambar 3.6 Sequence Diagram Tambah Admin

#### 4. Sequence Diagram Edit Admin (Superadmin)

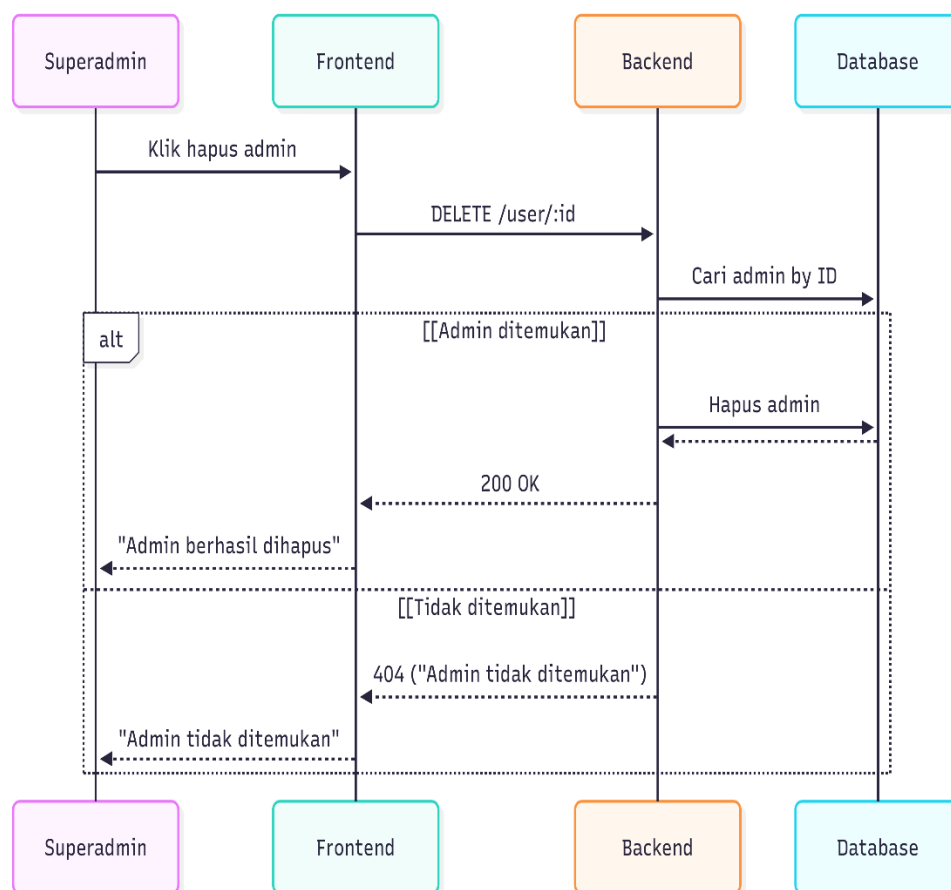
Superadmin juga dapat memperbarui informasi akun admin yang sudah ada. Langkah ini dimulai dengan memilih salah satu akun dari daftar, kemudian membuka formulir edit yang sudah terisi data lama. Setelah perubahan dilakukan, data dikirim ke *backend* melalui metode *PUT*. *Backend* akan memvalidasi ulang data yang dikirim, termasuk memastikan email yang baru tidak digunakan oleh akun lain. Jika validasi berhasil, data diperbarui di *database*, dan *backend* mengirimkan respons sukses. *Frontend* kemudian menampilkan notifikasi bahwa perubahan berhasil. Namun jika validasi gagal, misalnya karena format email tidak valid atau email sudah digunakan oleh akun lain, maka *backend* mengembalikan pesan error. Pesan seperti “Email tidak valid” akan ditampilkan di antarmuka *frontend*. Sequence diagram edit admin dapat dilihat pada Gambar 3.7.



Gambar 3.7 Sequence Diagram Edit Admin

### 5. Sequence Diagram Hapus Admin (Superadmin)

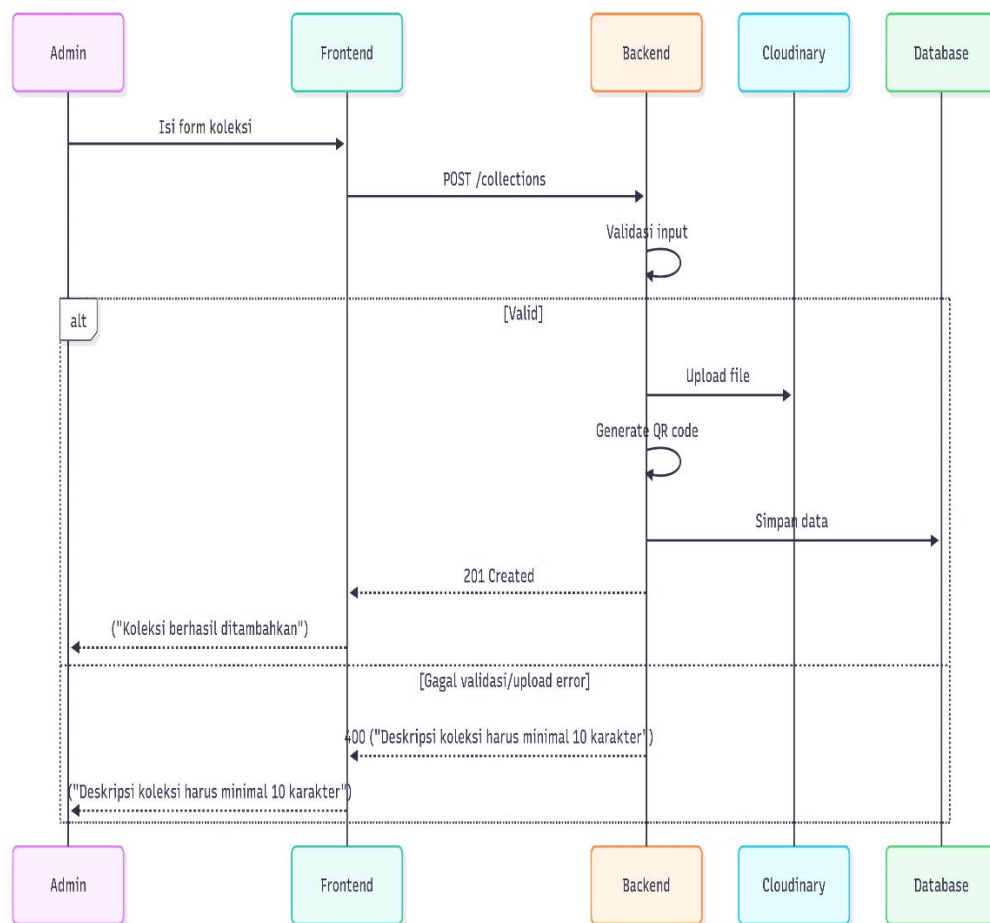
Superadmin dapat menghapus akun admin dari sistem. Proses ini dimulai saat superadmin menekan tombol hapus pada salah satu entri admin di daftar. Setelah muncul dialog konfirmasi dan superadmin menyetujui, *frontend* mengirimkan permintaan *DELETE* ke *backend*. *Backend* memverifikasi apakah *ID* admin tersebut ada dalam *database*. Jika ditemukan, data admin akan dihapus dan *backend* mengirimkan respons sukses. *Frontend* kemudian memperbarui tampilan daftar dan menampilkan notifikasi bahwa admin berhasil dihapus. Namun, jika admin tidak ditemukan atau *ID* tidak valid, *backend* akan mengembalikan respons *error* seperti “Admin tidak ditemukan”. *Frontend* menampilkan pesan tersebut sebagai peringatan agar superadmin mengetahui bahwa proses penghapusan gagal. Sequence diagram hapus admin dapat dilihat pada Gambar 3.8.



Gambar 3.8 Sequence Diagram Hapus Admin

## 6. Sequence Diagram Tambah Koleksi (Admin Koleksi)

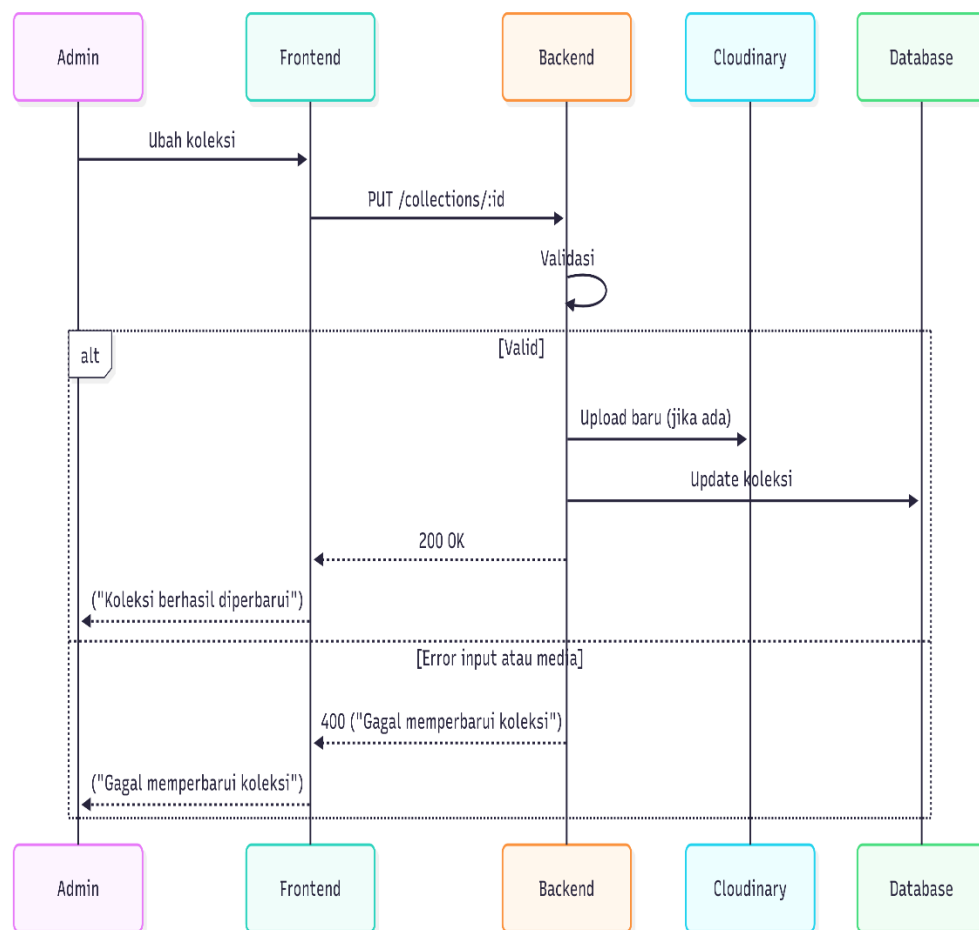
Admin memiliki akses untuk menambahkan koleksi ke sistem. Proses dimulai saat admin mengisi formulir yang memuat informasi seperti nama koleksi, deskripsi, tahun, kategori, dan *file* media (gambar atau video 360°). Setelah formulir dikirim, *frontend* meneruskan data ke *backend*. *Backend* akan memvalidasi *input*, mengunggah *file* media ke Cloudinary, menghasilkan *QR Code* yang merujuk ke halaman detail koleksi, lalu menyimpan seluruh data ke dalam *database PostgreSQL*. Jika seluruh proses berjalan lancar, *backend* mengembalikan status berhasil dan *frontend* menampilkan notifikasi bahwa koleksi berhasil ditambahkan. Jika terdapat kesalahan seperti *input* tidak valid atau *file* gagal diunggah, *backend* akan mengirimkan pesan error seperti “Deskripsi koleksi harus minimal 10 karakter”, dan *frontend* menampilkan pesan tersebut agar admin dapat memperbaiki input. Sequence diagram tambah koleksi dapat dilihat pada Gambar 3.9.



Gambar 3.9 Sequence Diagram Tambah Koleksi

### 7. Sequence Diagram Edit Koleksi (Admin Koleksi)

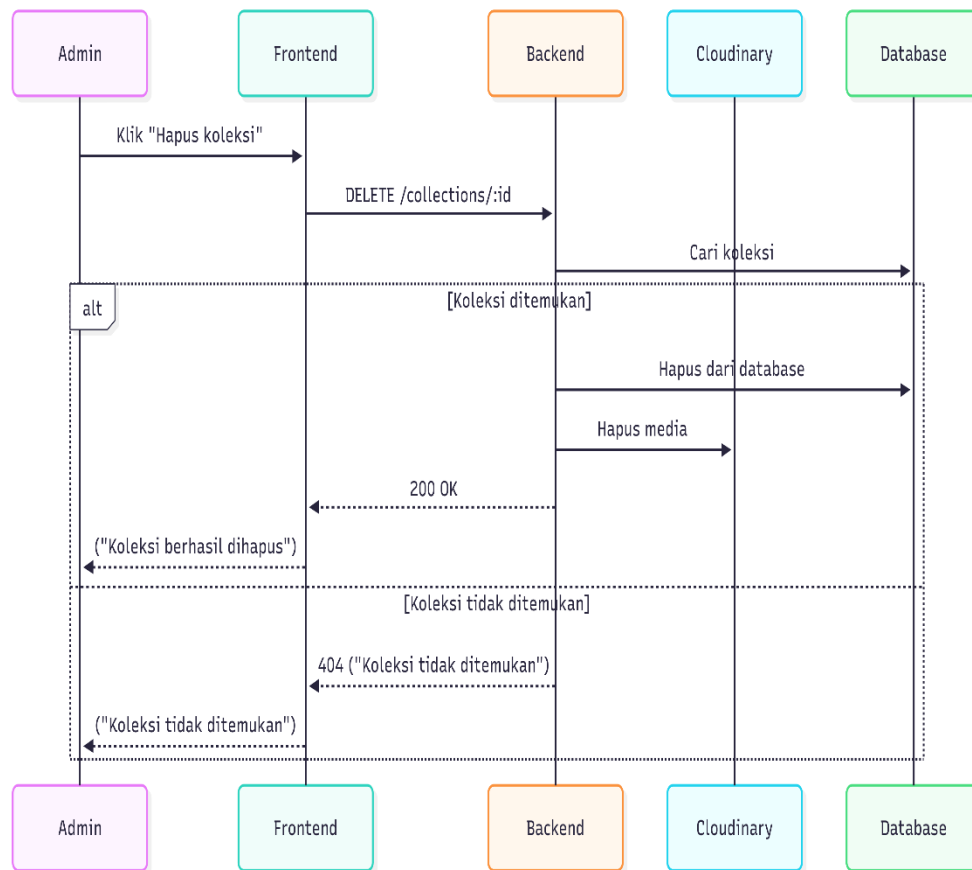
Admin dapat memperbarui data koleksi yang sudah ada. Proses ini diawali dengan memilih koleksi yang ingin diperbarui, lalu membuka halaman edit yang sudah menampilkan data lama. Setelah admin melakukan perubahan dan mengunggah *file* baru (jika diperlukan), data dikirim ke *backend* melalui metode *PUT*. *Backend* memvalidasi data baru, mengganti *file* lama jika ada media baru, dan memperbarui entri koleksi di *database*. Jika semua proses berhasil, *backend* mengirimkan respons sukses dan *frontend* menampilkan pesan “Koleksi berhasil diperbarui”. Namun, jika validasi gagal atau proses unggah media tidak berhasil, *backend* akan mengembalikan pesan error seperti “Gagal memperbarui koleksi”. *Frontend* akan menampilkan pesan tersebut agar admin dapat mengetahui letak kesalahan. *Sequence* diagram edit koleksi dapat dilihat pada Gambar 3.10.



Gambar 3.10 Sequence Diagram Edit Koleksi

### 8. Sequence Diagram Hapus Koleksi (Admin Koleksi)

Admin dapat menghapus koleksi dari sistem untuk menjaga kebersihan dan relevansi data. Saat admin menekan tombol hapus, *frontend* akan menampilkan konfirmasi. Jika admin menyetujui, permintaan *DELETE* dikirim ke *backend*. *Backend* mencari koleksi berdasarkan *ID* yang diberikan. Jika ditemukan, data koleksi dihapus dari *database* dan *file* media yang terkait juga dihapus dari Cloudinary. Setelah proses selesai, *backend* mengirimkan respons sukses dan *frontend* memperbarui tampilan serta menampilkan pesan “Koleksi berhasil dihapus”. Jika koleksi tidak ditemukan, *backend* mengembalikan pesan “Koleksi tidak ditemukan”, dan *frontend* akan menampilkan notifikasi tersebut kepada admin. *Sequence* diagram hapus koleksi dapat dilihat pada Gambar 3.11.

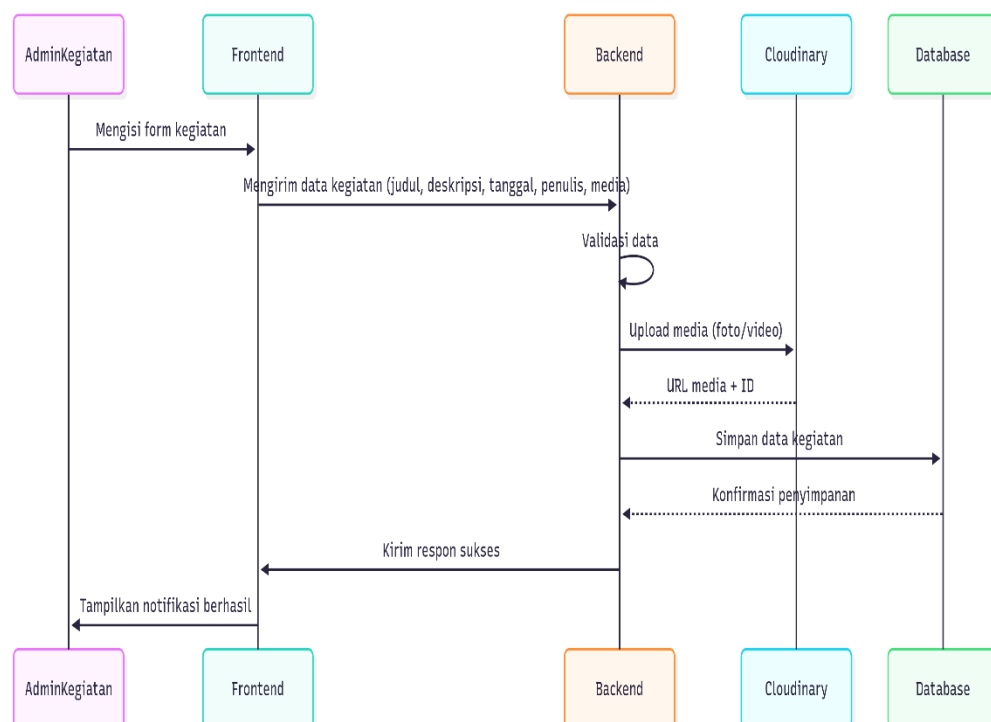


Gambar 3.11 Sequence Diagram Hapus Koleksi

## 9. Sequence Diagram Tambah Kegiatan (Admin Kegiatan)

*Sequence* diagram untuk proses penambahan kegiatan menggambarkan interaksi antara admin kegiatan dengan sistem ketika ingin menambahkan data kegiatan baru. Proses diawali ketika admin mengisi formulir yang memuat informasi seperti judul kegiatan, deskripsi, tanggal pelaksanaan, nama penulis (author), serta media pendukung berupa foto atau video. Setelah data lengkap dan valid, permintaan dikirim ke sistem *backend* untuk diproses.

Sistem kemudian melakukan validasi terhadap seluruh isian formulir, termasuk memverifikasi keberadaan media sebagai bahan pendukung. Apabila data yang dikirimkan telah memenuhi seluruh persyaratan validasi, maka *file* media akan diunggah ke layanan penyimpanan eksternal (Cloudinary), dan seluruh data kegiatan akan disimpan dalam basis data. Setelah proses penyimpanan selesai, sistem mengirimkan respon keberhasilan kepada pengguna, yang kemudian ditampilkan dalam bentuk notifikasi di antarmuka *frontend*.

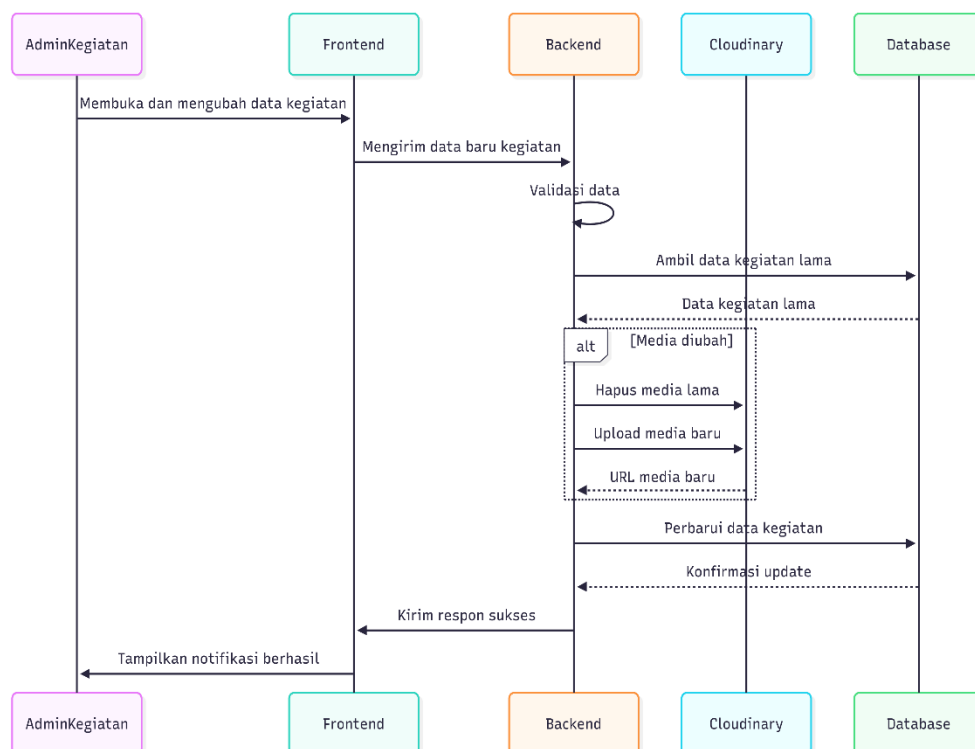


Gambar 3.12 *Sequence* Diagram Tambah Kegiatan

## 10. Sequence Diagram Edit Kegiatan (Admin Kegiatan)

Pada proses pengeditan kegiatan, *sequence* diagram menunjukkan alur komunikasi saat admin kegiatan memperbarui data kegiatan yang telah terdaftar dalam sistem. Proses ini dimulai dengan pemilihan salah satu kegiatan yang telah ada. Sistem akan menampilkan data kegiatan lama, kemudian admin melakukan perubahan sesuai kebutuhan, baik berupa judul, deskripsi, tanggal kegiatan, nama penulis, maupun *file* media.

Setelah perubahan *disubmit*, sistem akan melakukan validasi ulang terhadap data baru yang dikirimkan. Jika terdapat *file* media yang diganti, maka sistem akan terlebih dahulu menghapus *file* lama dari penyimpanan sebelum menggantinya dengan *file* baru. Setelah seluruh data berhasil diperbarui dalam basis data, sistem akan memberikan respon positif yang ditampilkan dalam antarmuka sebagai notifikasi keberhasilan.

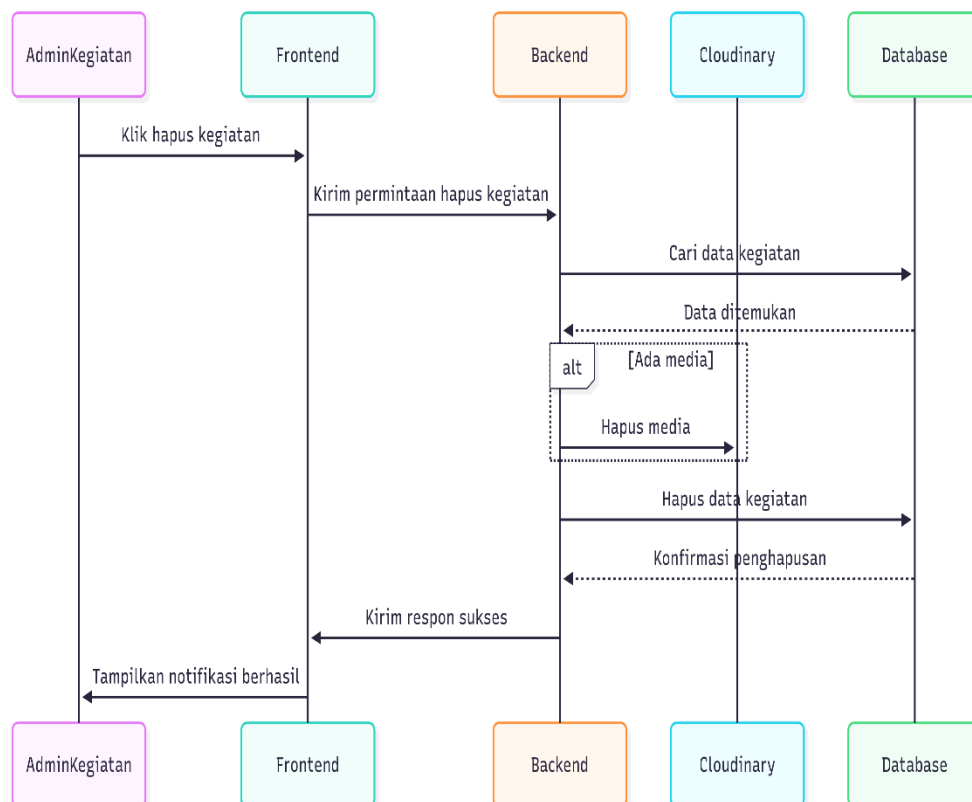


Gambar 3.13 Sequence Diagram Edit Kegiatan

## 11. Sequence Diagram Hapus Kegiatan (Admin Kegiatan)

*Sequence* diagram untuk proses penghapusan kegiatan menggambarkan langkah-langkah yang dilakukan saat admin kegiatan menghapus entri kegiatan dari sistem. Proses ini dimulai saat admin menekan tombol hapus pada daftar kegiatan, dilanjutkan dengan konfirmasi dari sistem. Setelah disetujui, permintaan untuk menghapus dikirim ke backend.

Sistem kemudian memverifikasi keberadaan ID kegiatan dalam database. Jika kegiatan ditemukan, maka data kegiatan akan dihapus dari basis data, termasuk menghapus *file* media yang terkait dari layanan penyimpanan eksternal. Setelah proses penghapusan berhasil, sistem akan mengirimkan notifikasi bahwa kegiatan telah berhasil dihapus. Namun, apabila ID kegiatan tidak ditemukan, sistem akan mengembalikan pesan kesalahan kepada pengguna.



Gambar 3.14 *Sequence* Diagram Hapus Kegiatan

### 3.3.4.3 Perancangan Tabel Database

Perancangan basis data dalam sistem digitalisasi koleksi Museum Lampung menggunakan *PostgreSQL* dengan pendekatan relasional dan menerapkan prinsip normalisasi. Perancangan dilakukan berdasarkan model data yang didefinisikan melalui *Sequelize ORM* di sisi *backend*, dengan tujuan untuk menjamin integritas data, meminimalkan redundansi, serta mendukung skalabilitas sistem. Setiap tabel dirancang untuk mencerminkan kebutuhan fungsional dari fitur utama yang tersedia, seperti manajemen pengguna, koleksi, *QR Code*, dan pencatatan kunjungan.

#### a. Tabel Users

Tabel *users* menyimpan data akun pengguna sistem, baik yang berperan sebagai admin maupun superadmin. Tabel ini mencakup informasi *login* pengguna seperti username, email, dan password yang sudah di-hash untuk keamanan. Field *role* menentukan hak akses pengguna, dan tipe datanya berupa ENUM untuk membatasi nilai hanya pada "admin" dan "superadmin".

Tabel 3.3 Struktur Tabel *Users*

| Kolom     | Tipe Data             | Keterangan                        |
|-----------|-----------------------|-----------------------------------|
| id        | Integer (Primary Key) | ID unik pengguna                  |
| username  | Varchar (Unique)      | Nama pengguna                     |
| email     | Varchar (Unique)      | Alamat email pengguna             |
| password  | Varchar               | <i>Password</i> terenkripsi       |
| role      | Enum                  | Peran pengguna (admin/superadmin) |
| createdAt | Timestamp             | Tanggal akun dibuat               |
| updatedAt | Timestamp             | Tanggal akun diperbarui terakhir  |

#### b. Tabel Collections

Tabel *collections* menyimpan informasi utama mengenai koleksi digital museum. Setiap koleksi memiliki nama, deskripsi, tahun, dan kategori. Selain itu, koleksi dapat memiliki gambar atau video 360° yang disimpan melalui layanan Cloudinary. Sistem juga menyimpan *ID file* media (*publicId*) untuk keperluan manajemen *file* (hapus/edit).

Tabel 3.4 Struktur Tabel *Collections*

| Kolom            | Tipe Data             | Keterangan                          |
|------------------|-----------------------|-------------------------------------|
| id               | Integer (Primary Key) | ID koleksi                          |
| name             | Varchar               | Nama koleksi                        |
| description      | Text                  | Deskripsi koleksi                   |
| year             | Integer               | Tahun koleksi                       |
| category         | Enum                  | Kategori koleksi (9 jenis kategori) |
| image            | Varchar               | URL gambar koleksi (Cloudinary)     |
| video360         | Varchar               | URL video 360° koleksi (Cloudinary) |
| imagePublicId    | Varchar               | Public ID gambar di Cloudinary      |
| video360PublicId | Varchar               | Public ID video 360° di Cloudinary  |
| createdAt        | Timestamp             | Tanggal koleksi ditambahkan         |
| updatedAt        | Timestamp             | Tanggal koleksi diperbarui terakhir |

## c. Tabel QRCode

Tabel qrCodes menyimpan QR Code yang dihasilkan secara otomatis untuk setiap koleksi. QR Code ini digunakan sebagai representasi fisik yang dapat dipindai oleh pengunjung untuk membuka halaman detail koleksi secara langsung. Relasi antartabel adalah satu-ke-satu, di mana satu QR Code hanya dimiliki oleh satu koleksi.

Tabel 3.5 Struktur Tabel QRCode

| Kolom        | Tipe Data             | Keterangan                          |
|--------------|-----------------------|-------------------------------------|
| id           | Integer (Primary Key) | ID QR Code                          |
| collectionId | Integer (Foreign Key) | ID koleksi yang memiliki QR Code    |
| qrCodeUrl    | Varchar               | URL <i>file</i> QR Code koleksi     |
| createdAt    | Timestamp             | Tanggal QR Code dibuat              |
| updatedAt    | Timestamp             | Tanggal QR Code diperbarui terakhir |

## d. Tabel Visitors

Tabel visitors mencatat kunjungan pengguna berbasis alamat IP. Fitur ini digunakan untuk mencatat statistik pengunjung unik ke dalam sistem dalam kurun waktu tertentu. Sistem mencegah pencatatan *IP* yang sama secara berulang dalam waktu berdekatan untuk menjaga akurasi data statistik.

Tabel 3.6 Struktur Tabel Visitors

| Kolom     | Tipe Data             | Keterangan              |
|-----------|-----------------------|-------------------------|
| id        | Integer (Primary Key) | ID kunjungan            |
| ipAddress | Varchar               | Alamat IP pengunjung    |
| timestamp | Timestamp             | Waktu kunjungan dicatat |

## e. Tabel Events

Tabel events menyimpan data kegiatan yang diunggah oleh admin kegiatan. Setiap kegiatan memiliki informasi berupa judul, deskripsi, tanggal pelaksanaan (activityDate), serta nama penulis (author) yang merepresentasikan siapa yang membuat atau bertanggung jawab atas kegiatan tersebut. Admin juga dapat melampirkan media berupa foto atau video untuk mendukung dokumentasi kegiatan. Media yang diunggah disimpan di Cloudinary, dan sistem menyimpan URL serta public ID file untuk keperluan manajemen file. Tabel ini memiliki relasi dengan tabel users melalui kolom creatorId, yang menunjukkan admin yang membuat entri kegiatan tersebut.

Tabel 3.7 Struktur Tabel Events

| Kolom         | Tipe Data             | Keterangan                                        |
|---------------|-----------------------|---------------------------------------------------|
| id            | Integer (Primary Key) | ID unik kegiatan                                  |
| title         | Varchar               | Judul kegiatan                                    |
| description   | Text                  | Deskripsi kegiatan secara rinci                   |
| imageUrl      | Varchar (Nullable)    | URL foto kegiatan (Cloudinary)                    |
| videoUrl      | Varchar (Nullable)    | URL video kegiatan (Cloudinary)                   |
| imagePublicId | Varchar (Nullable)    | ID file foto di Cloudinary                        |
| videoPublicId | Varchar (Nullable)    | ID file video di Cloudinary                       |
| creatorId     | Integer (Foreign Key) | ID admin pembuat kegiatan (relasi ke tabel Users) |
| author        | Varchar               | Nama penulis kegiatan                             |
| activityDate  | Date                  | Tanggal pelaksanaan kegiatan                      |
| createdAt     | Timestamp             | Tanggal kegiatan dicatat ke dalam sistem          |
| updatedAt     | Timestamp             | Tanggal kegiatan terakhir diperbarui              |

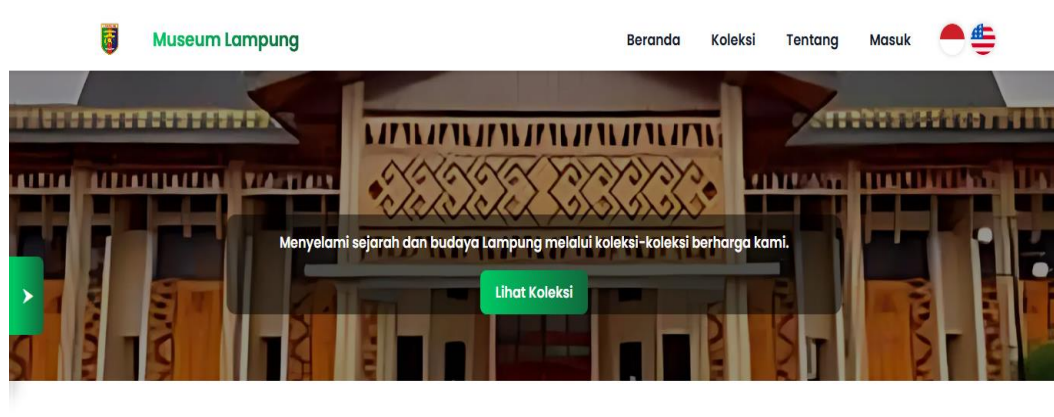
### 3.3.4.4 Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna pada sistem digitalisasi koleksi Museum Lampung menggunakan teknologi *React.js*, yang memungkinkan pengembangan

berbasis komponen modular dan fleksibel. Setiap halaman dirancang untuk memenuhi kebutuhan fungsional pengunjung umum, admin, dan superadmin, dengan memperhatikan prinsip desain *responsive*, *user-friendly*, dan *minimal distraction interface*. Halaman-halaman ini mendukung pengelolaan koleksi secara efisien, sekaligus memfasilitasi eksplorasi informasi koleksi oleh masyarakat luas.

#### a. Halaman *Landing Page (Home)*

Halaman beranda merupakan titik masuk utama yang memberikan pengantar aplikasi secara ringkas dan informatif. Di bagian atas halaman terdapat navbar yang menyediakan tautan ke halaman koleksi, *Login*, dan pilihan bahasa. Halaman ini juga menampilkan tombol untuk membuka *drawer* interaktif yang menampilkan jumlah total pengunjung (*visitor*) secara *real-time*. *Drawer* muncul secara animatif dari sisi kiri layar dan memanfaatkan data dari *backend*. Tampilan halaman ini bersifat bersih, terpusat, dan responsif di berbagai perangkat, serta menjadi penunjuk awal nilai edukatif dari sistem digitalisasi Museum Lampung. Tampilan halaman beranda dapat dilihat pada Gambar 3.15.



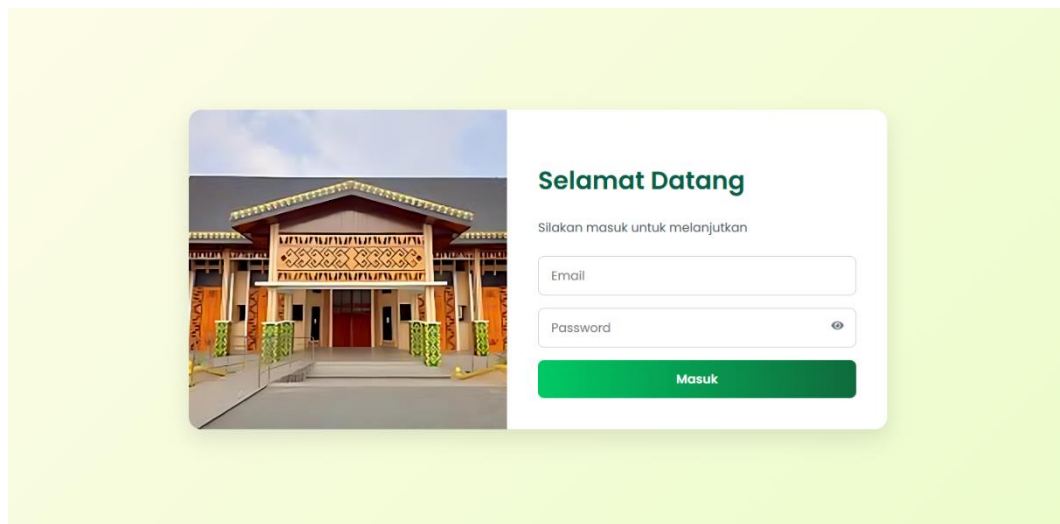
Gambar 3.15 Halaman *Landing Page (Home)*

#### b. Halaman *Login*

Halaman *login* berperan sebagai gerbang autentikasi untuk pengguna yang memiliki peran sebagai admin atau superadmin. Di halaman ini, pengguna diminta untuk memasukkan kredensial berupa email dan *password*. Setelah data dikirimkan, sistem akan melakukan verifikasi dan menyesuaikan tampilan

*dashboard* berdasarkan peran pengguna. Jika data sesuai dan berhasil diverifikasi, pengguna diarahkan ke halaman admin *dashboard* atau superadmin *dashboard*. Sebaliknya, jika terjadi kesalahan, sistem akan menampilkan pesan bahwa email atau *password* tidak valid.

Antarmuka halaman *login* dirancang minimalis dan terfokus, dengan *form* yang berada di tengah layar dan memiliki validasi *real-time* terhadap kesalahan input. Ini bertujuan untuk meningkatkan kenyamanan dan efisiensi saat pengguna melakukan proses autentikasi. Tampilan halaman login dapat dilihat pada Gambar 3.16.

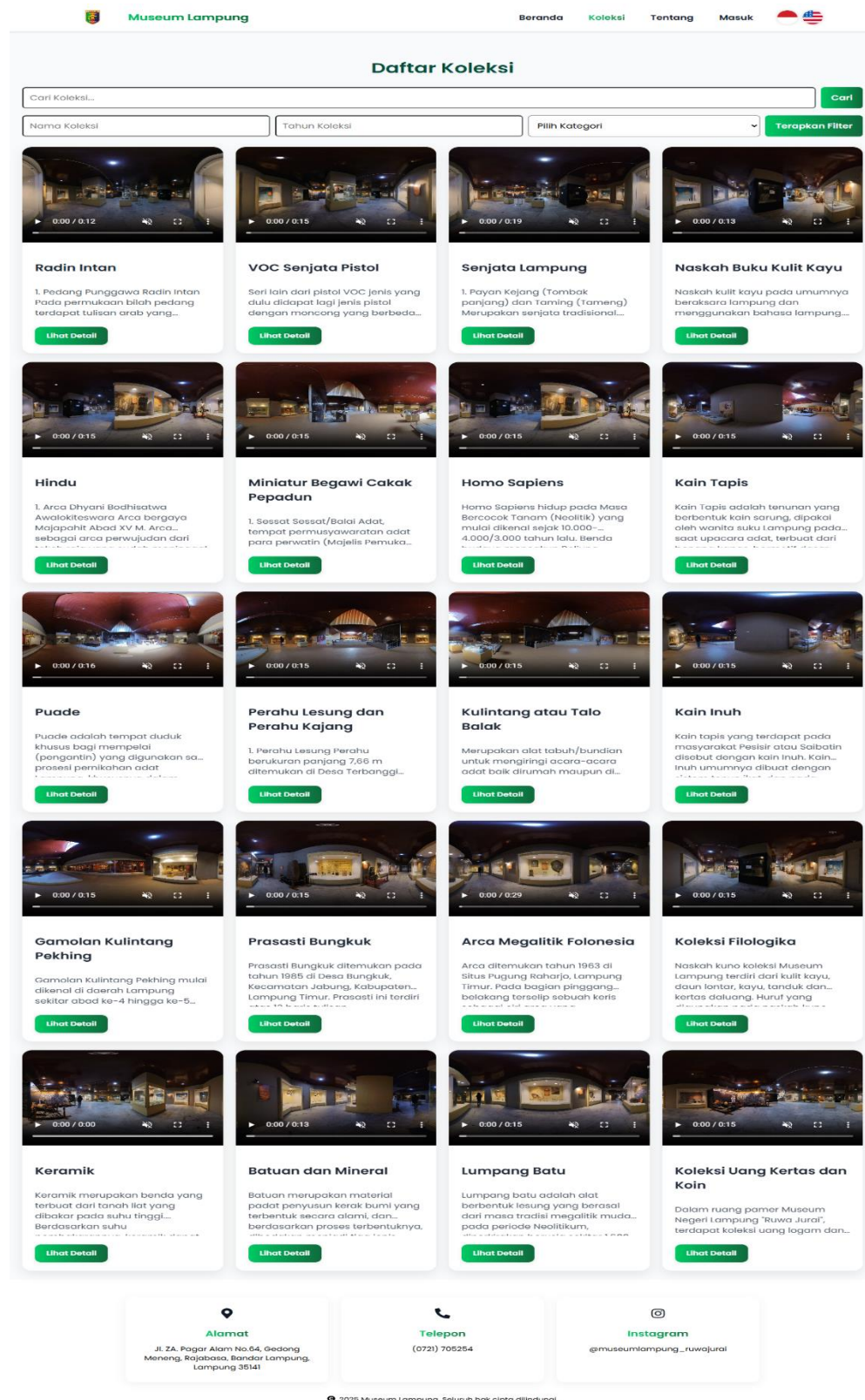


Gambar 3.16 Halaman *Login*

### c. Halaman Koleksi

Halaman koleksi berfungsi sebagai pusat eksplorasi data koleksi digital Museum Lampung yang dapat diakses oleh semua pengunjung. Koleksi ditampilkan dalam bentuk kartu (*card*) yang menampilkan gambar atau thumbnail video, nama koleksi, dan deskripsi singkat. Desain kartu bersifat responsif, mengikuti ukuran layar pengguna. Fitur tambahan seperti pencarian dinamis berdasarkan kata kunci dan *filter* berdasarkan tahun atau kategori koleksi tersedia di bagian atas halaman. Hal ini membantu pengguna untuk menelusuri koleksi dengan cepat dan tepat sasaran. Selain itu, setiap kartu dilengkapi dengan tombol “Lihat Detail” yang mengarahkan pengguna ke halaman informasi lengkap dari koleksi.

yang dipilih. Tampilan halaman koleksi dapat dilihat pada Gambar 3.17.

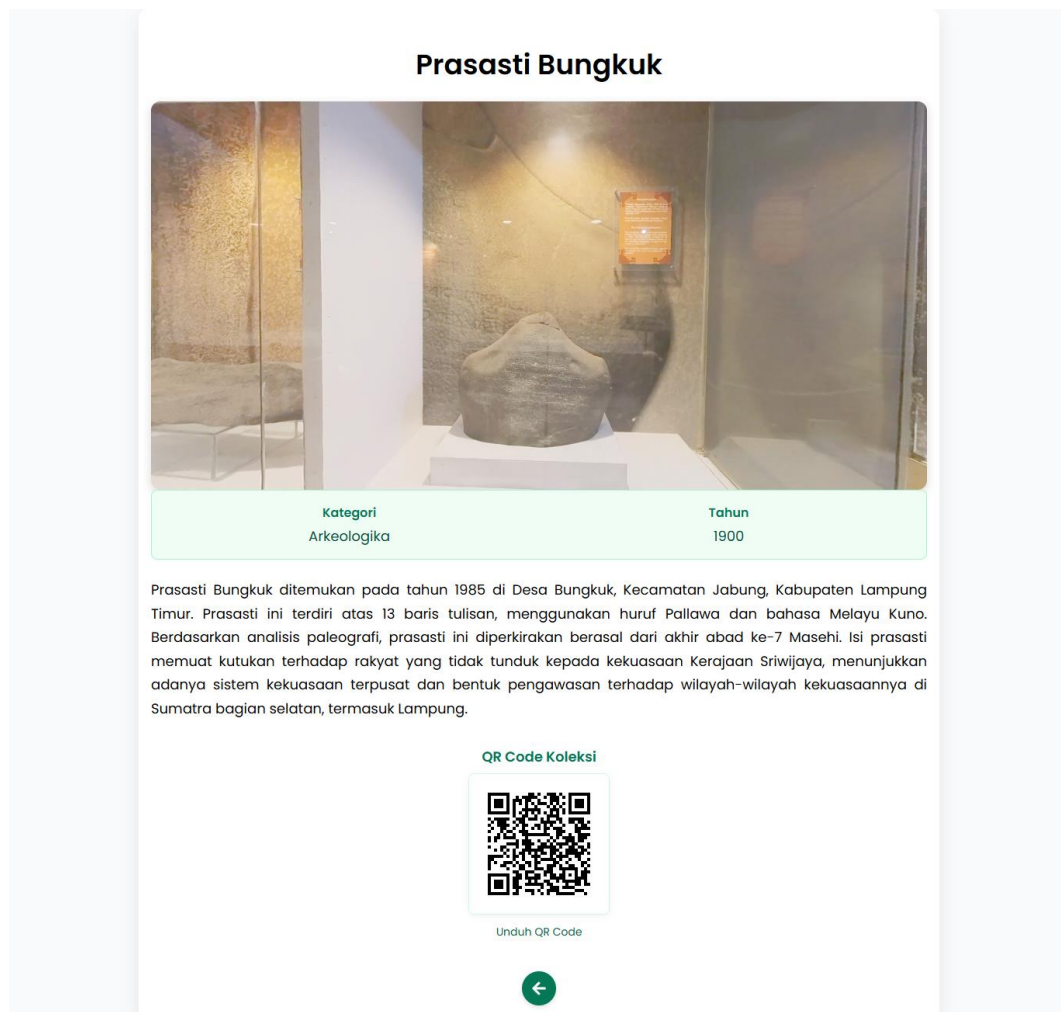


Gambar 3.14 Halaman Koleksi

#### d. Halaman Detail Koleksi

Halaman detail koleksi menyajikan informasi lengkap dan menyeluruh dari satu koleksi tertentu. Informasi yang ditampilkan mencakup nama koleksi, tahun perolehan, kategori, deskripsi panjang, gambar utama, serta video 360° jika tersedia. Gambar ditampilkan dalam ukuran besar dengan kemampuan untuk diperbesar (*zoom*), sedangkan video dapat diputar langsung di halaman.

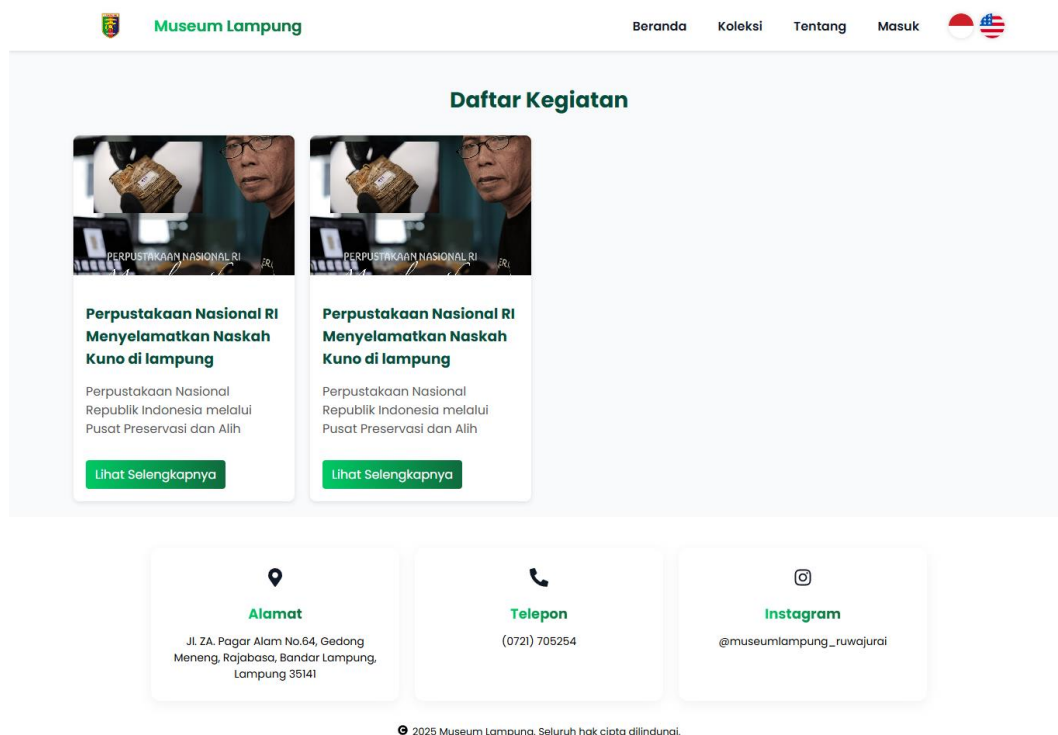
Halaman ini juga kompatibel dengan pemindaian *QR Code*. Jika pengguna memindai *QR Code* yang terdapat pada objek fisik di museum, mereka akan diarahkan langsung ke halaman ini. Fitur ini memperkuat integrasi antara pengalaman fisik dan digital. Tampilan halaman detail koleksi dapat dilihat pada Gambar 3.15.



Gambar 3.17 Halaman Detail Koleksi

### e. Halaman Daftar Kegiatan

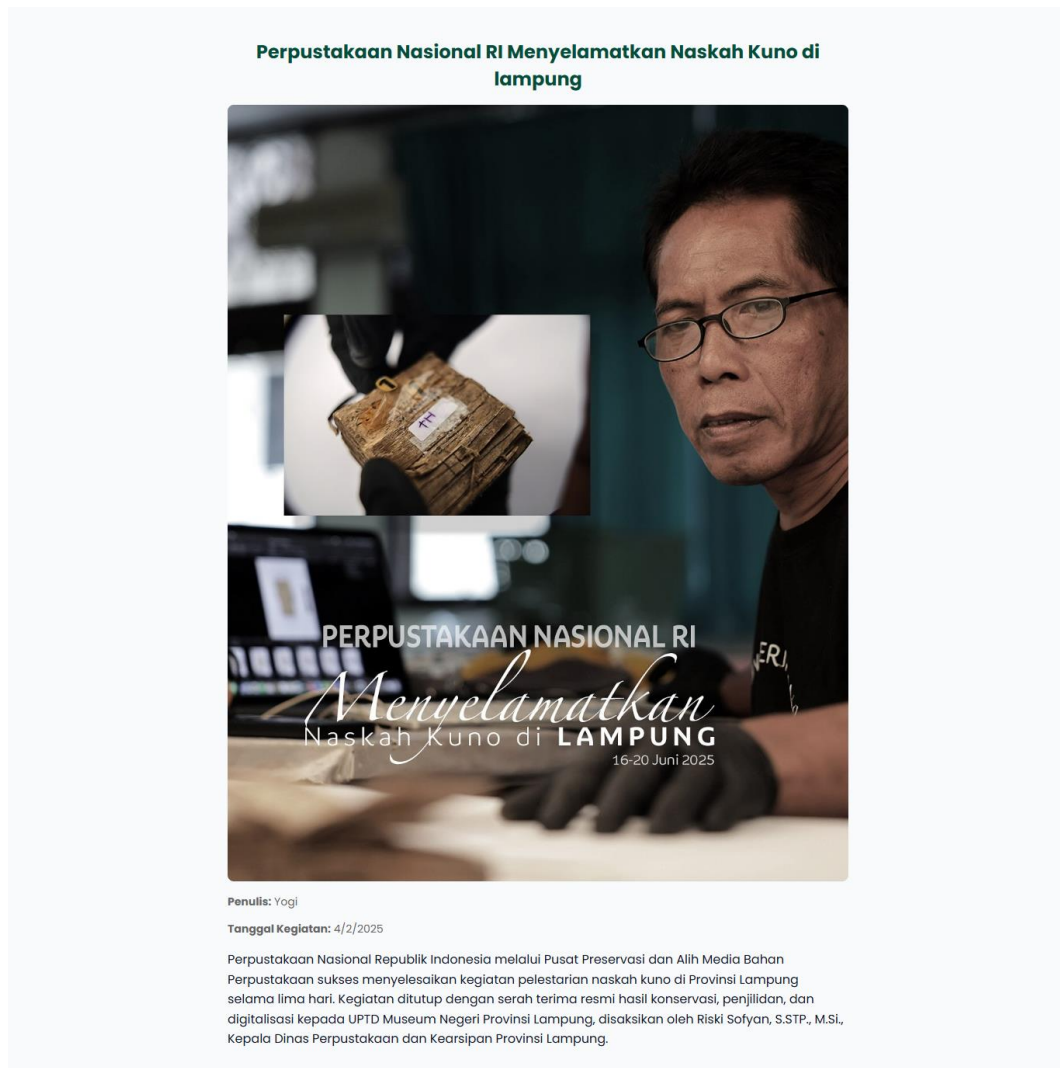
Halaman kegiatan berfungsi sebagai pusat informasi mengenai kegiatan atau acara yang diselenggarakan oleh Museum Lampung, yang dapat diakses oleh semua pengunjung. Kegiatan ditampilkan dalam bentuk kartu (*card*) yang menampilkan gambar atau video, nama kegiatan, penulis, tanggal, dan deskripsi singkat. Desain kartu bersifat responsif, mengikuti ukuran layar pengguna. Setiap kartu dilengkapi dengan tombol “Lihat Selengkapnya” yang mengarahkan pengguna ke halaman informasi lengkap dari kegiatan yang dipilih.



Gambar 3.18 Halaman Daftar Kegiatan

### f. Halaman Detail Kegiatan

Halaman detail kegiatan menyajikan informasi lengkap dan menyeluruh dari satu kegiatan tertentu. Informasi yang ditampilkan mencakup nama kegiatan, tanggal pelaksanaan, penulis, lokasi, deskripsi lengkap, dan gambar atau video kegiatan. Halaman ini juga mendukung pemindaian *QR Code* yang mengarahkan pengunjung langsung ke halaman detail kegiatan.



Gambar 3.19 Halaman Detail Kegiatan

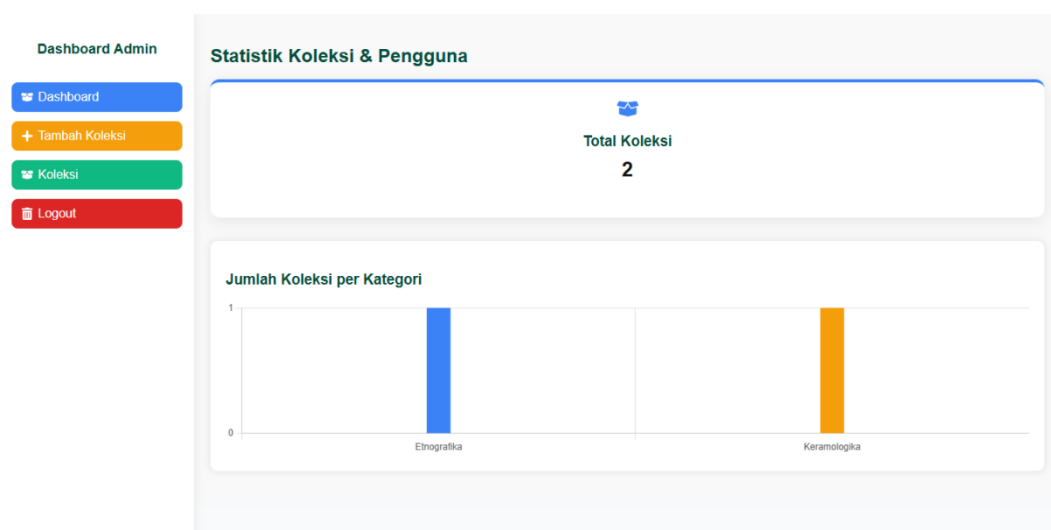
#### g. Halaman *Dashboard* Admin Koleksi

Admin koleksi memiliki akses eksklusif ke halaman *dashboard* yang menyajikan informasi ringkas dan fitur pengelolaan koleksi museum. Di tampilan awal *dashboard*, admin disuguhkan statistik jumlah total koleksi dan pengguna melalui tampilan kartu, serta grafik batang yang memperlihatkan distribusi koleksi berdasarkan kategori. Halaman ini dirancang agar ringkas dan mudah dipantau.

Admin dapat mengakses *form* tambah koleksi untuk mendaftarkan koleksi baru, yang mencakup informasi nama, deskripsi, tahun, kategori, gambar, dan video 360°. Setelah pengisian selesai, admin dapat menyimpan data dan sistem akan

mengunggah *file* media ke Cloudinary serta menghasilkan *QR Code*. Admin juga dapat mengakses daftar koleksi yang ditampilkan dalam bentuk tabel, lengkap dengan fitur pencarian, serta tombol aksi “edit” dan “hapus”.

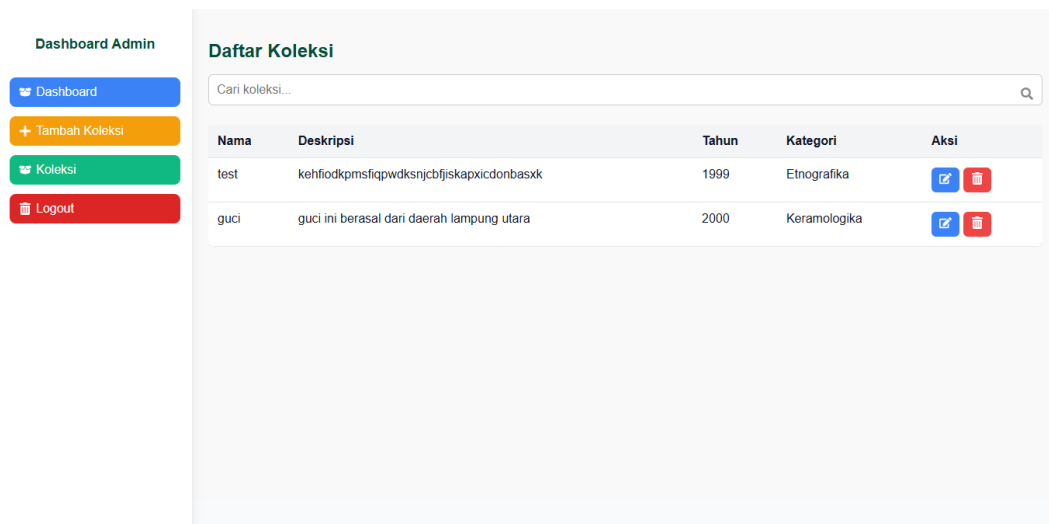
Selain itu, admin memiliki akses ke halaman edit koleksi, di mana data koleksi yang telah disimpan ditampilkan kembali dalam form isian untuk diperbarui. Admin juga dapat membuka halaman daftar pengguna yang menampilkan seluruh akun yang ada di sistem, sebagai bagian dari fungsi monitoring dan pengendalian akses. Tampilan halaman dashboard dan fungsinya dapat dilihat secara bertahap pada Gambar 3.20 hingga Gambar 3.23.



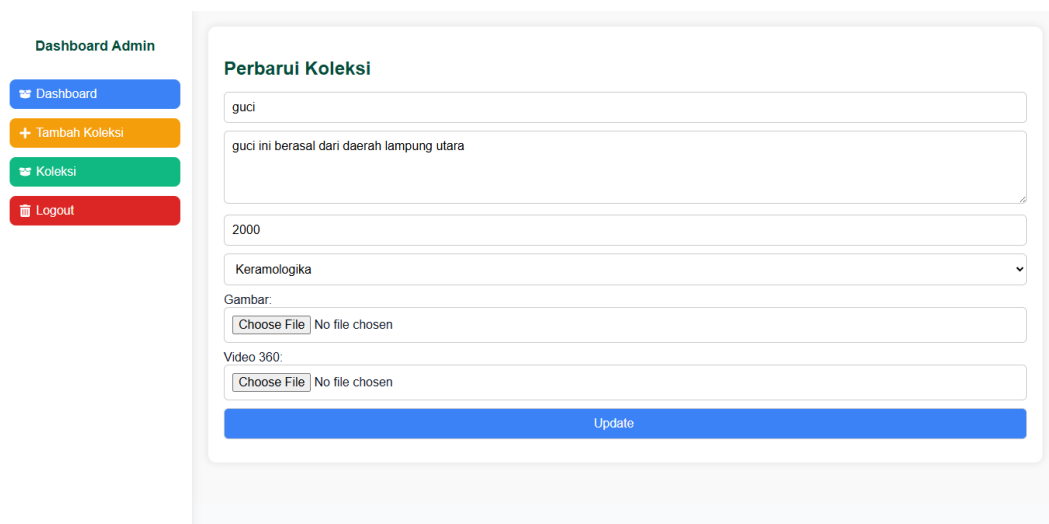
Gambar 3.20 Halaman Statistik *Dashboard* Admin

The screenshot shows the 'Tambah Koleksi' (Add Collection) form within the 'Dashboard Admin'. The sidebar is identical to the previous image. The form contains the following fields: 'Nama Koleksi' (text input), 'Deskripsi' (text area), 'Tahun' (text input), 'Pilih Kategori' (dropdown menu), 'Gambar:' (file upload section with a 'Choose File' button and 'No file chosen' text), and 'Video 360:' (file upload section with a 'Choose File' button and 'No file chosen' text). A blue 'Simpan' (Save) button is located at the bottom of the form.

Gambar 3.21 Halaman Tambah Koleksi



Gambar 3.22 Halaman Daftar Koleksi



Gambar 3.23 Halaman Edit Koleksi

#### h. Halaman *Dashboard* Admin Kegiatan

Admin Kegiatan memiliki peran khusus untuk mengelola seluruh informasi terkait kegiatan museum. Halaman *dashboard* ini didedikasikan untuk manajemen kegiatan, dan memiliki fitur utama Halaman Tambah Kegiatan, yang memungkinkan admin untuk mendaftarkan kegiatan baru melalui *form* isian yang mencakup nama, deskripsi, tanggal, penulis, dan gambar atau video kegiatan. Selain itu, terdapat Halaman Daftar Kegiatan yang menampilkan seluruh kegiatan dalam format tabel, lengkap dengan fitur tombol "edit" dan "hapus". Admin juga dapat memperbarui informasi kegiatan yang sudah ada

melalui Halaman Edit Kegiatan, di mana data lama ditampilkan kembali dalam *form* isian untuk diperbarui. Tampilan *dashboard* admin kegiatan dapat dilihat pada Gambar 3.24 hingga Gambar 3.26

**Admin Kegiatan**

Daftar Kegiatan

Tambah Kegiatan

Logout

**Tambah Kegiatan**

**Judul**

Judul kegiatan

**Deskripsi**

Deskripsi kegiatan

**Penulis**

Nama penulis

**Tanggal Kegiatan**

mm/dd/yyyy

**Foto**

Choose File No file chosen

**Video**

Choose File No file chosen

Simpan

Gambar 3.24 Halaman Tambah Kegiatan

**Admin Kegiatan**

Daftar Kegiatan

Tambah Kegiatan

Logout

**Daftar Kegiatan**

**Perpustakaan Nasional RI**

**Perpustakaan Nasional RI Menyelamatkan Naskah Kuno di Lampung**

Perpustakaan Nasional Republik Indonesia melalui Pusat Preservasi dan Alih Media Bahan Perpustakaan sukses menyelesaikan kegiatan pelestarian naskah kuno di Provinsi Lampung selama lima hari. Kegiatan ditutup dengan serah terima resmi hasil

Edit Hapus

**Perpustakaan Nasional RI**

**Perpustakaan Nasional RI Menyelamatkan Naskah Kuno di Lampung**

Perpustakaan Nasional Republik Indonesia melalui Pusat Preservasi dan Alih Media Bahan Perpustakaan sukses menyelesaikan kegiatan pelestarian naskah kuno di Provinsi Lampung selama lima hari. Kegiatan ditutup dengan serah terima resmi hasil

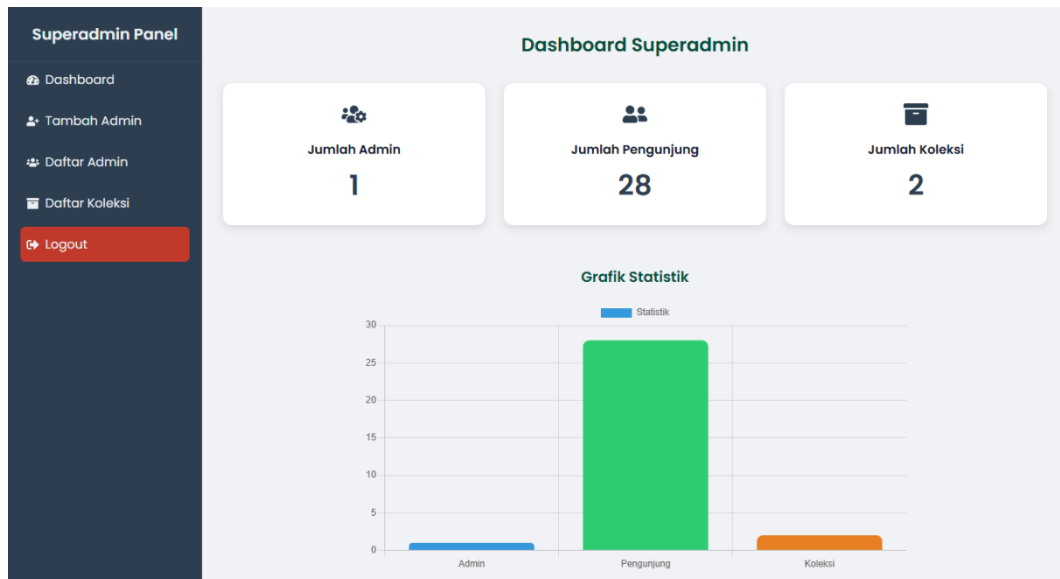
Edit Hapus

Gambar 3.25 Halaman Daftar Kegiatan

Gambar 3.26 Halaman Edit Kegiatan

#### i. Halaman *Dashboard* Superadmin

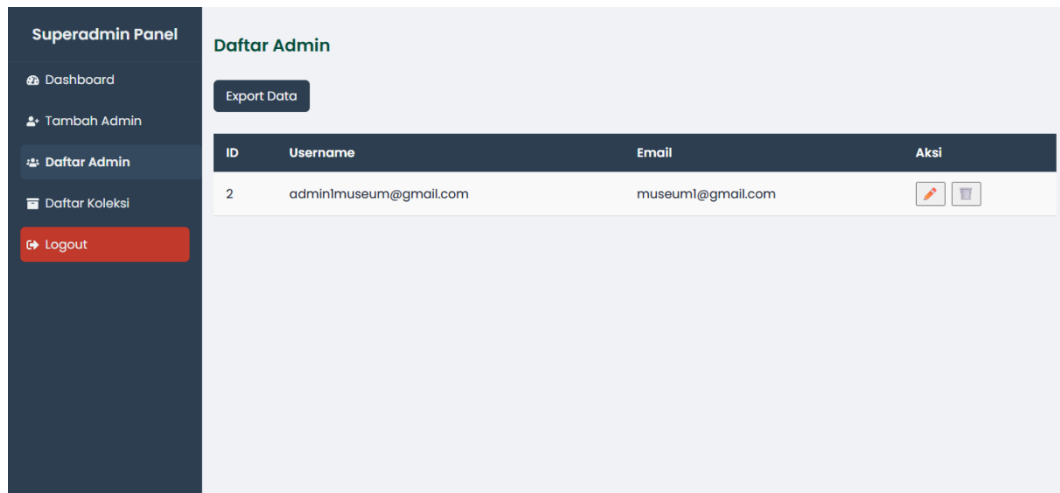
Superadmin memiliki peran tertinggi dalam sistem, dan memiliki akses ke *dashboard* manajemen admin. Di halaman ini, superadmin dapat menambahkan admin baru (baik admin koleksi maupun admin kegiatan) melalui *form* input, mengedit data admin yang sudah ada, serta menghapus akun jika diperlukan. Selain itu, superadmin juga dapat memantau seluruh koleksi dan kegiatan yang terdaftar dalam sistem melalui halaman daftar khusus. Berbeda dari admin koleksi dan admin kegiatan, superadmin tidak memiliki kontrol langsung untuk mengedit koleksi atau kegiatan, namun dapat memantau dan mengekspor data koleksi ke format CSV sebagai bentuk dokumentasi atau audit internal. Tampilan halaman *dashboard* superadmin dan seluruh komponennya dapat dilihat pada Gambar 3.27 hingga Gambar 3.7.



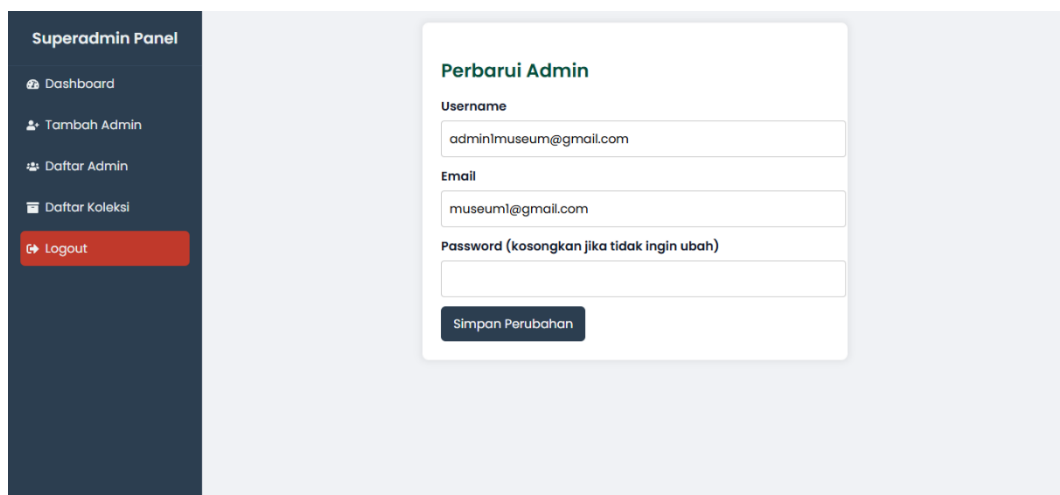
Gambar 3.27 Halaman Statistik Dashboard Superadmin

The screenshot shows the 'Superadmin Panel' with the 'Tambah Admin' option selected in the sidebar. The main content area displays a form titled 'Tambah Admin' with three input fields: 'Username' (placeholder: 'Masukkan username'), 'Email' (placeholder: 'Masukkan email'), and 'Password' (placeholder: 'Masukkan password'). A dark blue button labeled 'Tambah Admin' is positioned at the bottom of the form.

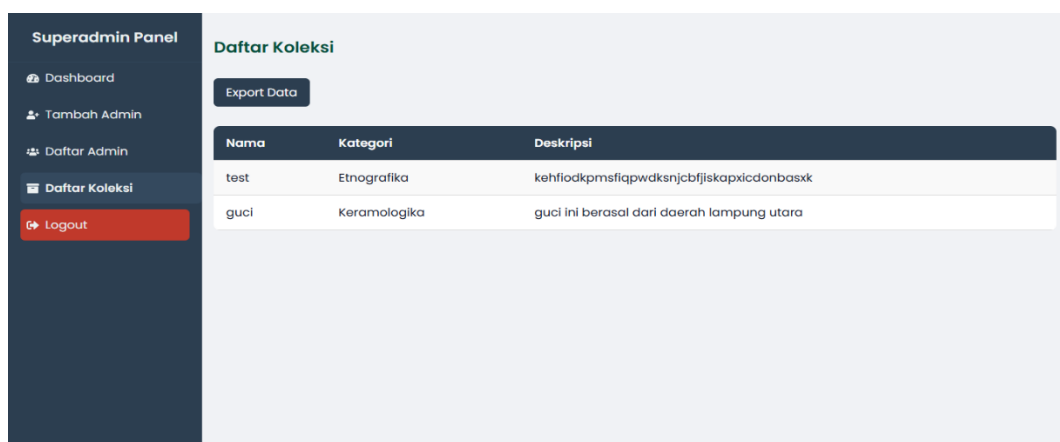
Gambar 3.28 Halaman Tambah Admin



Gambar 3.29 Halaman Daftar Admin



Gambar 3.30 Halaman Edit Admin



Gambar 3.31 Halaman Daftar Koleksi (Superadmin)



Gambar 3.32 Halaman Daftar Kegiatan (Superadmin)

### 3.3.5 Pengembangan Sistem

Pengembangan sistem digitalisasi koleksi Museum Lampung dilaksanakan dengan menerapkan arsitektur fullstack berbasis *PERN Stack*, yang terdiri atas *PostgreSQL* sebagai sistem basis data, *Express.js* sebagai kerangka kerja *backend*, *React.js* sebagai kerangka kerja *frontend*, dan *Node.js* sebagai lingkungan eksekusi *server-side* JavaScript. Pengembangan sistem ini dibagi menjadi tiga komponen utama, yaitu: *backend*, *frontend*, dan integrasi sistem. Masing-masing komponen memiliki fungsi dan tanggung jawab yang saling melengkapi dalam memastikan kinerja sistem berjalan secara optimal dan terintegrasi.

#### 3.3.5.1 Backend

Sisi *backend* bertanggung jawab dalam mengelola logika bisnis, autentikasi pengguna, komunikasi dengan basis data, dan pemrosesan data yang dibutuhkan *frontend*. *Backend* dikembangkan secara modular dan terstruktur, agar memudahkan proses pemeliharaan, pengembangan lebih lanjut, serta mendukung skalabilitas sistem.

##### A. Struktur Folder Backend

Struktur folder *backend* dirancang untuk mendukung prinsip pemisahan tanggung jawab, di mana setiap komponen dikelompokkan berdasarkan

fungsinya. Rincian struktur dapat dilihat pada Tabel 3.8 berikut:

Tabel 3.8 Struktur Folder *Backend*

| Folder/File    | Fungsi                                                                                                                                                                 |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| controllers/   | Menyimpan logika pengendali yang menangani permintaan ( <i>request</i> ) dan tanggapan ( <i>response</i> ).                                                            |
| services/      | Menjalankan logika bisnis, seperti manipulasi data atau integrasi fitur tambahan.                                                                                      |
| models/        | Mewakili skema tabel basis data menggunakan <i>Object-Relational Mapping (ORM)</i> Sequelize.                                                                          |
| routes/        | Menyediakan endpoint <i>REST API</i> yang dapat diakses oleh <i>frontend</i> .                                                                                         |
| middlewares/   | Menyediakan fungsi perantara seperti autentikasi, otorisasi, dan validasi data.                                                                                        |
| utils/         | Menyimpan fungsi bantu seperti pengunggahan <i>file</i> dan pembuatan <i>QR Code</i> .                                                                                 |
| config/        | Mengatur koneksi ke basis data serta pengelolaan variabel lingkungan ( <i>.env</i> ).                                                                                  |
| app.js         | Merupakan titik masuk utama untuk menginisialisasi server dan konfigurasi awal aplikasi.                                                                               |
| _test_/        | Menyimpan semua <i>file</i> pengujian unit dan integrasi untuk komponen <i>backend</i> . Jest secara otomatis mendeteksi dan menjalankan <i>file-file</i> di dalamnya. |
| .env.test      | Menyimpan variabel lingkungan khusus yang hanya berlaku selama eksekusi pengujian.                                                                                     |
| jest.config.js | <i>File</i> konfigurasi utama untuk <i>framework</i> pengujian <i>Jest</i> .                                                                                           |

## B. Pembuatan Model Data

Pembuatan model data dilakukan dengan memanfaatkan *Sequelize*, yaitu sebuah

*Object-Relational Mapping* (ORM) untuk *Node.js*. Setiap entitas dalam sistem seperti pengguna, koleksi, kegiatan, dan *QR code* diimplementasikan sebagai model terpisah yang mencerminkan struktur tabel dalam basis data *PostgreSQL*. Relasi antar tabel seperti satu-ke-satu (*one-to-one*) dan satu-ke-banyak (*one-to-many*) juga didefinisikan secara eksplisit, sehingga dapat mempermudah integrasi data dan validasi antar entitas.

### C. Autentikasi dan Keamanan Pengguna

Autentikasi pengguna diterapkan dengan pendekatan *stateless* menggunakan *JSON Web Token* (JWT). Setelah berhasil melakukan proses *login*, sistem akan menghasilkan token yang disimpan di sisi klien dan digunakan dalam setiap permintaan yang membutuhkan otorisasi. Token ini memuat informasi penting seperti identitas dan peran pengguna (superadmin, admin koleksi, atau admin kegiatan), yang kemudian diverifikasi oleh *middleware* di *backend* sebelum akses diberikan.

Demi menjaga kerahasiaan informasi, khususnya kata sandi, sistem menggunakan algoritma *bcrypt* untuk proses *hashing*. Penambahan *salt* (data acak unik) dalam proses *hashing* memastikan bahwa dua kata sandi identik tetap menghasilkan *hash* yang berbeda, sehingga memperkuat keamanan terhadap serangan *rainbow table* dan *brute force*. Selain itu, sistem juga menerapkan kontrol akses berbasis peran (*Role-Based Access Control*, RBAC) untuk membatasi akses pengguna berdasarkan kewenangannya.

### D. Operasi CRUD dan Validasi Data

Seluruh pengelolaan data dalam sistem *backend* dilakukan melalui layanan *REST API* dengan menerapkan metode *Create*, *Read*, *Update*, dan *Delete* (CRUD). Setiap *endpoint* dalam sistem dikembangkan sesuai dengan standar metode *HTTP*, yang terdiri atas metode *GET* untuk mengambil data seperti daftar koleksi, pengguna, atau kegiatan; *POST* untuk menambahkan entri baru ke dalam sistem, seperti akun atau data koleksi; *PUT* untuk memperbarui informasi yang telah ada sebelumnya; serta *DELETE* untuk menghapus data yang tidak lagi diperlukan.

Sebelum suatu permintaan diproses lebih lanjut oleh server, data yang diterima akan divalidasi terlebih dahulu melalui *middleware*. Proses validasi ini bertujuan untuk memastikan bahwa data yang dikirim telah sesuai dengan format yang ditentukan, lengkap, dan aman dari potensi ancaman. Validasi ini sangat krusial dalam menjaga konsistensi dan integritas data dalam sistem, serta mencegah serangan berbasis input yang berbahaya, seperti *script injection*, yang dapat mengancam keamanan dan stabilitas aplikasi.

#### **E. Pengunggahan Media dan Pembuatan Kode *QR***

Sistem mendukung pengunggahan media seperti gambar koleksi, video 360°, serta dokumentasi kegiatan. *File* media tidak disimpan langsung pada server aplikasi, melainkan dikelola melalui layanan penyimpanan eksternal berbasis *cloud*, seperti Cloudinary. Hal ini dilakukan untuk meningkatkan efisiensi dan kecepatan akses *file*, sekaligus mengurangi beban penyimpanan pada server utama.

Selain itu, setiap koleksi yang ditambahkan akan secara otomatis dihubungkan dengan *QR Code* unik. *QR Code* tersebut akan mengarah ke halaman detail koleksi di *frontend*, dan dapat dipindai oleh pengunjung secara langsung di lokasi fisik museum, sehingga mendukung integrasi informasi digital dalam ruang pameran nyata.

#### **3.3.5.2 Frontend**

*Frontend* merupakan bagian sistem yang menyajikan antarmuka kepada pengguna dan menjadi media utama interaksi dengan sistem. Pengembangan *frontend* dilakukan menggunakan *React.js* yang mendukung pendekatan berbasis komponen dan bersifat dinamis.

##### **A. Struktur Folder *Frontend***

Struktur folder *frontend* disusun untuk memisahkan komponen antarmuka berdasarkan fungsionalitasnya. Penjelasan tiap folder ditunjukkan pada Tabel 3.9 berikut:

Tabel 3.9 Struktur Folder *Frontend*

| Folder/File | Fungsi Utama                                                                                                  |
|-------------|---------------------------------------------------------------------------------------------------------------|
| src/        | Direktori utama tempat seluruh kode sumber disimpan.                                                          |
| components/ | Menyimpan komponen <i>UI</i> yang digunakan kembali antar halaman.                                            |
| pages/      | Berisi komponen halaman utama seperti beranda, detail koleksi, dan lainnya.                                   |
| services/   | Menyediakan fungsi untuk komunikasi dengan <i>backend</i> melalui <i>API</i> .                                |
| routes/     | Mengatur navigasi dan rute aplikasi menggunakan <i>React Router</i> .                                         |
| styles/     | Menyimpan berkas <i>CSS</i> untuk pengaturan tampilan visual.                                                 |
| App.js      | Komponen utama yang menyusun struktur dan routing aplikasi secara menyeluruh.                                 |
| index.js    | Titik masuk aplikasi yang merender tampilan ke dalam <i>Document Object Model (DOM)</i> .                     |
| .env        | Menyimpan variabel lingkungan yang digunakan oleh aplikasi <i>frontend</i> , seperti <i>URL API backend</i> . |
| .gitignore  | Memberitahu <i>Git file</i> dan folder mana yang harus diabaikan dan tidak dilacak dalam repositori.          |

## B. Manajemen *State* Aplikasi

*Frontend* memanfaatkan *React Hooks*, khususnya *useState* dan *useEffect*, untuk mengelola *state* aplikasi secara terpusat. Pendekatan ini memfasilitasi koordinasi data antar-komponen, menjaga konsistensi informasi, dan meminimalkan duplikasi data. Penggunaan *Hooks* menyederhanakan pengelolaan *state* dan efek samping dalam komponen fungsional, menghasilkan kode yang lebih ringkas, mudah dibaca, dan mudah dipelihara. Selain itu, *Hooks* meningkatkan

komposabilitas dengan memungkinkan ekstraksi dan penggunaan kembali logika *stateful* di berbagai komponen, yang esensial untuk skalabilitas aplikasi.

### **C. Desain antarmuka responsif dan aksesibel**

Tampilan antarmuka dirancang agar bersifat responsif terhadap berbagai ukuran layar dan perangkat. Desain menggunakan kombinasi *flexbox*, *grid layout* untuk memastikan pengalaman pengguna tetap optimal di perangkat desktop, tablet, maupun ponsel. Selain itu, sistem juga memperhatikan aspek aksesibilitas dengan memperhatikan ukuran teks, kontras warna, dan struktur navigasi yang mudah diakses, sehingga dapat digunakan oleh pengguna dengan beragam kebutuhan.

#### **3.3.5.3 Integrasi Sistem**

Integrasi sistem dilakukan melalui komunikasi antara *frontend* dan *backend* menggunakan *REST API* yang memanfaatkan protokol *HTTP* dan format pertukaran data berbasis *JSON*. Setiap interaksi dari pengguna pada sisi *frontend*, seperti proses autentikasi, permintaan data koleksi, atau pengelolaan akun, akan dikirim sebagai permintaan *HTTP* ke *endpoint backend* yang sesuai.

*Backend* akan memproses permintaan tersebut, yang mungkin mencakup akses ke basis data atau media penyimpanan eksternal, dan mengembalikan respons dalam bentuk data *JSON*. Untuk *endpoint* yang memerlukan autentikasi, *frontend* menyisipkan *JWT token* dalam *header* permintaan, yang kemudian diverifikasi oleh *backend* sebelum akses diberikan.

#### **3.3.6 Pengujian Sistem**

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fitur yang dikembangkan dalam sistem digitalisasi koleksi Museum Lampung berjalan sesuai spesifikasi dan dapat digunakan secara optimal oleh pengguna.

##### **1. Pengujian dengan *Postman***

Pengujian menggunakan *Postman* dilakukan untuk mengevaluasi fungsionalitas *endpoint* pada sisi backend. Melalui metode *HTTP* seperti *GET*, *POST*, *PUT*, dan *DELETE*, pengujian ini bertujuan untuk memastikan bahwa proses *CRUD* (*Create*,

*Read, Update, Delete*) pada entitas seperti koleksi, pengguna, dan kegiatan dapat berjalan sesuai dengan logika bisnis. Selain itu, pengujian juga mencakup mekanisme otentikasi dan otorisasi berbasis JWT (*JSON Web Token*) untuk memastikan bahwa akses ke fitur tertentu hanya diberikan kepada peran pengguna yang berwenang, seperti admin dan superadmin. Pengujian ini juga memperhatikan respons server terhadap skenario keberhasilan maupun kegagalan, seperti validasi input dan status akses.

## **2. Pengujian *Black-Box***

Pengujian *black-box* diterapkan pada sisi *frontend* untuk memverifikasi perilaku sistem dari sudut pandang pengguna tanpa memeriksa struktur kode internal. Pengujian ini mencakup skenario penggunaan utama, seperti proses *login*, navigasi antar halaman, interaksi admin dalam mengelola data koleksi dan kegiatan, serta interaksi pengguna umum dalam menelusuri dan mencari koleksi. Fokus utama dari pengujian ini adalah memastikan bahwa seluruh elemen antarmuka dapat merespons input pengguna secara tepat, menampilkan data dengan benar, serta memberikan umpan balik dan validasi sesuai dengan ekspektasi.

## **3. Pengujian unit dengan *Jest***

Sebagai pelengkap pengujian dengan *Postman*, pengujian unit dilakukan guna memastikan bahwa setiap komponen logika pada sisi *backend* bekerja secara mandiri sesuai dengan fungsinya. *Framework Jest* digunakan dalam proses ini karena kemampuannya dalam menyediakan lingkungan pengujian yang efisien, cepat, dan mendukung isolasi fungsi melalui teknik *mocking* terhadap dependensi eksternal seperti layanan, middleware, atau utilitas.

Pengujian unit mencakup berbagai komponen utama, termasuk middleware otorisasi, validator *input*, serta layanan internal yang menangani pemrosesan data koleksi, pengguna, dan kegiatan. Setiap fungsi diuji berdasarkan skenario keberhasilan dan kegagalan untuk mengevaluasi stabilitas respons terhadap berbagai kondisi, seperti data yang valid, tidak valid, serta absennya token otorisasi.

Pada middleware otorisasi, pengujian menilai ketepatan sistem dalam mengelola akses berdasarkan peran pengguna, seperti admin dan superadmin. Pada sisi validator, pengujian mengevaluasi kriteria validasi data seperti panjang minimal karakter, format email, serta kategori koleksi yang diizinkan. Selain itu, utilitas unggah *file* juga diuji untuk memverifikasi proses konversi dan penyimpanan media ke layanan eksternal.

Pendekatan pengujian ini dirancang untuk memberikan keyakinan bahwa setiap bagian sistem *backend* telah diuji secara menyeluruh sebelum diintegrasikan ke dalam sistem secara keseluruhan. Pengujian dengan Jest memperkuat keandalan dan konsistensi logika aplikasi, serta meminimalkan kemungkinan terjadinya regresi (*regression*) selama proses pemeliharaan dan pengembangan lanjutan.

### 3.3.7 *Deployment Website*

Setelah tahap pengujian berhasil dilakukan dan sistem terbukti berjalan stabil, langkah selanjutnya adalah melakukan *deployment* agar aplikasi dapat diakses secara publik. Dalam pengembangan sistem ini, proses *deployment* terbagi menjadi dua bagian utama, yaitu *backend* dan *frontend*, dengan memanfaatkan layanan *cloud* modern yang mendukung integrasi dan otomatisasi.

Untuk *backend*, digunakan layanan Railway sebagai platform utama hosting. Railway memungkinkan integrasi langsung dengan repository GitHub, sehingga setiap pembaruan kode dapat langsung diterapkan secara otomatis. Aplikasi *backend* berbasis *Node.js* dan database *PostgreSQL* dijalankan melalui layanan ini dengan konfigurasi *environment variable* seperti *URL* database, *JWT secret*, dan kredensial lainnya yang dikelola aman melalui *dashboard* Railway. Railway juga menangani proses *deployment pipeline* dan menjalankan server tanpa perlu konfigurasi manual tambahan.

Sementara itu, *frontend* yang dibangun menggunakan *React.js* dideploy ke Vercel, sebuah platform *serverless hosting* yang mendukung *Continuous Deployment*. Setiap kali terdapat perubahan pada repository GitHub (misalnya melalui *commit* ke *branch main*), Vercel akan secara otomatis menjalankan proses build React dan

mempublikasikannya tanpa perlu menjalankan perintah *npm run build* secara manual. Pengaturan routing untuk aplikasi *Single Page Application* (SPA) seperti React juga ditangani otomatis oleh Vercel, sehingga tidak diperlukan konfigurasi *.htaccess* seperti pada server tradisional. Vercel juga menyediakan optimasi performa seperti *Content Delivery Network* (CDN) untuk mempercepat waktu akses pengguna. Dengan konfigurasi *deployment* seperti ini, sistem digitalisasi koleksi Museum Lampung dapat berjalan dengan efisien dan dapat diakses secara luas oleh pengguna melalui domain publik yang telah dikaitkan.

### 3.3.8 Analisis Capaian

Setelah sistem berhasil dikembangkan dan diimplementasikan, dilakukan analisis terhadap capaian sistem berdasarkan aspek fungsional dan non-fungsional. Analisis ini bertujuan untuk menilai sejauh mana sistem memenuhi kebutuhan pengguna dan tujuan pengembangan. Adapun indikator capaian sistem meliputi:

1. Fungsionalitas: Seluruh fitur utama seperti *login*, pengelolaan koleksi, pencarian koleksi, pengelolaan kegiatan, integrasi media 360°, dan *QR Code* telah berjalan sesuai dengan rancangan awal.
2. Keamanan: Sistem telah menerapkan otentikasi berbasis *JSON Web Token* (*JWT*) untuk membedakan hak akses antara pengguna umum dan admin, serta mencegah akses tidak sah terhadap fitur terbatas.
3. Aksesibilitas: Sistem dapat diakses melalui berbagai perangkat, termasuk *desktop*, *tablet*, dan *smartphone*, sehingga meningkatkan jangkauan dan kemudahan akses bagi pengunjung.
4. Antarmuka Pengguna: Tampilan aplikasi dirancang responsif dan intuitif, memudahkan interaksi pengguna dalam menjelajahi sistem, baik dari sisi admin maupun pengunjung umum.

### 3.3.9 Laporan Penelitian

Tahap akhir dari penelitian adalah menyusun laporan penelitian yang bertujuan memberikan gambaran menyeluruh mengenai proses, metodologi, hasil, dan analisis dalam pengembangan sistem digitalisasi koleksi Museum Lampung. Laporan ini mendokumentasikan setiap tahapan pengembangan sistem berbasis

*PERN Stack* serta mengevaluasi capaian yang diperoleh, sehingga dapat menjadi referensi untuk pengembangan lebih lanjut.

## IV. HASIL DAN PEMBAHASAN

### 4.1 Implementasi Sistem

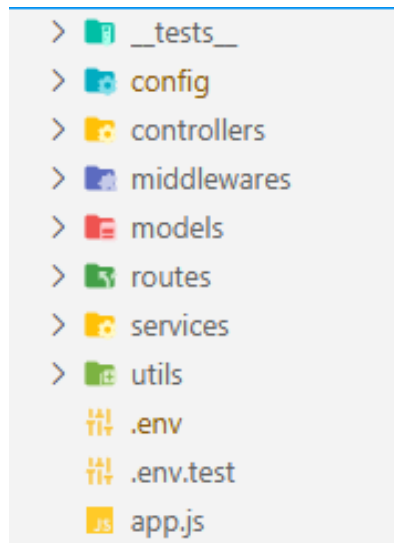
Pengembangan sistem digitalisasi koleksi Museum Lampung dilakukan dengan mengadopsi arsitektur *PERN Stack* (*PostgreSQL, Express.js, React.js, Node.js*), yang merupakan kombinasi teknologi *open-source* terintegrasi untuk membangun aplikasi *web* yang dinamis dan responsif. Pendekatan ini dipilih karena kemampuannya dalam menyediakan alur kerja yang efisien, skalabilitas tinggi, serta fleksibilitas dalam pengelolaan data dan antarmuka pengguna. Proses pengembangan dibagi menjadi tiga komponen utama: *backend*, *frontend*, dan integrasi sistem, yang masing-masing memiliki fungsi dan tanggung jawab saling melengkapi guna memastikan kinerja sistem berjalan optimal dan terintegrasi.

#### 4.1.1 Implementasi *Backend*

Sisi *backend* memegang peranan krusial dalam mengelola logika bisnis, autentikasi pengguna, komunikasi dengan basis data, dan pemrosesan data yang dibutuhkan oleh *frontend*. Pengembangan *backend* dirancang secara modular dan terstruktur untuk mempermudah pemeliharaan, pengembangan lanjutan, serta mendukung skalabilitas sistem.

##### 4.1.1.1 Struktur Proyek *Backend*

Struktur folder *backend* dirancang untuk mendukung prinsip pemisahan tanggung jawab (*separation of concerns*), di mana setiap komponen dikelompokkan berdasarkan fungsinya. Organisasi ini memfasilitasi navigasi kode dan kolaborasi tim pengembangan. Direktori utama *backend/* mencakup *file* konfigurasi aplikasi seperti *app.js*, *.env*, *.env.test*, dan *jest.config.js*, serta sub-direktori untuk *config/*, *controllers/*, *middlewares/*, *models/*, *routes/*, *services/*, *utils/*, dan *tests/* yang dapat dilihat pada Gambar 4.1.



Gambar 4.1 Struktur Folder *Backend*

*File* utama `app.js` berfungsi sebagai titik masuk utama untuk menginisialisasi server dan konfigurasi awal aplikasi. Direktori `config/` menyimpan konfigurasi aplikasi dan basis data, sementara `controllers/` menangani permintaan (*request*) dan tanggapan (*response*). Fungsi perantara seperti autentikasi dan validasi diatur dalam `middlewares/`, sedangkan skema basis data dan relasi didefinisikan dalam `models/`. Definisi *endpoint API* berada di `routes/`, logika bisnis diimplementasikan dalam `services/`, dan fungsi bantu seperti unggah *file* serta validasi dikelompokkan dalam `utils/`. Selain itu, tersedia juga direktori `tests/` yang berisi skrip-skrip pengujian unit maupun integrasi untuk memastikan stabilitas dan keandalan sistem pada setiap tahap pengembangan.

#### 4.1.1.2 Implementasi Model Data

Implementasi model data dilakukan dengan memanfaatkan *Sequelize*, sebuah *Object-Relational Mapping* (ORM) untuk *Node.js*. Setiap entitas dalam sistem, seperti pengguna, koleksi, kegiatan, dan *QR Code*, diimplementasikan sebagai model terpisah yang mencerminkan struktur tabel dalam basis data PostgreSQL. Relasi antar tabel seperti satu-ke-satu (*one-to-one*) dan satu-ke-banyak (*one-to-many*) juga didefinisikan secara eksplisit, sehingga mempermudah integrasi data dan validasi antar entitas.

|   | id<br>[PK] integer | name<br>character varying | description<br>text | year<br>integer | image<br>character varying (255) | video360<br>character varying (255) | imagePublicId<br>character varying (255) | video360PublicId<br>character varying (255) | category<br>'enum_Collections_category' | createdAt<br>timestamp with time zone | updatedAt<br>timestamp with time zone |
|---|--------------------|---------------------------|---------------------|-----------------|----------------------------------|-------------------------------------|------------------------------------------|---------------------------------------------|-----------------------------------------|---------------------------------------|---------------------------------------|
| 1 | 1                  | Mangkuk T...              | Bentuk mangkok...   | 1995            | https://res.cloudinar...         | [null]                              | museum-lampung-...                       | [null]                                      | Keramologika                            | 2025-07-...                           | 2025-07-10 14:57:20.105...            |
| 2 | 2                  | Guci                      | Guci tertutup, m... | 1995            | https://res.cloudinar...         | [null]                              | museum-lampung-...                       | [null]                                      | Keramologika                            | 2025-07-...                           | 2025-07-10 14:59:44.114...            |

Gambar 4.2 Tabel Collections

|   | id<br>[PK] integer | username<br>character varying | email<br>character varying | password<br>character varying (255)               | role<br>'enum_Users_role' | createdAt<br>timestamp with time zone | updatedAt<br>timestamp with time zone |
|---|--------------------|-------------------------------|----------------------------|---------------------------------------------------|---------------------------|---------------------------------------|---------------------------------------|
| 1 | 1                  | superadmin                    | superadmin@...             | \$2b\$10\$JEwhcx5BoT0cyjf6heE6LuNf.S.FKgiqtQam... | superadmin                | 2025-07-10 13:31:25.063768+07         | 2025-07-10 13:31:25.063768+07         |
| 2 | 4                  | Yogi Aprio                    | yogiaprio21@...            | \$2b\$10\$RB1MlvkYPV77w9X1g7ir.IV/Mo1VDRDvT...    | admin_event               | 2025-07-10 13:38:07.733+07            | 2025-07-10 13:38:07.733+07            |
| 3 | 5                  | Koleksiadmin                  | koleksiadmin...            | \$2b\$10\$4Es7v3XFR0Tva.sQFull.2osJ/vHa07zUGC...  | admin_collection          | 2025-07-10 14:28:30.377+07            | 2025-07-10 14:28:30.377+07            |

Gambar 4.3 Tabel Users

|   | id<br>[PK] integer | title<br>character varying (255) | description<br>text | imageUrl<br>character varying (255) | videoUrl<br>character v... | imagePublicId<br>character varying (255) | videoPublicId<br>character varyin | creatorId<br>integer | author<br>character | activityDate<br>timestamp with | createdAt<br>timestamp with | updatedAt<br>timestamp |
|---|--------------------|----------------------------------|---------------------|-------------------------------------|----------------------------|------------------------------------------|-----------------------------------|----------------------|---------------------|--------------------------------|-----------------------------|------------------------|
| 1 | 1                  | Perpustakaan Nasion...           | Perpustakaan N...   | https://res.cloudinary.com/dbjo...  | [null]                     | museum-lampung-coll...                   | [null]                            | 4                    | Yogi                | 2025-02-04 ...                 | 2025-07-10 ...              | 202...                 |
| 2 | 2                  | Perpustakaan Nasion...           | Perpustakaan N...   | https://res.cloudinary.com/dbjo...  | [null]                     | museum-lampung-coll...                   | [null]                            | 4                    | Yogi                | 2025-02-04 ...                 | 2025-07-10 ...              | 202...                 |

Gambar 4.4 Tabel Events

|   | id<br>[PK] integer | ipAddress<br>character varying (255) | createdAt<br>timestamp with time zone | updatedAt<br>timestamp with time zone |
|---|--------------------|--------------------------------------|---------------------------------------|---------------------------------------|
| 1 | 1                  | ::1                                  | 2025-07-10 13:32:29.934+07            | 2025-07-10 13:32:29.934+07            |
| 2 | 2                  | ::1                                  | 2025-07-12 20:43:41.386+07            | 2025-07-12 20:43:41.386+07            |
| 3 | 3                  | ::1                                  | 2025-07-14 02:12:23.03+07             | 2025-07-14 02:12:23.03+07             |

Gambar 4.5 Tabel Visitors

|   | id<br>[PK] integer | collectionId<br>integer | qrCodeUrl<br>character varying (255)                                                                         | createdAt<br>timestamp with time zone | updatedAt<br>timestamp with time zone |
|---|--------------------|-------------------------|--------------------------------------------------------------------------------------------------------------|---------------------------------------|---------------------------------------|
| 1 | 1                  | 1                       | https://res.cloudinary.com/dbjormvzq/image/upload/v1752134240/museum-lampung-qr-codes/qr-1-1752134240271.png | 2025-07-10 14:57:21.383+07            | 2025-07-10 14:57:21.383+07            |
| 2 | 2                  | 2                       | https://res.cloudinary.com/dbjormvzq/image/upload/v1752134384/museum-lampung-qr-codes/qr-2-1752134384233.png | 2025-07-10 14:59:45.11+07             | 2025-07-10 14:59:45.11+07             |

Gambar 4.6 Tabel QR Code

#### 4.1.1.3 Autentikasi dan Keamanan Pengguna

Autentikasi pengguna diimplementasikan dengan pendekatan *stateless* menggunakan JSON Web Token (JWT). Setelah berhasil melakukan proses *login*, sistem akan menghasilkan token yang disimpan di sisi klien dan digunakan dalam

setiap permintaan yang membutuhkan otorisasi. Token ini memuat informasi penting seperti identitas dan peran pengguna (*superadmin*, *admin koleksi*, atau *admin kegiatan*), yang kemudian diverifikasi oleh *middleware* di *backend* sebelum akses diberikan. Potongan kode untuk autentikasi dapat dilihat pada Gambar 4.7.

```
exports.authorizeSuperadmin = (req, res, next) => {
  if (req.user.role !== 'superadmin') {
    return res.status(403).json({ message: 'Akses khusus superadmin' });
  }
  next();
};

Windsurf: Refactor | Explain | Generate JSDoc | X
exports.authorizeAdminCollection = (req, res, next) => {
  if (req.user.role !== 'admin_collection') {
    return res.status(403).json({ message: 'Akses khusus admin koleksi' });
  }
  next();
};

Windsurf: Refactor | Explain | Generate JSDoc | X
exports.authorizeAdminEvent = (req, res, next) => {
  if (req.user.role !== 'admin_event') {
    return res.status(403).json({ message: 'Akses khusus admin event' });
  }
  next();
};
```

Gambar 4.7 Kode Autentikasi

Demi menjaga kerahasiaan informasi, khususnya kata sandi, sistem menggunakan algoritma *bcrypt* untuk proses *hashing*. Penambahan *salt* (data acak unik) dalam proses *hashing* memastikan bahwa dua kata sandi identik tetap menghasilkan *hash* yang berbeda, sehingga memperkuat keamanan terhadap serangan *rainbow table* dan *brute force*. Selain itu, sistem juga menerapkan kontrol akses berbasis peran (*Role-Based Access Control*, RBAC) untuk membatasi akses pengguna berdasarkan kewenangannya. Kode *hashing* password dapat dilihat pada Gambar 4.8.

```
hooks: {
  Windsurf: Refactor | Explain | Generate JSDoc
  beforeCreate: async (user) => {
    user.password = await bcrypt.hash(user.password, 10);
  },
  Windsurf: Refactor | Explain | Generate JSDoc
  beforeUpdate: async (user) => {
    if (user.changed('password')) {
      user.password = await bcrypt.hash(user.password, 10);
    }
  }
}
```

Gambar 4.8 Kode Hashing Password

#### 4.1.1.4 Operasi *CRUD* dan Validasi Data

Seluruh pengelolaan data dalam sistem *backend* dilakukan melalui layanan *REST API* dengan menerapkan metode *Create*, *Read*, *Update*, dan *Delete (CRUD)*. Setiap *endpoint* dalam sistem dikembangkan sesuai dengan standar metode *HTTP*, yang terdiri atas metode *GET* untuk mengambil data seperti daftar koleksi, pengguna, atau kegiatan; *POST* untuk menambahkan entri baru ke dalam sistem, seperti akun atau data koleksi; *PUT* untuk memperbarui informasi yang telah ada sebelumnya; serta *DELETE* untuk menghapus data yang tidak lagi diperlukan. Salah satu kode untuk pengaturan rute *API* koleksi dapat dilihat pada Gambar 4.9.

```
router.post('/',
  authMiddleware.authenticateUser,
  authMiddleware.authorizeAdminCollection,
  uploadFields,
  collectionController.createCollection
);

router.put(
 ('/:id',
  authMiddleware.authenticateUser,
  authMiddleware.authorizeAdminCollection,
  uploadFields,
  collectionController.updateCollection
);

router.get('/:id', collectionController.getCollectionById);
router.get('/', collectionController.getAllCollections);

router.delete('/:id',
  authMiddleware.authenticateUser,
  authMiddleware.authorizeAdminCollection,
  collectionController.deleteCollection
);
```

Gambar 4.9 Kode Rute *API* Koleksi

Sebelum suatu permintaan diproses lebih lanjut oleh *server*, data yang diterima akan divalidasi terlebih dahulu melalui *middleware*. Proses validasi ini bertujuan untuk memastikan bahwa data yang dikirim telah sesuai dengan format yang ditentukan, lengkap, dan aman dari potensi ancaman. Validasi ini sangat krusial dalam menjaga konsistensi dan integritas data dalam sistem, serta mencegah serangan berbasis *input* yang berbahaya, seperti *script injection*, yang dapat mengancam keamanan dan stabilitas aplikasi. Salah satu contoh kode, yaitu validasi koleksi dapat dilihat pada Gambar 4.10.

```

exports.validateCollection = async (name, description, year, image, video360, category) => {
  if (!name || name.length < 3) throw new Error('Nama koleksi harus minimal 3 karakter');
  if (!description || description.length < 10) throw new Error('Deskripsi koleksi harus minimal 10 karakter');
  if (!year || year < 1900 || year > new Date().getFullYear()) throw new Error('Tahun koleksi tidak valid');
  const validCategories = ['Keramologika', 'Etnografika', 'Filologika', 'Numismatika', 'Arkeologika', 'Histori'];
  if (!validCategories.includes(category)) throw new Error('Kategori tidak valid');
};

```

Gambar 4.10 Kode Validasi Koleksi

#### 4.1.1.5 Pengunggahan Media dan Pembuatan Kode QR

Sistem mendukung pengunggahan media seperti gambar koleksi, video 360°, serta dokumentasi kegiatan. *File* media tidak disimpan langsung pada *server* aplikasi, melainkan dikelola melalui layanan penyimpanan eksternal berbasis *cloud*, yaitu Cloudinary. Pendekatan ini dipilih untuk meningkatkan efisiensi dan kecepatan akses *file*, sekaligus mengurangi beban penyimpanan pada *server* utama. Pengaturan untuk upload koleksi sapat dilihat pada Gambar 4.11.

```

exports.uploadFile = async (file) => {
  if (!file || !file.buffer || !file.mimetype) {
    throw new Error('File tidak valid untuk diunggah');
  }

  return new Promise((resolve, reject) => {
    let resourceType = 'auto';
    if (file.mimetype.startsWith('video/')) resourceType = 'video';

    const uploadStream = cloudinary.uploader.upload_stream(
      {
        resource_type: resourceType,
        folder: 'museum-lampung-collections',
        public_id: `upload_${Date.now()}`
      },
      (error, result) => {
        if (error) return reject(error);
        resolve({
          url: result.secure_url,
          publicId: result.public_id
        });
      }
    );

    streamifier.createReadStream(file.buffer).pipe(uploadStream);
  });
};

```

Gambar 4.11 Kode Upload Koleksi ke Cloudinary

Selain itu, setiap koleksi yang ditambahkan akan secara otomatis dihubungkan dengan *QR Code* unik. *QR Code* tersebut akan mengarah ke halaman detail koleksi di *frontend*, dan dapat dipindai oleh pengunjung secara langsung di lokasi fisik

museum, sehingga mendukung integrasi informasi digital dalam ruang eksibisi nyata. Kode untuk melakukan pembuatan *QR Code* dapat dilihat pada Gambar 4.12.

```
exports.generateQRCode = async (url, collectionId) => {
  try {
    const buffer = await QRCode.toBuffer(url);

    const qrCodeUrl = await new Promise((resolve, reject) => {
      const uploadStream = cloudinary.uploader.upload_stream(
        {
          folder: 'museum-lampung-qrcodes',
          resource_type: 'image',
          format: 'png',
          public_id: `qr-${collectionId}-${Date.now()}`
        },
        (error, result) => {
          if (error) return reject(error);
          resolve(result.secure_url);
        }
      );
      streamifier.createReadStream(buffer).pipe(uploadStream);
    });

    return qrCodeUrl;
  } catch (err) {
    console.error("Gagal mengunggah QR Code:", err);
    throw new Error('Gagal menyimpan QR Code ke Cloudinary');
  }
};
```

Gambar 4.12 kode Pembuatan *QR Code*

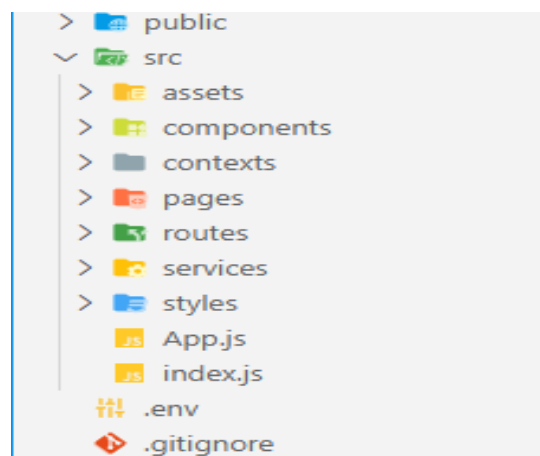
Penggunaan autentikasi berbasis JWT dengan *hashing* kata sandi menggunakan *bcrypt*, ditambah dengan validasi data yang ketat pada setiap operasi *CRUD*, serta pengelolaan media dan *QR Code* melalui layanan *cloud* eksternal, secara kolektif meningkatkan keamanan data dan efisiensi operasional sistem. Mekanisme ini memastikan bahwa data sensitif pengguna terlindungi dari akses tidak sah dan serangan siber, sementara validasi *input* mencegah korupsi data dan kerentanan aplikasi. Selain itu, pendelegasian penyimpanan media ke Cloudinary mengurangi beban server utama dan meningkatkan kecepatan akses konten, yang pada gilirannya mengoptimalkan kinerja aplikasi secara keseluruhan. Pendekatan ini tidak hanya mengamankan sistem tetapi juga menciptakan fondasi yang kokoh untuk skalabilitas dan keandalan jangka panjang, memungkinkan museum untuk mengelola koleksi digitalnya dengan lebih efektif dan aman.

#### 4.1.2 Implementasi *Frontend*

*Frontend* merupakan bagian sistem yang menyajikan antarmuka kepada pengguna dan menjadi media utama interaksi dengan sistem. Pengembangan *frontend* dilakukan menggunakan *React.js* yang mendukung pendekatan berbasis komponen dan bersifat dinamis.

##### 4.1.2.1 Struktur Folder *Frontend*

Struktur folder *frontend* disusun untuk memisahkan komponen antarmuka berdasarkan fungsionalitasnya, sehingga memudahkan proses pengembangan, pemeliharaan, dan kolaborasi tim. Seluruh kode sumber disimpan dalam direktori `src/`, yang di dalamnya terdapat beberapa sub-direktori penting, yaitu `components/`, `pages/`, `services/`, `routes/`, `styles/`, dan `assets/`. Selain itu, terdapat juga direktori `public/` yang menyimpan berkas statis yang dapat diakses langsung oleh browser. Struktur folder frontend dapat dilihat pada Gambar 4.13.



Gambar 4.13 Struktur Folder *Frontend*

Folder `components/` berisi komponen antarmuka pengguna (*UI components*) yang dapat digunakan kembali di berbagai halaman, misalnya komponen kartu koleksi atau bilah navigasi. Folder `pages/` digunakan untuk menyimpan komponen halaman utama aplikasi seperti beranda, halaman detail koleksi, dan halaman administratif. Selanjutnya, folder `services/` menyimpan fungsi-fungsi yang berfungsi sebagai

penghubung antara *frontend* dengan *backend* melalui pemanggilan *API*, sedangkan folder *routes/* digunakan untuk mengatur navigasi dan rute aplikasi dengan memanfaatkan *React Router*. Folder *styles/* menyimpan berkas-berkas untuk mengatur tampilan visual aplikasi, dan folder *assets/* digunakan untuk menyimpan sumber daya statis seperti gambar, ikon, font, atau video yang digunakan pada antarmuka aplikasi. Sementara itu, folder *public/* menyimpan berkas statis yang dapat diakses langsung tanpa melalui proses *bundling*, seperti *index.html*, *favicon*, atau berkas konfigurasi manifest.

Selain sub-direktori tersebut, terdapat juga *file* utama yang menjalankan aplikasi. *File App.js* berfungsi sebagai komponen utama yang menyusun struktur dan mendefinisikan routing aplikasi secara menyeluruh. *File index.js* berperan sebagai titik masuk (*entry point*) yang merender seluruh aplikasi ke dalam *Document Object Model* (DOM) pada browser. *File .env* digunakan untuk menyimpan variabel lingkungan (*environment variables*) seperti *URL API* atau kunci layanan pihak ketiga yang diperlukan oleh aplikasi. Sedangkan *file .gitignore* digunakan untuk menentukan berkas atau direktori yang diabaikan oleh sistem kontrol versi Git, misalnya folder *node\_modules/* atau *file* konfigurasi yang bersifat sensitif.

#### 4.1.2.2 Manajemen State Aplikasi

*Frontend* memanfaatkan *React Hooks*, khususnya *useState* dan *useEffect*, untuk mengelola *state* aplikasi secara terpusat. Pendekatan ini memfasilitasi koordinasi data antar-komponen, menjaga konsistensi informasi, dan meminimalkan duplikasi data. Penggunaan Hooks menyederhanakan pengelolaan *state* dan efek samping dalam komponen fungsional, menghasilkan kode yang lebih ringkas, mudah dibaca, dan mudah dipelihara. Selain itu, *Hooks* meningkatkan komposabilitas dengan memungkinkan ekstraksi dan penggunaan kembali logika *stateful* di berbagai komponen, yang esensial untuk skalabilitas aplikasi. Untuk *state* global, seperti preferensi bahasa, *LanguageContext* digunakan untuk menyediakan akses *state* ke seluruh komponen yang membutuhkannya.

#### 4.1.2.3 Desain Antarmuka Responsif dan Aksesibel

Tampilan antarmuka dirancang agar bersifat responsif terhadap berbagai ukuran layar dan perangkat. Desain menggunakan kombinasi *flexbox* dan *grid layout* untuk memastikan pengalaman pengguna tetap optimal di perangkat *desktop*, *tablet*, maupun ponsel. Selain itu, sistem juga memperhatikan aspek aksesibilitas dengan memperhatikan ukuran teks, kontras warna, dan struktur navigasi yang mudah diakses, sehingga dapat digunakan oleh pengguna dengan beragam kebutuhan.

#### 4.1.2.4 Implementasi Halaman Publik

Halaman-halaman publik dirancang untuk memfasilitasi eksplorasi informasi koleksi oleh masyarakat luas, dengan fokus pada kemudahan akses dan pengalaman interaktif.

##### A. Halaman Beranda (*Landing Page*)

Halaman beranda merupakan titik masuk utama yang memberikan pengantar aplikasi secara ringkas dan informatif. Di bagian atas halaman terdapat navbar yang menyediakan tautan ke halaman koleksi, Login, dan pilihan bahasa. Halaman ini juga menampilkan tombol untuk membuka drawer interaktif yang menampilkan jumlah total pengunjung (*visitor*) secara *real-time*. Drawer muncul secara animatif dari sisi kiri layar dan memanfaatkan data dari *backend*. Tampilan halaman ini bersifat bersih, terpusat, dan responsif di berbagai perangkat, serta menjadi penunjuk awal nilai edukatif dari sistem digitalisasi Museum Lampung. Tampilan halaman beranda dapat dilihat pada Gambar 4.14.



### Galeri Koleksi



**Kain Inuh**  
Kain tapis yang terdapat pada masyarakat Pesisir atau Salabatin disebut dengan kain Inuh. Kain Inuh u...



**Gamolan Kulintang Pekhing**  
Gamolan Kulintang Pekhing mulai dikenal di daerah Lampung sekitar abad ke-4 hingga ke-5 Masehi. Gamo...



**Prasasti Bungkok**  
Prasasti Bungkok ditemukan pada tahun 1985 di Desa Bungkok, Kecamatan Jabung, Kabupaten Lampung Timu...

### Kegiatan Museum



**Perpustakaan Nasional Republik Indonesia Menyelamatkan Naskah Kuno di Lampung**  
Perpustakaan Nasional Republik Indonesia melalui Pusat Preservasi dan Alin Media Bahan Perpustakaan sukses

Lihat Semua Kegiatan

### Info Pengunjung



| Jam Buka |               |
|----------|---------------|
| Senin    | 08.00 – 14.00 |
| Selasa   | 08.00 – 14.00 |
| Rabu     | 08.00 – 14.00 |
| Kamis    | 08.00 – 14.00 |
| Jumat    | 08.00 – 10.30 |
| Sabtu    | 08.00 – 14.00 |
| Minggu   | 08.00 – 14.00 |

| Harga Tiket |          |
|-------------|----------|
| Dewasa:     | Rp 5.000 |
| Anak-Anak:  | Rp 1.000 |
| Mahasiswa:  | Rp 2.000 |

### Lokasi



**Alamat**

Jl. ZA. Pagar Alam No.64, Gedung Meneng, Rajabasa, Bandar Lampung, Lampung 35141

**Telepon**

(0721) 706264

**Instagram**

@museumlampung\_ruwajurai

© 2025 Museum Lampung. Seluruh hak cipta dilindungi.

Gambar 4.14 Halaman Beranda

## B. Halaman Koleksi

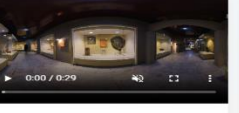
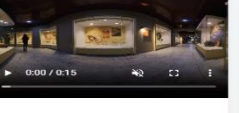
Halaman koleksi berfungsi sebagai pusat eksplorasi data koleksi digital Museum Lampung yang dapat diakses oleh semua pengunjung. Koleksi ditampilkan dalam bentuk kartu (*card*) yang menampilkan gambar atau thumbnail video, nama koleksi, dan deskripsi singkat. Desain kartu bersifat responsif, mengikuti ukuran layar pengguna. Fitur tambahan seperti pencarian dinamis berdasarkan kata kunci dan *filter* berdasarkan tahun atau kategori koleksi tersedia di bagian atas halaman. Hal ini membantu pengguna untuk menelusuri koleksi dengan cepat dan tepat sasaran. Selain itu, setiap kartu dilengkapi dengan tombol “Lihat Detail” yang mengarahkan pengguna ke halaman informasi lengkap dari koleksi yang dipilih. Tampilan halaman koleksi dapat dilihat pada Gambar 4.15.


**Museum Lampung**

[Beranda](#)
[Koleksi](#)
[Tentang](#)
[Masuk](#)



## Daftar Koleksi

|                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p><b>Radin Intan</b></p> <p>1. Pedang Punggawa Radin Intan Pada permukaan bilah pedang terdapat tulisan arab yang...</p> <input type="button" value="Lihat Detail"/>                     |  <p><b>VOC Senjata Pistol</b></p> <p>Seri lain dari pistol VOC jenis yang dulu didapat lagi jenis pistol dengan moncong yang berbeda...</p> <input type="button" value="Lihat Detail"/>                                |  <p><b>Senjata Lampung</b></p> <p>1. Payan Kejang (Tambak panjang) dan Taming (Tameng) Merupakan senjata tradisional...</p> <input type="button" value="Lihat Detail"/>                                                  |  <p><b>Naskah Buku Kulit Kayu</b></p> <p>Naskah kulit kayu pada umumnya beraksara lampung dan menggunakan bahasa lampung...</p> <input type="button" value="Lihat Detail"/>                                       |
|  <p><b>Hindu</b></p> <p>1. Arca Dhyani Bodhisatwa Awalakiteswara Arca bergaya Majapahit Abad XV M. Arca... sebagai arca perwujudan dari...</p> <input type="button" value="Lihat Detail"/> |  <p><b>Miniatur Begawi Cakak Pepadun</b></p> <p>1. Sessat Sessat/Balai Adat, tempat permusyawaratan adat para perwatin (Majelis Pemuka...</p> <input type="button" value="Lihat Detail"/>                              |  <p><b>Homo Sapiens</b></p> <p>Homo Sapiens hidup pada Masa Beracak Tanam (Neolitik) yang mulai dikenal sejak 10.000-... 4.000/3.000 tahun lalu. Benda...</p> <input type="button" value="Lihat Detail"/>                |  <p><b>Kain Tapis</b></p> <p>Kain tapis adalah tenunan yang berbentuk kain sarung, dipakai oleh wanita suku Lampung pada... saat upacara adat, terbuat dari...</p> <input type="button" value="Lihat Detail"/>    |
|  <p><b>Puade</b></p> <p>Puade adalah tempat duduk khusus bagi mempelai (pengantin) yang digunakan sa... prosesi pernikahan adat...</p> <input type="button" value="Lihat Detail"/>        |  <p><b>Perahu Lesung dan Perahu Kajang</b></p> <p>1. Perahu Lesung Perahu berukuran panjang 7,66 m ditemukan di Desa Terbanggi...</p> <input type="button" value="Lihat Detail"/>                                     |  <p><b>Kulintang atau Talo Balak</b></p> <p>Merupakan alat tabuh/bundian untuk mengiringi acara-acara adat baik dirumah maupun di...</p> <input type="button" value="Lihat Detail"/>                                    |  <p><b>Kain Inuh</b></p> <p>Kain tapis yang terdapat pada masyarakat Pesisir atau Saibatin disebut dengan kain Inuh. Kain... Inuh umumnya dibuat dengan...</p> <input type="button" value="Lihat Detail"/>       |
|  <p><b>Gamolan Kulintang Pekhing</b></p> <p>Gamolan Kulintang Pekhing mulai dikenal di daerah Lampung sekitar abad ke-4 hingga ke-5...</p> <input type="button" value="Lihat Detail"/>   |  <p><b>Prasasti Bungkok</b></p> <p>Prasasti Bungkok ditemukan pada tahun 1985 di Desa Bungkok, Kecamatan Jabung, Kabupaten... Lampung Timur. Prasasti ini terdiri...</p> <input type="button" value="Lihat Detail"/> |  <p><b>Arca Megalitik Folonesia</b></p> <p>Arca ditemukan tahun 1963 di Situs Pujung Ranario, Lampung Timur. Pada bagian pinggang... belakang terselip sebuah keris...</p> <input type="button" value="Lihat Detail"/> |  <p><b>Koleksi Filologika</b></p> <p>Naskah kuno koleksi Museum Lampung terdiri dari kulit kayu, daun lontar, kayu, tanduk dan... kertas daluang. Huruf yang...</p> <input type="button" value="Lihat Detail"/> |
|  <p><b>Keramik</b></p> <p>Keramik merupakan benda yang terbuat dari tanah liat yang dibakar pada suhu tinggi...</p> <input type="button" value="Lihat Detail"/>                          |  <p><b>Batuan dan Mineral</b></p> <p>Batuan merupakan material padat penyusun kerak bumi yang terbentuk secara alami dan... berdasarkan proses terbentuknya...</p> <input type="button" value="Lihat Detail"/>       |  <p><b>Lumpang Batu</b></p> <p>Lumpang batu adalah alat berbentuk lesung yang berasal dari masa tradisi megalitik muda... pada periode Neolitikum...</p> <input type="button" value="Lihat Detail"/>                   |  <p><b>Koleksi Uang Kertas dan Koin</b></p> <p>Dalam ruang pameran Museum Negeri Lampung "Ruwa Jura", terdapat koleksi uang logam dan...</p> <input type="button" value="Lihat Detail"/>                        |



**Alamat**

Jl. ZA. Pagar Alam No.64, Gedung Meneng, Rajabasa, Bandar Lampung, Lampung 35141



**Telepon**

(0721) 705264



**Instagram**

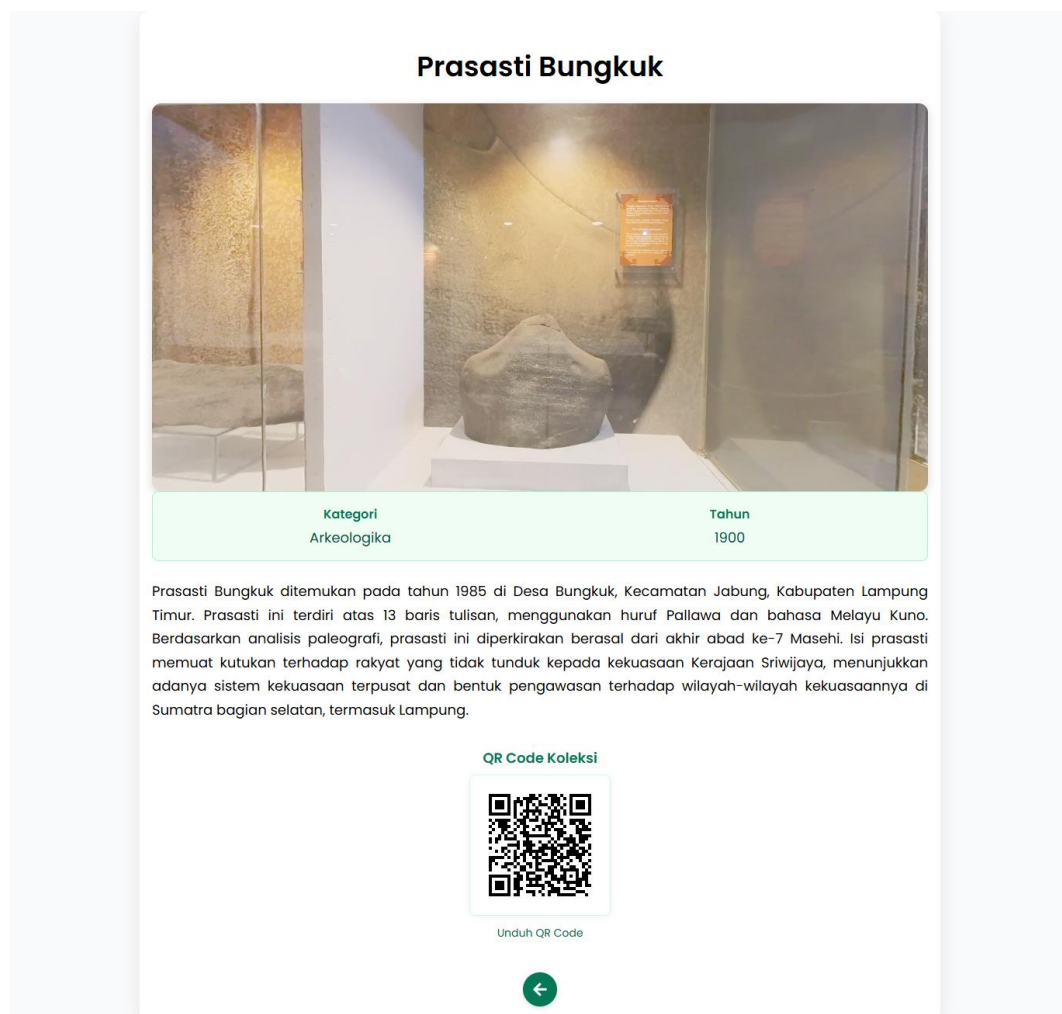
@museumlampung\_ruwajurai

© 2025 Museum Lampung. Seluruh hak cipta dilindungi.

Gambar 4. 15 Halaman Daftar Koleksi

### C. Halaman Detail Koleksi

Halaman detail koleksi menyajikan informasi lengkap dan komprehensif untuk koleksi tertentu. Ini mencakup nama koleksi, tahun akuisisi, kategori, deskripsi panjang, gambar utama, dan video 360° jika tersedia. Gambar ditampilkan dalam ukuran besar dengan kemampuan untuk diperbesar (*zoom*), sementara video 360° dapat diputar dan dinavigasi secara interaktif. Halaman ini juga kompatibel dengan pemindaian *QR Code*, yang memungkinkan pengunjung diarahkan langsung ke informasi detail koleksi dari objek fisik di museum. Tampilan halaman detail koleksi dapat dilihat pada Gambar 4.16.

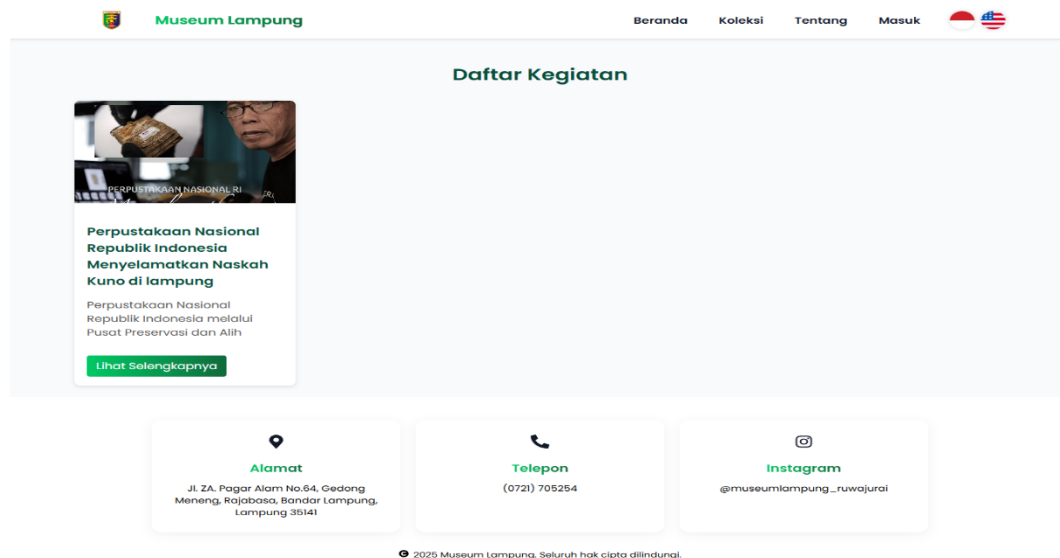


Gambar 4.16 Halaman Detail Koleksi

### D. Halaman Kegiatan

Halaman daftar kegiatan berfungsi sebagai pusat informasi mengenai kegiatan atau acara yang diselenggarakan oleh Museum Lampung, dan dapat diakses oleh semua

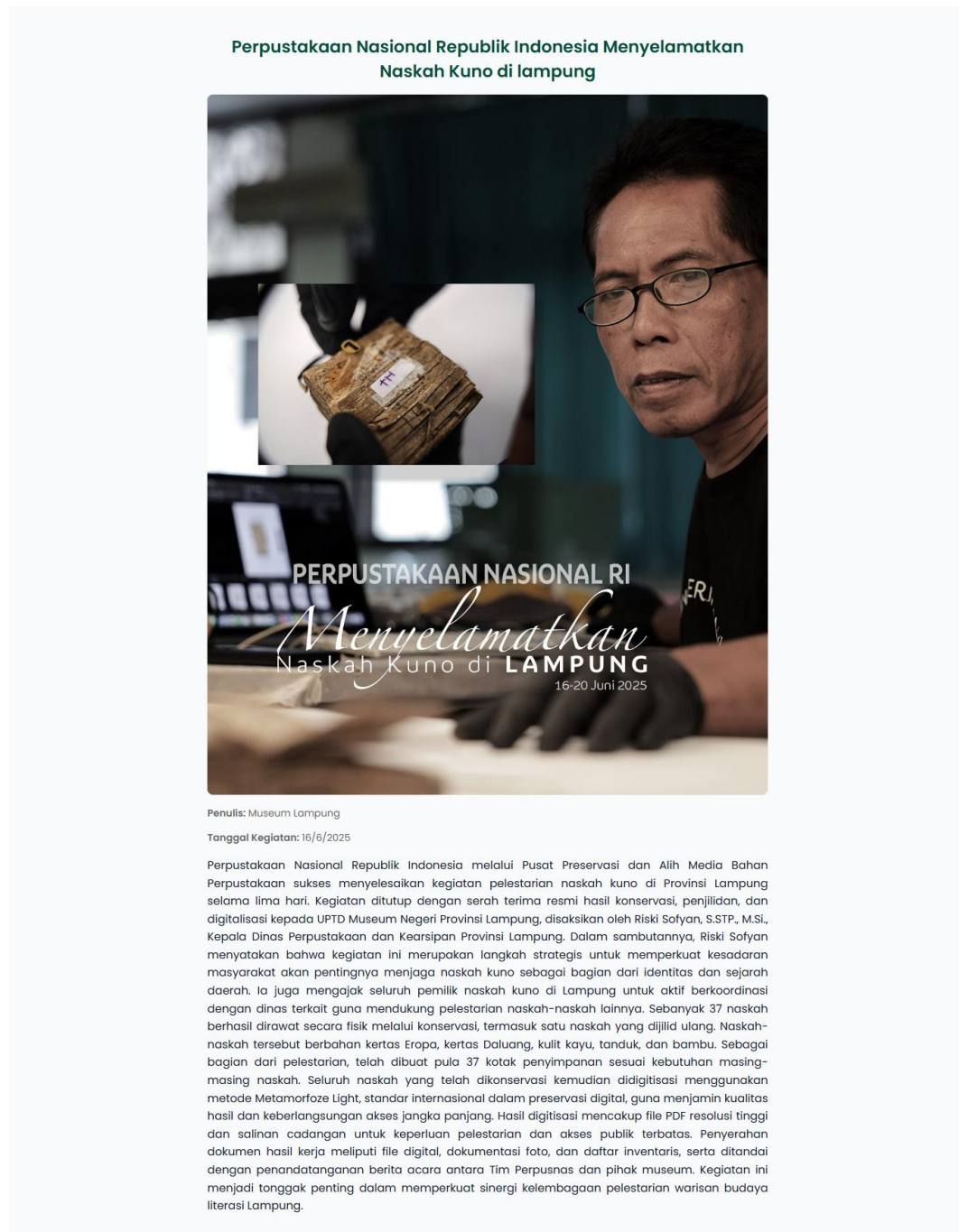
pengunjung. Kegiatan ditampilkan dalam bentuk kartu (*card*) yang berisi gambar atau video, judul kegiatan, nama penulis, tanggal pelaksanaan, dan deskripsi singkat. Desain kartu bersifat responsif, menyesuaikan diri dengan ukuran layar pengguna. Setiap kartu dilengkapi dengan tombol “Lihat Selengkapnya” yang mengarahkan pengguna ke halaman informasi detail kegiatan yang dipilih. Tampilan halaman ini dapat dilihat pada Gambar 4.17.



Gamabr 4.17 Halaman Daftar Kegiatan

#### E. Halaman Detail Kegiatan

Halaman detail kegiatan menyajikan informasi lengkap dan komprehensif untuk kegiatan tertentu yang diselenggarakan oleh Museum Lampung. Ini mencakup judul kegiatan, tanggal pelaksanaan, nama penulis atau penyelenggara, lokasi, deskripsi lengkap, serta media pendukung berupa gambar atau video. Halaman ini juga mendukung pemindaian *QR Code*, yang memungkinkan pengguna untuk langsung mengakses informasi detail kegiatan dari materi promosi fisik. Tampilan halaman ini dapat dilihat pada Gambar 4.18.

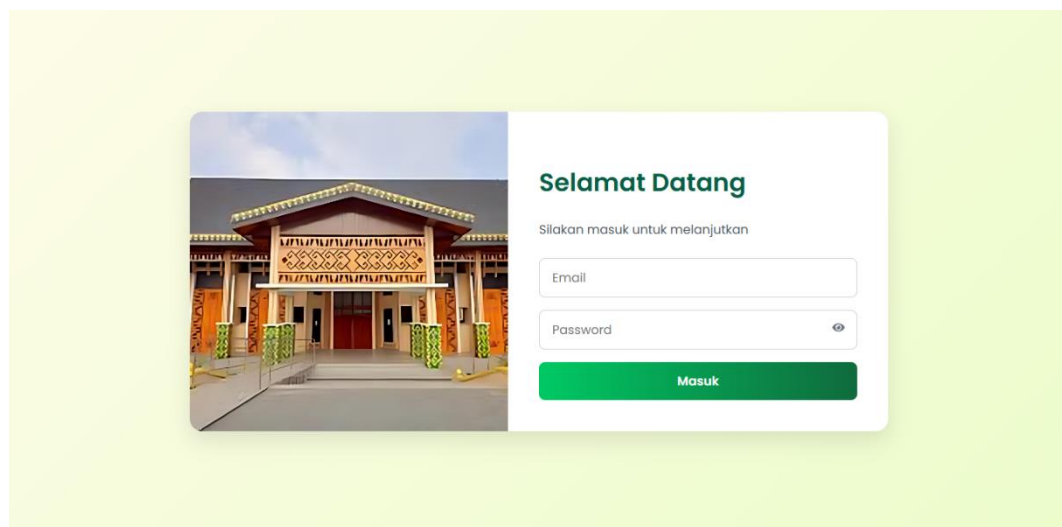


Gambar 4. 18 Halaman Detail Kegiatan

#### F. Halaman Login

Halaman login berperan sebagai gerbang autentikasi untuk pengguna yang memiliki peran sebagai admin atau superadmin. Di halaman ini, pengguna diminta untuk memasukkan kredensial berupa email dan kata sandi. Setelah data dikirimkan, sistem akan melakukan verifikasi dan menyesuaikan tampilan dashboard berdasarkan peran pengguna. Jika data sesuai dan berhasil diverifikasi,

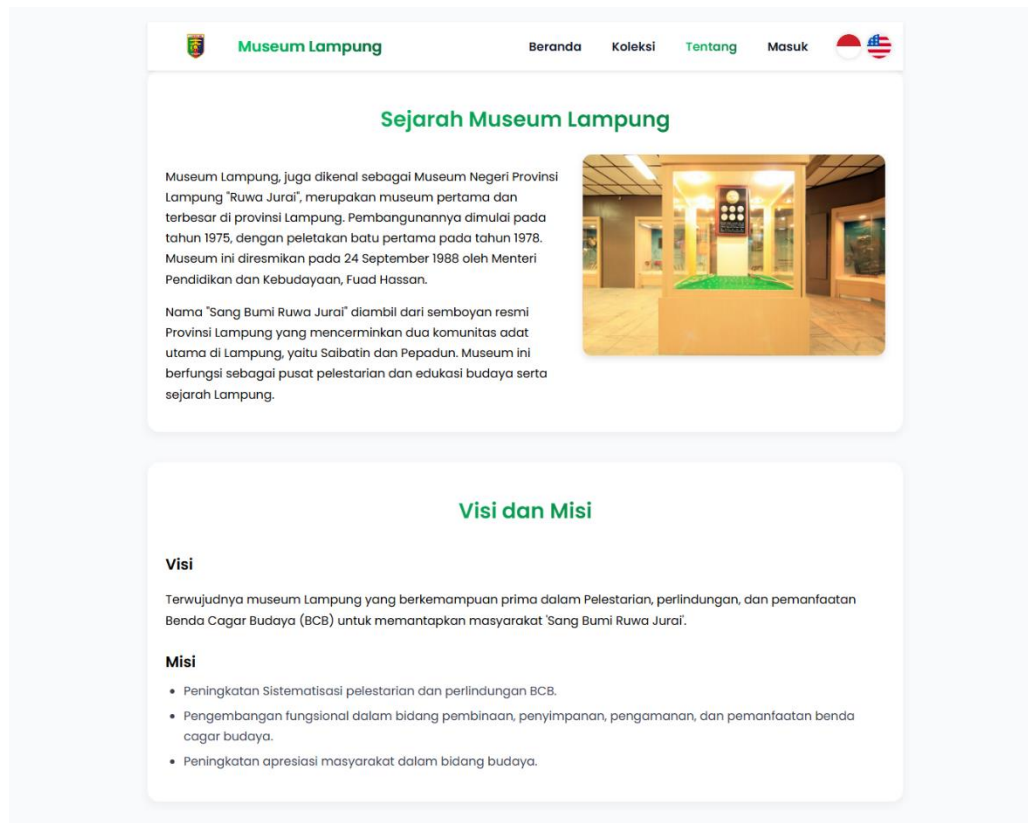
pengguna diarahkan ke halaman admin *dashboard* atau superadmin *dashboard*. Sebaliknya, jika terjadi kesalahan, sistem akan menampilkan pesan bahwa email atau kata sandi tidak valid. Antarmuka halaman login dirancang minimalis dan terfokus, dengan *form* yang berada di tengah layar dan memiliki validasi *real-time* terhadap kesalahan *input*. Hal ini bertujuan untuk meningkatkan kenyamanan dan efisiensi saat pengguna melakukan proses autentikasi. Tampilan halaman ini dapat dilihat pada Gambar 4.19.



Gambar 4.19 Halaman Login

#### G. Halaman Tentang

Halaman "Tentang" memberikan informasi mendalam mengenai Museum Lampung, termasuk sejarah, visi dan misi, sarana dan prasarana, serta struktur organisasi.<sup>1</sup> Bagian sejarah menguraikan pembangunan dan peresmian museum, serta makna di balik nama "Sang Bumi Ruwa Jurai". Visi dan misi museum dijelaskan untuk memberikan pemahaman tentang tujuan pelestarian dan edukasi budaya. Daftar sarana dan prasarana memberikan gambaran fasilitas yang tersedia. Struktur organisasi menguraikan peran kepala museum dan daftar pimpinan sejak peresmian, serta unit-unit fungsional lainnya. Halaman ini juga menyertakan denah museum untuk membantu pengunjung memahami tata letak fisik. Seluruh konten pada halaman ini mendukung fitur multibahasa, yang memungkinkan informasi disajikan dalam Bahasa Indonesia atau Bahasa Inggris. Tampilan halaman ini dapat dilihat pada Gambar 4.20.



Gambar 4.20 Halaman Tentang

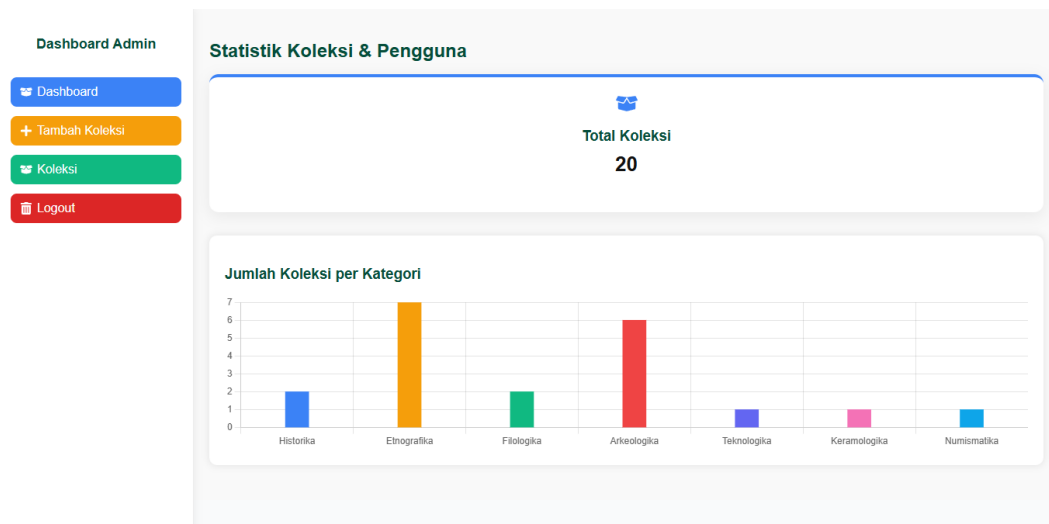
#### 4.1.2.5 Implementasi Dashboard Admin Koleksi dan Admin Kegiatan

*Dashboard* ini dirancang untuk memberikan admin kontrol yang efisien atas data yang menjadi tanggung jawabnya, dengan tampilan yang ringkas dan fungsionalitas yang terpusat.

##### A. Halaman *Dashboard* Admin Koleksi:

Admin koleksi memiliki akses eksklusif ke halaman *dashboard* yang menyajikan informasi ringkas dan fitur pengelolaan koleksi museum. Di tampilan awal *dashboard*, admin disuguhkan statistik jumlah total koleksi dan pengguna melalui tampilan kartu, serta grafik batang yang memperlihatkan distribusi koleksi berdasarkan kategori. Halaman ini dirancang agar ringkas dan mudah dipantau. Admin dapat mengakses *form* tambah koleksi untuk mendaftarkan koleksi baru, yang mencakup informasi nama, deskripsi, tahun, kategori, gambar, dan video 360°. Setelah pengisian selesai, admin dapat menyimpan data dan sistem akan mengunggah *file* media ke Cloudinary serta menghasilkan *QR Code*. Admin juga

dapat mengakses daftar koleksi yang ditampilkan dalam bentuk tabel, lengkap dengan fitur pencarian, serta tombol aksi “edit” dan “hapus”. Selain itu, admin memiliki akses ke halaman edit koleksi, di mana data koleksi yang telah disimpan ditampilkan kembali dalam *form* isian untuk diperbarui. Admin juga dapat membuka halaman daftar pengguna yang menampilkan seluruh akun yang ada di sistem, sebagai bagian dari fungsi monitoring dan pengendalian akses.



Gambar 4.21 Halaman Statistik *Dashboard* Admin

The screenshot shows the 'Tambah Koleksi' form within the 'Dashboard Admin' sidebar. The form includes the following fields and controls:

- Nama Koleksi:** A text input field.
- Deskripsi:** A text area for detailed description.
- Tahun:** A text input field for the year.
- Pilih Kategori:** A dropdown menu for selecting a category.
- Gambar:** A file upload section with a 'Choose File' button and the text 'No file chosen'.
- Video 360:** A file upload section with a 'Choose File' button and the text 'No file chosen'.
- Simpan:** A large blue button at the bottom to save the collection.

Gambar 4.22 Halaman Tambah Koleksi

Dashboard Admin

Dashboard
Tambah Koleksi
Koleksi
Logout

Daftar Koleksi

Cari koleksi...

| Nama               | Deskripsi                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Tahun | Kategori    | Aksi                                                                                                                                                                    |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Radin Intan        | 1. Pedang Punggawa Radin Intan Pada permukaan bilah pedang terdapat tulisan arab yang berbunyi "Innama Amruhu Idza Arrodallahu Sya'an Ayaquula Lahu Kun Fayakun". Pedang ini juga menjadi bukti sejarah kegigihan Radin Intan II bersama pasukannya berjuang melawan penjajah Belanda yang sangat berkeinginan untuk menguasai Lampung. 2. Meriam Lela/Meriam Benteng Indonesia mengenal meriam sejak kedatangan bangsa Portugis pada abad ke 16. Meriam merupakan salah satu jenis senjata dengan berbagai bentuk dan ukuran. Fungsinya menembak jarak jauh. 3. Topeng Topping sebagai simbol prajurit rahasia Radin Intan II. Topping dikenakan dalam tari topping klasik dalam bentuk drama tari, digunakan untuk menggambarkan kepahlawanan di daerah Lampung selatan. | 1900  | Historika   |   |
| VOC Senjata Pistol | Seri lain dari pistol VOC jenis yang dulu didapat lagi jenis pistol dengan moncong yang berbeda (mekar/mekrok/terompet). Dengan barang kawat pembersih sisa mesiu yang tersedia di sisi bawah depan ujung pistol. Relik Sejarah (Historical Relics) adalah peninggalan masa lampau berupa benda-benda yang pernah dibuat dan dipergunakan manusia sebagai bukti kehidupan pada masa lalu dan merupakan peninggalan masa kolonial.                                                                                                                                                                                                                                                                                                                                          | 1900  | Historika   |   |
| Senjata Lampung    | 1. Payan Kejang (Tombak panjang) dan Taming (Tameng) Merupakan senjata tradisional. Pada masyarakat Lampung digunakan juga sebagai perlengkapan upacara adat. 2. Punduk, Tekhapang (Keris) Merupakan benda pusaka pada masyarakat lampung digunakan sebagai perlengkapan upacara adat. 3. Panderang (Pedang) merupakan senjata tradisional. 4. Panderang (pedang Type Lampung) Merupakan senjata tradisional, juga digunakan untuk perlengkapan pencak silat dalam acara Begawi Adat Lampung.                                                                                                                                                                                                                                                                              | 1950  | Etnografika |   |

Gambar 4.23 Halaman Daftar Koleksi

Dashboard Admin

Dashboard
Tambah Koleksi
Koleksi
Logout

Perbarui Koleksi

Radin Intan

1. Pedang Punggawa Radin Intan Pada permukaan bilah pedang terdapat tulisan arab yang berbunyi "Innama Amruhu Idza Arrodallahu Sya'an Ayaquula Lahu Kun Fayakun". Pedang ini juga menjadi bukti sejarah kegigihan Radin Intan II bersama pasukannya berjuang melawan penjajah Belanda yang sangat berkeinginan untuk menguasai Lampung. 2. Meriam Lela/Meriam Benteng Indonesia mengenal meriam sejak kedatangan bangsa Portugis pada abad ke 16. Meriam merupakan salah satu jenis senjata dengan berbagai bentuk dan ukuran. Fungsinya menembak jarak jauh. 3. Topeng Topping sebagai simbol prajurit rahasia Radin Intan II. Topping dikenakan dalam tari topping klasik dalam bentuk drama tari, digunakan untuk menggambarkan kepahlawanan di daerah Lampung selatan.

1900

Historika

Gambar:

Choose File

No file chosen

Video 360:

Choose File

No file chosen

Update

Gambar 4.24 Halaman Edit Koleksi

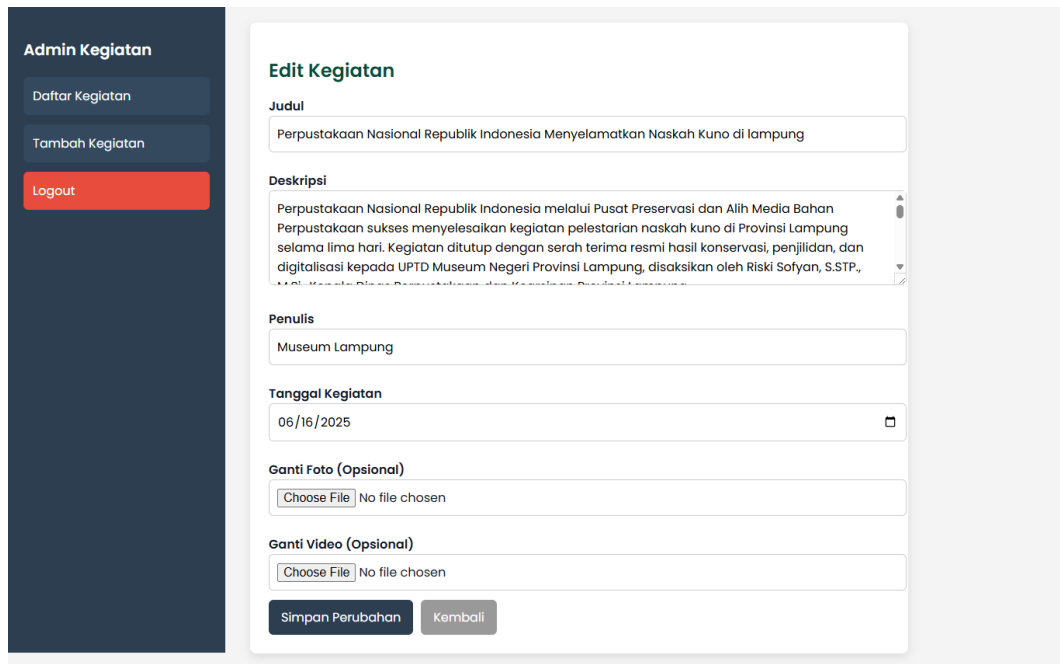
## B. Halaman *Dashboard* Admin Kegiatan:

Admin Kegiatan memiliki peran khusus untuk mengelola seluruh informasi terkait kegiatan museum. Halaman *dashboard* ini didedikasikan untuk manajemen kegiatan, dan memiliki fitur utama Halaman Tambah Kegiatan, yang memungkinkan admin untuk mendaftarkan kegiatan baru melalui *form* isian yang mencakup nama, deskripsi, tanggal, penulis, dan gambar atau video kegiatan. Selain itu, terdapat Halaman Daftar Kegiatan yang menampilkan seluruh kegiatan dalam format tabel, lengkap dengan fitur tombol "edit" dan "hapus". Admin juga dapat

memperbarui informasi kegiatan yang sudah ada melalui Halaman Edit Kegiatan, di mana data lama ditampilkan kembali dalam *form* isian untuk diperbarui.

Gambar 4.25 Halaman Tambah Kegiatan

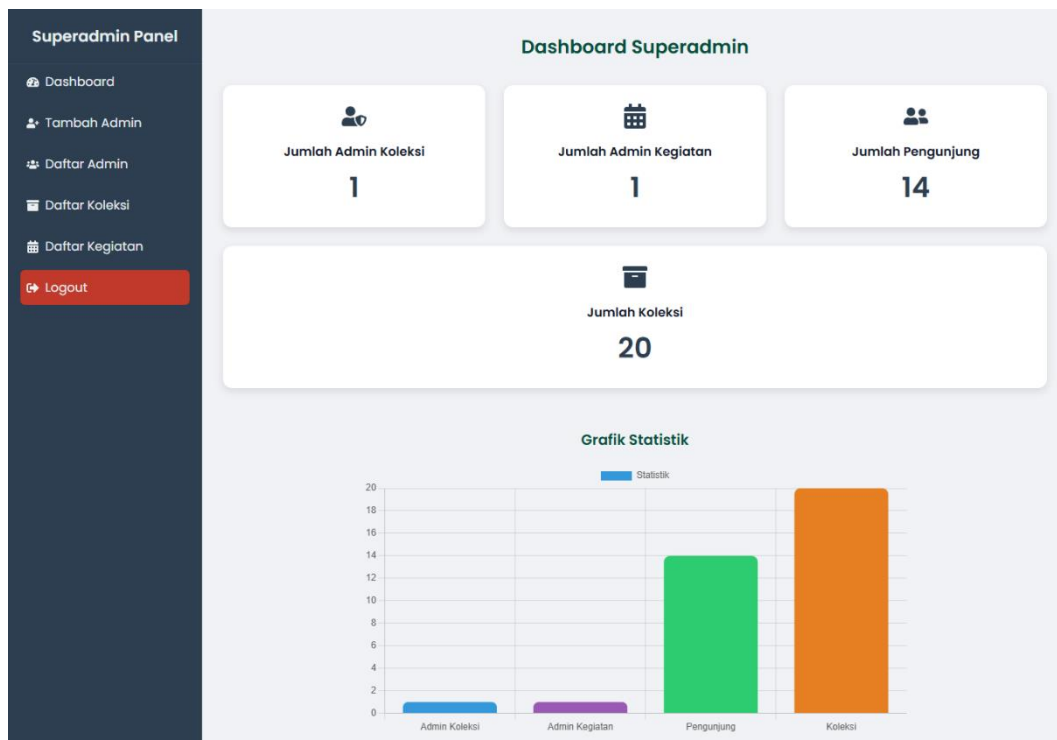
Gambar 4.26 Halaman Daftar Kegiatan



Gambar 4.27 Halaman Edit Kegiatan

#### 4.1.2.6 Implementasi *Dashboard Superadmin*

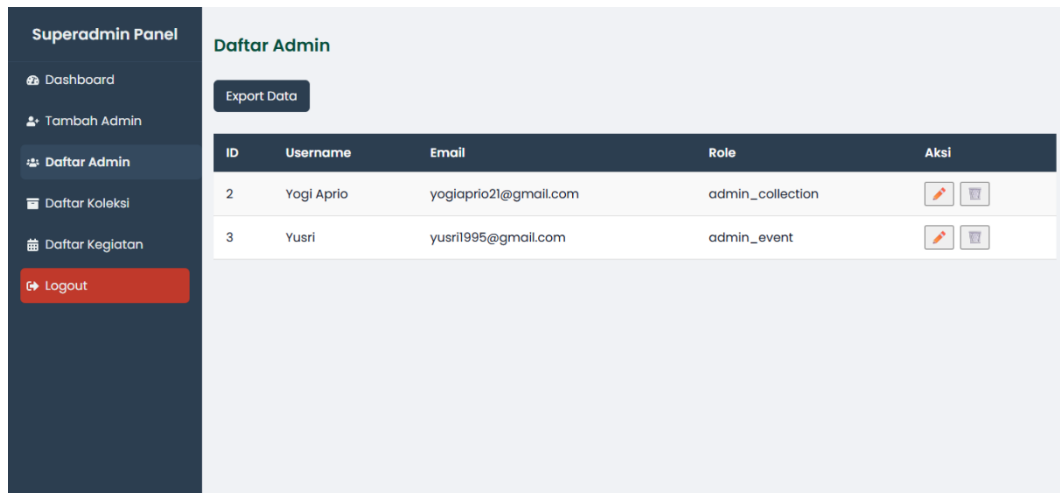
*Dashboard* ini memberikan kontrol tertinggi kepada *superadmin* untuk mengelola akun admin dan memantau data sistem. Di halaman ini, *superadmin* dapat menambahkan admin baru (baik admin koleksi maupun admin kegiatan) melalui *form input*, mengedit data admin yang sudah ada, serta menghapus akun jika diperlukan. Selain itu, *superadmin* juga dapat memantau seluruh koleksi dan kegiatan yang terdaftar dalam sistem melalui halaman daftar khusus. Berbeda dari admin koleksi dan admin kegiatan, *superadmin* tidak memiliki kontrol langsung untuk mengedit koleksi atau kegiatan, namun dapat memantau dan mengekspor data koleksi ke format CSV sebagai bentuk dokumentasi atau audit internal.



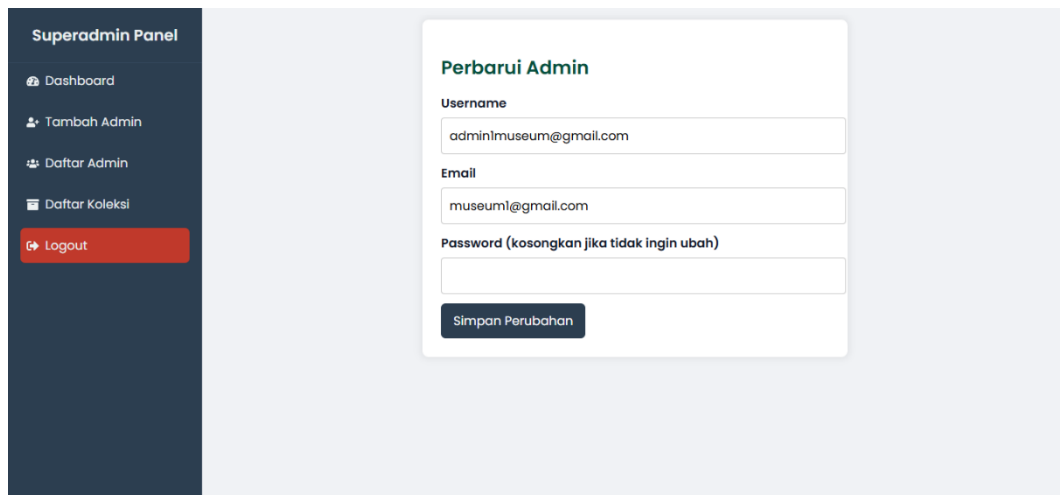
Gambar 4.28 Halaman Statistik Dashboard Superadmin

The screenshot shows the 'Tambah Admin' form within the 'Superadmin Panel'. The sidebar menu is the same as in the previous image, but the 'Tambah Admin' option is highlighted. The form itself has a title 'Tambah Admin' and three input fields: 'Username' (placeholder: 'Masukkan username'), 'Email' (placeholder: 'Masukkan email'), and 'Password' (placeholder: 'Masukkan password'). A dark blue button with a white user icon and the text 'Tambah Admin' is at the bottom of the form.

Gambar 4.29 Halaman Tambah Admin



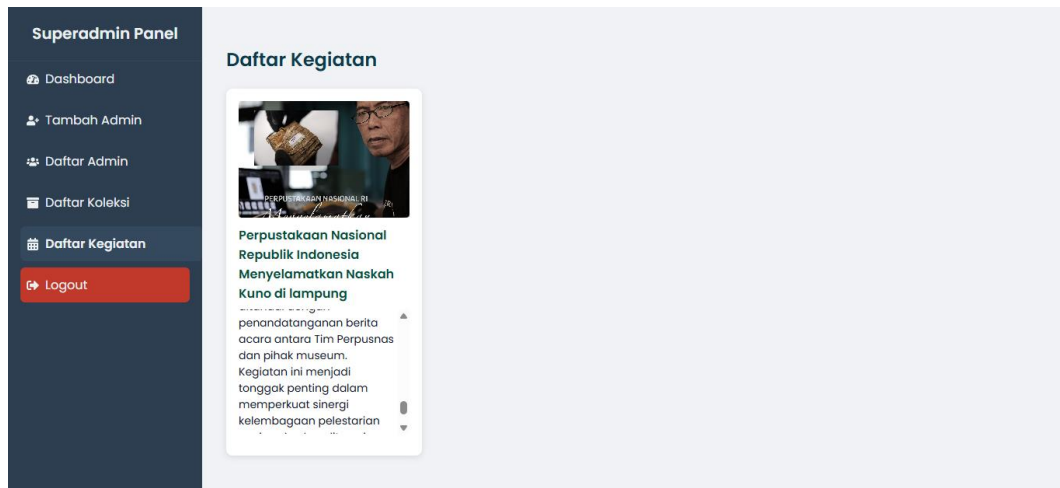
Gambar 4.30 Halaman Daftar Admin



Gambar 4.31 Halaman Edit Admin



Gambar 4.32 Halaman Daftar Koleksi (Superadmin)



Gambar 4.33 Halaman Daftar Kegiatan (Superadmin)

#### 4.1.2.7 Implementasi Viewer Video 360 Derajat

Komponen `ThreeJS360Viewer` ini memungkinkan pemutaran video 360° menggunakan pustaka *Three.js* dan *OrbitControls*, memberikan pengalaman imersif bagi pengguna saat melihat koleksi yang didokumentasikan dengan video 360°. *Three.js* digunakan untuk membuat lingkungan 3D di mana video dipetakan ke permukaan bola, menciptakan ilusi pengguna berada di tengah adegan. *OrbitControls* memungkinkan navigasi interaktif, seperti memutar pandangan, yang memperkaya eksplorasi visual koleksi.

Implementasi *viewer* video 360 derajat secara signifikan meningkatkan tingkat imersi dan pengalaman edukatif bagi pengunjung. Berbeda dengan gambar statis atau video linear, video 360 derajat memungkinkan pengguna untuk "hadir" di lingkungan koleksi, menjelajahi detail dari berbagai sudut pandang seolah-olah mereka berada di museum.

## 4.2 Pengujian Sistem

Pengujian sistem dilakukan untuk memverifikasi bahwa setiap komponen dan fungsionalitas sistem berjalan sesuai dengan spesifikasi yang telah ditetapkan, serta untuk mengidentifikasi dan memperbaiki *bug* atau anomali. Pengujian meliputi *black-box testing* untuk fungsionalitas *frontend*, *Postman testing* untuk *API backend*, dan *unit testing* dengan Jest untuk logika *backend*.

#### 4.2.1 Pengujian Fungsionalitas (*Black-Box Testing*)

Pengujian *black-box* dilakukan dari perspektif pengguna akhir tanpa melihat struktur kode internal. Tujuannya adalah untuk memverifikasi bahwa semua fitur yang terlihat oleh pengguna berfungsi dengan benar.

##### a. Metodologi Pengujian

Pengujian ini mencakup skenario penggunaan utama seperti proses login untuk berbagai peran (admin koleksi, admin kegiatan, superadmin), navigasi antar halaman, interaksi pengguna umum dalam menelusuri dan mencari koleksi, serta operasi *CRUD* oleh admin pada koleksi dan kegiatan. Pengujian ini memastikan sistem memenuhi kebutuhan fungsional pengguna dan memberikan pengalaman yang diharapkan.

##### b. Hasil Pengujian Fungsionalitas (*Black-Box Testing*)

Tabel 4.1 menyajikan hasil pengujian fungsionalitas sistem menggunakan metode *black-box testing*. Pengujian ini berfokus pada verifikasi perilaku sistem dari perspektif pengguna akhir, memastikan bahwa semua fitur yang dirancang berfungsi dengan benar dan sesuai dengan spesifikasi.

Hasil yang disajikan menunjukkan keberhasilan implementasi berbagai skenario kritis, mulai dari proses autentikasi hingga manajemen koleksi dan kegiatan, serta fitur-fitur publik seperti tampilan detail koleksi dan dukungan multibahasa.

Tabel 4.1. Hasil Pengujian Fungsionalitas Sistem (*Black-Box Testing*)

| No. | Modul/Fitur yang Diuji | Skenario Pengujian                          | Hasil yang Diharapkan                                | Hasil Pengujian |
|-----|------------------------|---------------------------------------------|------------------------------------------------------|-----------------|
| 1.  | Autentikasi Pengguna   | Login Admin Koleksi dengan kredensial valid | Berhasil login, diarahkan ke dashboard admin koleksi | Berhasil        |
| 2.  |                        | Login Admin Koleksi dengan password salah   | Pesan error "Email atau password salah" ditampilkan  | Berhasil        |

Tabel 4.1 Hasil Pengujian Fungsionalitas Sistem

| No. | Modul/Fitur yang Diuji            | Skenario Pengujian                                      | Hasil yang Diharapkan                                                                | Hasil Pengujian |
|-----|-----------------------------------|---------------------------------------------------------|--------------------------------------------------------------------------------------|-----------------|
| 3.  | Manajemen Koleksi (Admin Koleksi) | Menambah koleksi baru dengan gambar dan video 360°      | Koleksi berhasil ditambahkan, media terunggah, <i>QR Code</i> tergenerasi            | Berhasil        |
| 4.  |                                   | Mengedit koleksi yang sudah ada dengan mengganti gambar | Koleksi berhasil diperbarui, gambar lama terhapus, gambar baru terunggah             | Berhasil        |
| 5.  |                                   | Mencari koleksi berdasarkan nama di halaman publik      | Koleksi ditampilkan, jika tidak ada, pesan "Koleksi tidak ditemukan"                 | Berhasil        |
| 8.  | Manajemen Admin (Superadmin)      | Menghapus akun admin                                    | Akun admin berhasil dihapus, tidak dapat login lagi                                  | Berhasil        |
| 9.  | Halaman Publik                    | Melihat detail koleksi dengan video 360°                | Video 360° dapat diputar dan dinavigasi secara interaktif                            | Berhasil        |
| 10. |                                   | Mengganti bahasa (ID ke EN)                             | Teks antarmuka dan konten dinamis (nama/deskripsi koleksi) berubah ke Bahasa Inggris | Berhasil        |

#### 4.2.2 Pengujian API (*Postman Testing*)

Pengujian *API* dilakukan untuk memverifikasi fungsionalitas *endpoint backend*, termasuk operasi *CRUD*, autentikasi, dan otorisasi.

a. Metodologi Pengujian

Postman digunakan untuk mengirim permintaan *HTTP* (*GET*, *POST*, *PUT*, *DELETE*) ke *endpoint API backend*. Pengujian ini memastikan bahwa *backend* merespons dengan benar terhadap berbagai skenario, termasuk *input* valid, *input* tidak valid, dan akses tanpa otorisasi. Pengujian ini memastikan integritas komunikasi antara *frontend* dan *backend*, serta keandalan logika bisnis di *backend*.

b. Hasil Pengujian *API* (*Postman Testing*)

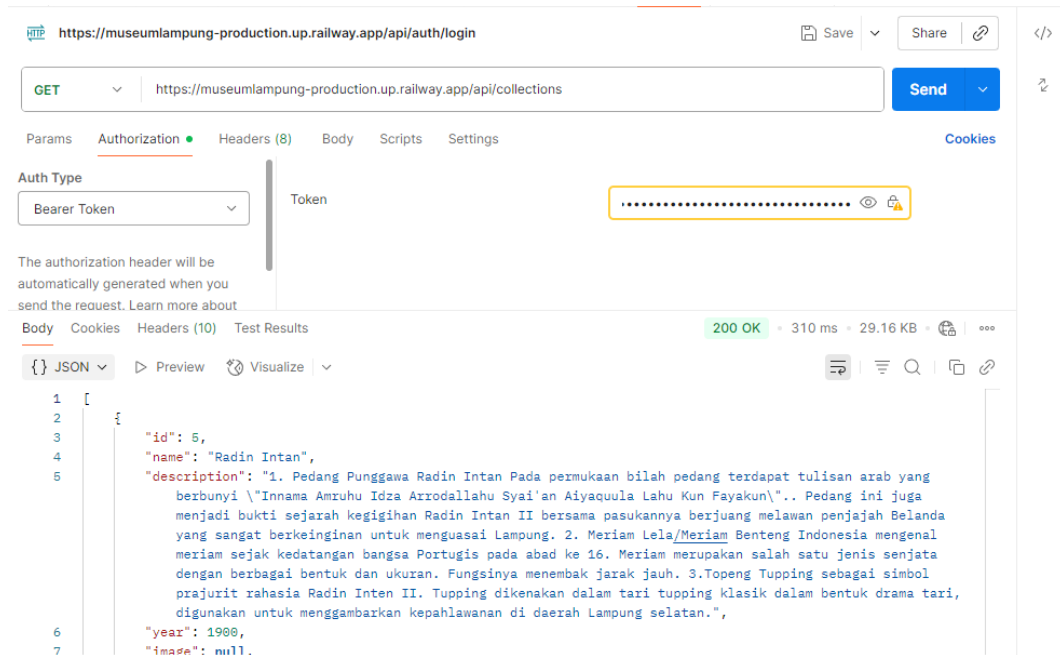
Tabel 4.2 menyajikan hasil pengujian *API* menggunakan Postman. Pengujian ini berfokus pada validasi *endpoint backend* dan interaksinya dengan basis data, serta mekanisme autentikasi dan otorisasi. Hasil yang ditunjukkan mengkonfirmasi bahwa setiap *endpoint API* berfungsi sesuai dengan desain, memberikan respons yang diharapkan untuk skenario sukses maupun gagal, termasuk penanganan token JWT dan rate limiting. Tabel ini merupakan bukti penting dari keandalan dan keamanan lapisan *backend* sistem.

Tabel 4.2. Hasil Pengujian *API* (*Postman Testing*)

| No. | <i>Endpoint API</i>     | Metode <i>HTTP</i> | Skenario Pengujian                                         | Hasil yang Diharapkan                                      | Hasil Pengujian |
|-----|-------------------------|--------------------|------------------------------------------------------------|------------------------------------------------------------|-----------------|
| 1.  | <i>/API/auth/login</i>  | POST               | Login dengan kredensial valid                              | Status 200 OK, respons berisi token JWT dan role           | Berhasil        |
| 2.  |                         | POST               | Login dengan kredensial tidak valid                        | Status 401 Unauthorized, pesan "Email atau password salah" | Berhasil        |
| 3.  | <i>/API/collections</i> | POST               | Tambah koleksi (sebagai admin) dengan data lengkap & media | Status 201 Created, respons berisi data koleksi baru       | Berhasil        |
| 4.  |                         | POST               | Tambah koleksi (tanpa token/otorisasi)                     | Status 401 Unauthorized / 403 Forbidden                    | Berhasil        |

Tabel 4.2 Hasil Pengujian *API (Postman Testing)*

| No. | Endpoint API         | Metode HTTP | Skenario Pengujian                                      | Hasil yang Diharapkan                                      | Hasil Pengujian |
|-----|----------------------|-------------|---------------------------------------------------------|------------------------------------------------------------|-----------------|
| 5.  | /API/collections/:id | GET         | Ambil detail koleksi berdasarkan ID                     | Status 200 OK, respons berisi detail koleksi & QR Code URL | Berhasil        |
| 6.  | /API/collections/:id | PUT         | Update koleksi (sebagai admin) dengan data & media baru | Status 200 OK, koleksi terupdate di DB & Cloudinary        | Berhasil        |
| 7.  | /API/collections/:id | DELETE      | Hapus koleksi (sebagai admin)                           | Status 200 OK, pesan "Koleksi berhasil dihapus"            | Berhasil        |
| 8.  | /API/users           | POST        | Tambah admin (sebagai superadmin)                       | Status 201 Created, respons berisi data user baru          | Berhasil        |
| 9.  | /API/users           | GET         | Ambil semua user (sebagai superadmin)                   | Status 200 OK, respons berisi daftar user                  | Berhasil        |
| 10. | /API/visitors        | POST        | Catat kunjungan (IP baru)                               | Status 201 Created, pesan "Kunjungan baru tercatat"        | Berhasil        |
| 11. | /API/visitors        | POST        | Catat kunjungan (IP yang sudah tercatat hari ini)       | Status 200 OK, pesan "Kunjungan hari ini sudah tercatat"   | Berhasil        |



Gambar 4.34 Contoh Pengujian *Endpoint GET /api/collections* di Postman

### 4.2.3 Pengujian Unit (*Jest Testing*)

Pengujian unit dilakukan untuk memverifikasi bahwa setiap unit kode (fungsi, *middleware*, *service*) pada sisi *backend* bekerja secara mandiri sesuai dengan fungsinya.

#### a. Metodologi Pengujian:

*Framework Jest* digunakan untuk menguji unit-unit kode secara terisolasi. *Mocking* digunakan untuk mengisolasi unit yang diuji dari dependensi eksternal (misalnya, basis data, layanan *cloud*). Pengujian ini memastikan keandalan dan konsistensi logika aplikasi pada tingkat terendah, meminimalkan kemungkinan terjadinya *regression* selama pengembangan lanjutan.

#### b. Tabel Hasil Pengujian Unit (*Jest Testing*):

Tabel 4.3 menyajikan hasil pengujian unit yang dilakukan menggunakan Jest. Hasil pengujian ini menunjukkan *code coverage* yang komprehensif, di mana setiap *file*, pernyataan (*statement*), percabangan (*branch*), fungsi (*function*), dan baris kode memiliki cakupan 100%.

Tabel 4.3. Hasil Pengujian Unit Sistem (Jest Testing)

| <i>File / Direktori</i> | % Stmts | % Branch | % Funcs | % Lines |
|-------------------------|---------|----------|---------|---------|
| <i>All files</i>        | 100     | 100      | 100     | 100     |
| controllers             | 100     | 100      | 100     | 100     |
| authController.js       | 100     | 100      | 100     | 100     |
| collectionController.js | 100     | 100      | 100     | 100     |
| eventController.js      | 100     | 100      | 100     | 100     |
| userController.js       | 100     | 100      | 100     | 100     |
| visitorController.js    | 100     | 100      | 100     | 100     |
| middlewares             | 100     | 100      | 100     | 100     |
| authMiddleware.js       | 100     | 100      | 100     | 100     |
| rateLimiter.js          | 100     | 100      | 100     | 100     |
| validationMiddleware.js | 100     | 100      | 100     | 100     |
| services                | 100     | 100      | 100     | 100     |
| authService.js          | 100     | 100      | 100     | 100     |
| collectionService.js    | 100     | 100      | 100     | 100     |
| eventService.js         | 100     | 100      | 100     | 100     |
| qrCodeService.js        | 100     | 100      | 100     | 100     |
| userService.js          | 100     | 100      | 100     | 100     |
| visitorService.js       | 100     | 100      | 100     | 100     |
| utils                   | 100     | 100      | 100     | 100     |
| <i>fileUpload.js</i>    | 100     | 100      | 100     | 100     |
| validator.js            | 100     | 100      | 100     | 100     |

Hasil tersebut mengindikasikan bahwa seluruh unit kode dalam sistem *backend*

telah diuji secara menyeluruh dan tidak terdapat bagian kode yang terlewat atau belum diuji. Dengan pengujian ini, sistem dipastikan memiliki fondasi logika yang kuat dan siap untuk digunakan dalam lingkungan produksi maupun pengembangan lanjutan.

### 4.3 Hasil *Deployment* dan Analisis Capaian

Bagian ini membahas proses *deployment* sistem ke lingkungan produksi dan menganalisis sejauh mana sistem telah memenuhi tujuan penelitian dan kebutuhan yang teridentifikasi. *Backend* dan *frontend* ke layanan *hosting* yang berbeda. *Backend* yang dibangun dengan *Node.js* dan *Express.js*, serta basis data *PostgreSQL*, di-deploy di Railway. Sementara itu, *frontend* yang dikembangkan dengan *React.js* di-deploy di Vercel. Pemisahan ini memungkinkan skalabilitas independen untuk setiap komponen dan optimalisasi kinerja berdasarkan karakteristik masing-masing layanan *hosting*. Komunikasi antara *frontend* di Vercel dan *backend* di Railway dilakukan melalui *REST API*, dengan *frontend* mengarahkan permintaan ke *URL backend* yang telah ditentukan.

#### 4.3.1 Proses *Deployment Website*

##### A. Langkah-Langkah *Deployment Backend* di Railway

*Backend* aplikasi Museum Lampung di-deploy di Railway, sebuah platform *hosting* yang dirancang untuk menyederhanakan *deployment* aplikasi *Node.js* dan basis data. Langkah-langkah *deployment* yang dilakukan adalah sebagai berikut:

1. Persiapan Aplikasi *Node.js*: Memastikan aplikasi *Node.js* (*Express.js*) telah disiapkan dengan skrip start yang benar di *package.json* dan mengelola *port* aplikasi agar dapat diatur oleh lingkungan *hosting*.
2. Integrasi dengan GitHub: Kode sumber *backend* di-*push* ke repositori GitHub. Railway terintegrasi dengan GitHub, memungkinkan *deployment* otomatis setiap kali ada *commit* baru ke cabang utama.
3. Login ke Railway: Masuk ke *dashboard* Railway menggunakan akun GitHub.
4. Membuat Proyek Baru: Pilih opsi "New Project" dan kemudian "Deploy from GitHub repo". Pilih repositori *backend* yang telah di-*push* sebelumnya. Railway akan secara otomatis mendeteksi aplikasi *Node.js* dan memulai proses

*build* dan *deployment*.

5. Konfigurasi Variabel Lingkungan: Mengatur variabel lingkungan penting seperti `DB_NAME`, `DB_USER`, `DB_PASS`, `DB_HOST`, `JWT_SECRET`, `CLOUDINARY_CLOUD_NAME`, `CLOUDINARY_API_KEY`, dan `CLOUDINARY_API_SECRET` di bagian "Variables" pada *dashboard* proyek Railway. Variabel-variabel ini sangat penting untuk koneksi basis data dan layanan eksternal seperti Cloudinary.
6. Penambahan *Plugin Database* (PostgreSQL): Di *dashboard* proyek Railway, pilih "Add Plugin" dan tambahkan basis data *PostgreSQL*. Railway akan secara otomatis menyediakan dan menghubungkan basis data, serta membuat variabel lingkungan seperti `DATABASE_URL` yang dapat digunakan oleh aplikasi. Meskipun aplikasi menggunakan variabel `DB_HOST`, `DB_NAME`, dll., `DATABASE_URL` dapat digunakan sebagai alternatif atau disesuaikan dalam konfigurasi.
7. Verifikasi *Deployment*: Memantau log *deployment* di *dashboard* Railway untuk memastikan tidak ada kesalahan selama proses *build* dan *deployment*.
8. Akses *URL Publik*: Setelah *deployment* berhasil, Railway akan menyediakan *URL publik* untuk *backend*. *URL backend* yang digunakan adalah [HTTPS://museumlampung-production.up.railway.app](https://museumlampung-production.up.railway.app).

#### B. Langkah-Langkah *Deployment Frontend* di Vercel:

*Frontend* aplikasi Museum Lampung di-deploy di Vercel, sebuah platform yang dioptimalkan untuk *deployment* aplikasi *frontend React.js*. Langkah-langkah *deployment* yang dilakukan adalah sebagai berikut:

1. Persiapan Aplikasi *React.js*: Memastikan aplikasi *React.js* telah di-*build* secara lokal menggunakan perintah `npm run build` untuk menghasilkan berkas produksi yang dioptimalkan.
2. Integrasi dengan GitHub: Kode sumber *frontend* di-*push* ke repositori GitHub. Vercel terintegrasi dengan GitHub, memungkinkan *deployment* otomatis setiap kali ada *commit* baru.
3. Login ke Vercel: Masuk ke *dashboard* Vercel menggunakan akun GitHub.

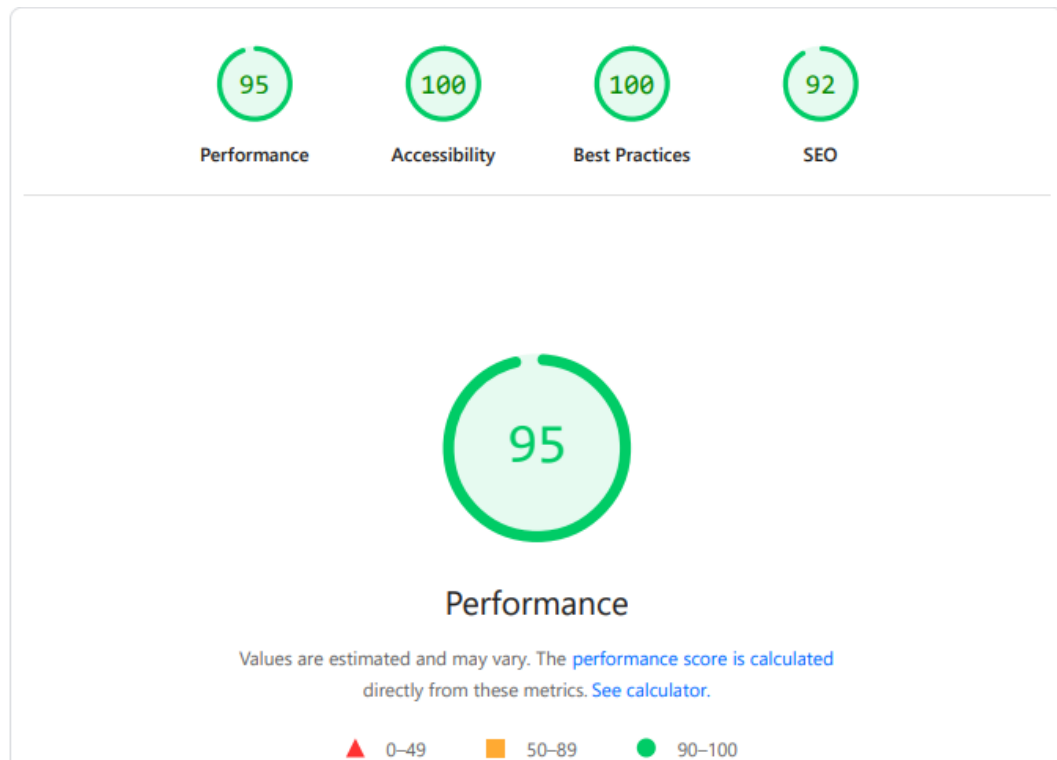
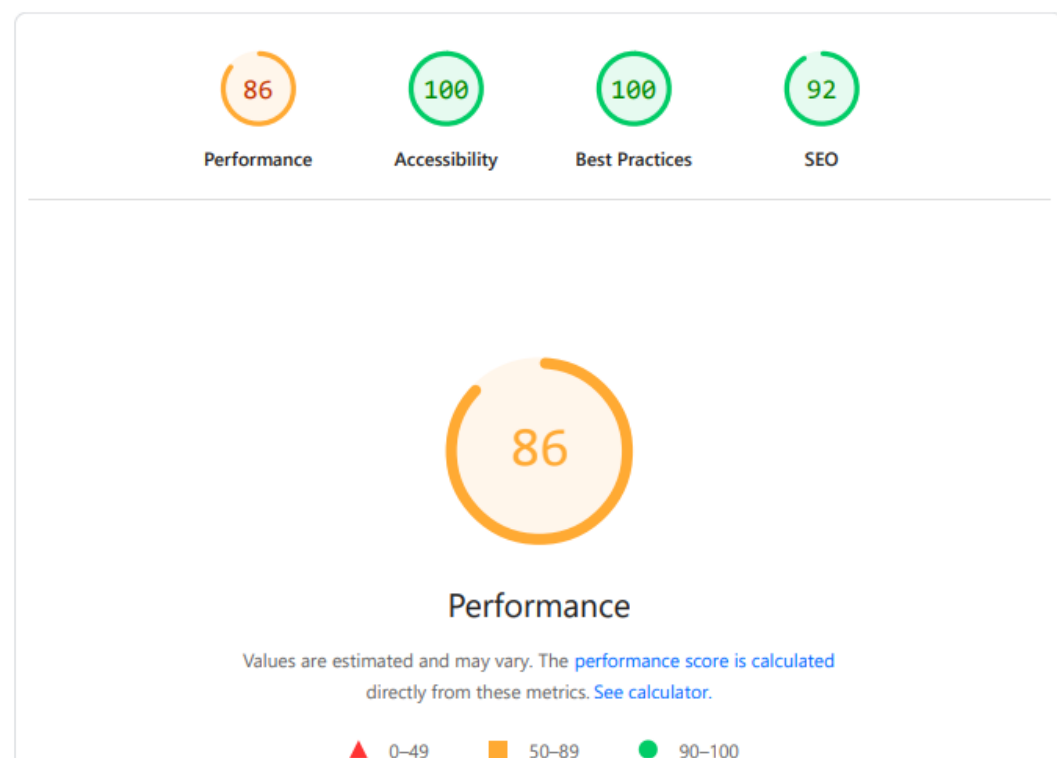
4. Mengimpor Proyek: Klik "Add New" dan pilih "Project". Pilih repositori GitHub *frontend* yang telah di-*push*. Vercel akan secara otomatis mendeteksi bahwa ini adalah aplikasi React.js.
5. Konfigurasi Variabel Lingkungan: Mengatur variabel lingkungan yang diperlukan, seperti `REACT_APP_API_URL` yang menunjuk ke *URL backend* yang telah di-*deploy* (`HTTPS://museumlampung-production.up.railway.app`). Variabel ini sangat penting agar *frontend* dapat berkomunikasi dengan *backend* yang benar.
6. Memulai *Deployment*: Klik tombol "Deploy". Vercel akan mengambil kode dari repositori, menjalankan proses *build*, dan mendeploy aplikasi.
7. Verifikasi *Deployment*: Memantau log *deployment* di *dashboard* Vercel untuk memastikan proses berjalan lancar.
8. Akses *URL* Publik: Setelah *deployment* berhasil, Vercel akan menyediakan *URL* publik untuk *frontend*. *URL frontend* yang digunakan adalah `HTTPS://frontend-museum-oe1o.vercel.app`.

#### 4.3.2 Analisis Performa Berdasarkan Pengujian *Desktop* dan *Mobile*

Pengujian performa sistem digitalisasi koleksi Museum Lampung menggunakan PageSpeed Insights menghasilkan nilai yang berbeda antara perangkat *desktop* dan *mobile*. Hasil pengujian ditunjukkan pada Tabel 4.4.

Tabel 4.4 Perbandingan Nilai *Desktop* dan *Mobile*

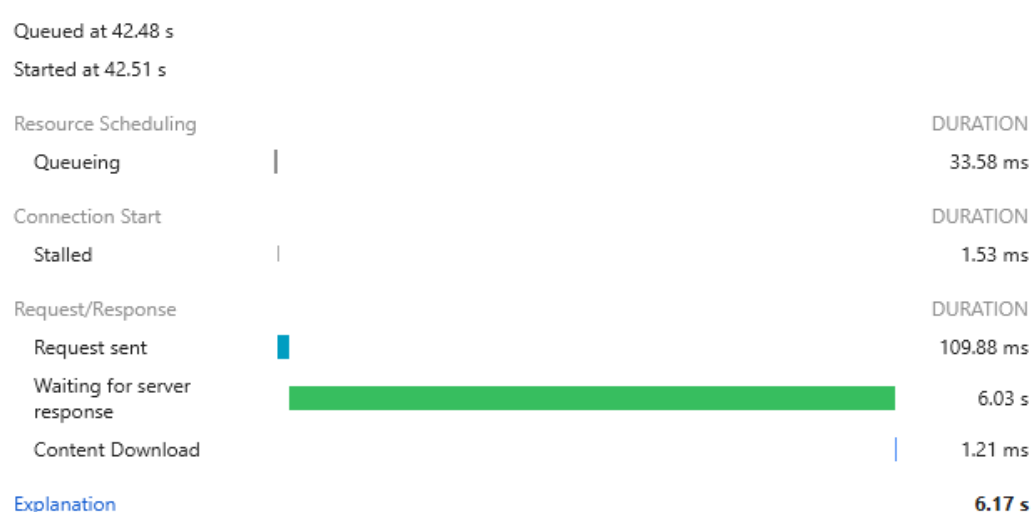
| Indikator      | Desktop (%) | Mobile (%) |
|----------------|-------------|------------|
| Performance    | 95          | 86         |
| Accessibility  | 100         | 100        |
| Best Practices | 100         | 100        |
| SEO            | 92          | 92         |

Gambar 4.35 Hasil Pengujian Performa *Desktop*Gambar 4.36 Hasil Pengujian Performa *Mobile*

Gambar 4.35 dan Gambar 4.36, menunjukkan hasil bahwa skor *Performance* pada desktop (95%) lebih tinggi dibanding mobile (86%). Perbedaan ini wajar mengingat pengujian mobile menggunakan simulasi perangkat dengan spesifikasi dan koneksi lebih terbatas. Sementara itu, skor *Accessibility* dan *Best Practices* mencapai 100% pada kedua platform, menandakan pemenuhan penuh terhadap kriteria aksesibilitas dan praktik terbaik pengembangan aplikasi. Skor *SEO* identik pada kedua platform (92%), mengindikasikan optimasi mesin pencari telah dilakukan dengan baik meskipun terdapat catatan minor terkait validitas *file robots.txt*.

### 4.3.3 Analisis Waktu Respons *Upload* Gambar dan Video

Dalam proses pengujian aplikasi pada tahap *deployment*, dilakukan analisis terhadap waktu respons saat melakukan proses unggah (*upload*) berkas, baik untuk gambar maupun video. Pengujian ini bertujuan untuk mengevaluasi performa aplikasi dalam menangani media dengan ukuran dan karakteristik yang berbeda, sehingga dapat diketahui kualitas layanan yang diberikan kepada pengguna. Berikut adalah hasil uji coba *upload* gambar dan video yang divisualisasikan pada Gambar 4.6 dan Gambar 4.7.



Gambar 4.37 Hasil Pengujian *Upload* Gambar

Dari Gambar 4.37, tercatat bahwa proses *upload* gambar memerlukan waktu total selama 6,17 detik. Rincian lebih lanjut menunjukkan bahwa waktu terbesar terletak pada tahap *waiting for server response*, yakni selama 6,03 detik, sedangkan tahap lain seperti *request sent* dan *content download* membutuhkan waktu relatif singkat,

masing-masing 109,88 ms dan 1,21 detik. Hal ini menandakan bahwa proses pengiriman data gambar ke server dan penanganannya oleh server berlangsung cukup efisien, namun terdapat waktu tunggu yang dapat dioptimalkan pada sisi server.



Gambar 4.38 Hasil Pengujian *Upload Video*

Sementara itu, berdasarkan Gambar 4.38, *upload* video memerlukan waktu yang jauh lebih lama yaitu 44,91 detik. Pada proses ini, tahap *request sent* memakan waktu paling besar, yaitu 29,61 detik, diikuti oleh *waiting for server response* selama 15,26 detik, dan *content download* hanya 1,74 detik. Besarnya waktu pada tahap *request sent* menunjukkan bahwa proses transfer video dari sisi klien ke server memerlukan bandwidth besar dan waktu yang lebih lama akibat ukuran *file* video yang tinggi. Selain itu, waktu tunggu server juga meningkat dibandingkan proses upload gambar, yang kemungkinan disebabkan oleh proses *decoding*, kompresi, atau penyimpanan video di server yang lebih kompleks.

#### 4.3.4 Analisis Fungsionalitas Sistem

Analisis ini menilai sejauh mana seluruh fitur utama sistem telah berjalan sesuai dengan rancangan awal dan memenuhi kebutuhan pengguna yang teridentifikasi. Seluruh fitur utama yang dirancang, seperti *login*, pengelolaan koleksi (*CRUD*), pencarian dan *filter* koleksi, integrasi media 360°, dan *QR Code*, telah berfungsi dengan baik. Fitur manajemen kegiatan dan manajemen user oleh superadmin juga berjalan sesuai spesifikasi. Ini mengkonfirmasi keberhasilan implementasi dari

rancangan fungsional yang telah ditetapkan.

Keberhasilan implementasi seluruh fitur utama, termasuk manajemen koleksi, pencarian, *filter*, integrasi media 360°, *QR Code*, serta manajemen kegiatan dan pengguna, secara langsung meningkatkan aksesibilitas informasi koleksi museum bagi masyarakat luas. Pengunjung kini dapat menjelajahi koleksi secara digital dengan mudah, sementara pengelola museum dapat mendokumentasikan dan memperbarui data dengan lebih efisien. Hal ini tidak hanya mengurangi risiko kehilangan atau kerusakan data yang sebelumnya terjadi pada sistem manual, tetapi juga mempercepat proses pemeliharaan dan penyebaran informasi. Peningkatan aksesibilitas dan efisiensi manajemen ini secara fundamental mengubah cara museum berinteraksi dengan koleksinya dan publik, memungkinkan jangkauan yang lebih luas dan pengelolaan yang lebih sistematis.

#### 4.3.5 Analisis Keamanan Sistem

Aspek keamanan sistem dievaluasi berdasarkan implementasi mekanisme autentikasi dan perlindungan data.

##### a. Autentikasi Berbasis JWT:

Sistem telah menerapkan autentikasi berbasis *JSON Web Token* (JWT) untuk membedakan hak akses antara pengguna umum dan admin. Mekanisme ini memastikan bahwa akses ke fitur terbatas, seperti pengelolaan koleksi dan pengguna, hanya diberikan kepada peran pengguna yang berwenang, sehingga mencegah akses tidak sah.

##### b. Enkripsi *Password* dengan *Bcrypt*:

Kata sandi pengguna dienkripsi menggunakan algoritma *hashing bcrypt* dengan penambahan *salt* otomatis. Pendekatan ini melindungi kata sandi dari serangan *brute-force* dan *rainbow table*, bahkan jika basis data berhasil ditembus, karena hash yang dihasilkan unik untuk setiap kata sandi, meskipun kata sandinya sama.

##### c. Kontrol Akses Berbasis Peran (RBAC):

Sistem menerapkan *Role-Based Access Control* (RBAC) yang secara ketat

membatasi hak akses pengguna berdasarkan peran mereka (superadmin, admin koleksi, admin kegiatan, dan pengunjung). Setiap peran hanya dapat mengakses fungsionalitas yang relevan dengan tanggung jawabnya, meminimalkan risiko penyalahgunaan hak akses dan menjaga integritas data.

Implementasi autentikasi berbasis JWT, enkripsi kata sandi dengan *bcrypt*, dan kontrol akses berbasis peran (RBAC) secara kolektif memastikan integritas data dan mitigasi risiko keamanan yang signifikan. Mekanisme ini tidak hanya melindungi informasi sensitif pengguna dari akses tidak sah, tetapi juga mencegah modifikasi data yang tidak sah dan serangan siber. Dengan membatasi akses berdasarkan peran dan mengenkripsi kredensial, sistem membangun lapisan pertahanan yang kuat, mengurangi potensi kerentanan, dan meningkatkan kepercayaan pengguna terhadap keamanan informasi yang disimpan. Ini merupakan langkah krusial dalam menjaga keandalan dan kepercayaan sistem informasi di lingkungan digital.

#### **4.3.6 Analisis Aksesibilitas dan Antarmuka Pengguna**

Analisis ini menilai kemudahan penggunaan dan jangkauan sistem bagi berbagai jenis pengguna.

##### **a. Aksesibilitas Multi-Perangkat:**

Sistem dapat diakses melalui berbagai perangkat, termasuk *desktop*, tablet, dan *smartphone*, berkat desain antarmuka yang responsif. Hal ini meningkatkan jangkauan dan kemudahan akses bagi pengunjung, memungkinkan mereka untuk menjelajahi koleksi museum kapan saja dan di mana saja.

##### **b. Antarmuka Intuitif:**

Tampilan aplikasi dirancang responsif dan intuitif, memudahkan interaksi pengguna dalam menjelajahi sistem, baik dari sisi admin maupun pengunjung umum. Navigasi yang jelas, tata letak yang bersih, dan umpan balik visual yang konsisten berkontribusi pada pengalaman pengguna yang positif. Desain antarmuka yang responsif dan intuitif secara signifikan meningkatkan jangkauan audiens dan pengalaman pengguna. Kemampuan sistem untuk diakses dengan mulus dari berbagai perangkat, mulai dari *desktop* hingga *smartphone*, menghilangkan

hambatan aksesibilitas geografis dan teknis, memungkinkan museum menjangkau audiens yang lebih luas secara global.

## V. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan hasil implementasi dan pengujian sistem digitalisasi dan manajemen koleksi Museum Lampung, didapatkan kesimpulan sebagai berikut:

1. Sistem telah berhasil diimplementasikan menggunakan *PERN Stack* (*PostgreSQL*, *Express.js*, *React.js*, *Node.js*), menyediakan platform terintegrasi untuk pengelolaan koleksi museum.
2. Penggunaan *PostgreSQL* dengan *Sequelize ORM* secara efektif mendukung penyimpanan dan manajemen data koleksi secara terstruktur dan terintegrasi, memastikan integritas dan konsistensi data.
3. Mekanisme autentikasi berbasis *JSON Web Token* (JWT) dan *hashing password* dengan Bcrypt, dikombinasikan dengan *Role-Based Access Control* (RBAC), berhasil membedakan hak akses antara pengunjung, admin koleksi, admin kegiatan, dan *superadmin*, meningkatkan keamanan sistem.
4. Fitur digitalisasi koleksi yang mencakup pengunggahan foto, video 360°, dan pembuatan *QR Code* telah berhasil diimplementasikan, memungkinkan dokumentasi visual yang komprehensif dan interaksi pengunjung yang lebih menarik.
5. Pengujian sistem melalui Postman, *black-box*, dan Jest menunjukkan bahwa seluruh fungsionalitas *backend* dan *frontend* berjalan sesuai spesifikasi dengan tingkat keandalan yang tinggi, serta penanganan kesalahan yang memadai.
6. *Deployment* sistem pada Railway (untuk *backend*) dan Vercel (untuk *frontend*) berhasil dilakukan, menjadikan aplikasi dapat diakses secara publik dan stabil.

7. Proses unggah gambar menunjukkan durasi 6,17 detik dengan sebagian besar waktu berada pada tahap *waiting for server response*, sedangkan unggah video memerlukan waktu 44,91 detik dengan durasi terpanjang pada tahap *request sent*. Perbedaan signifikan ini menunjukkan bahwa ukuran *file* dan kompleksitas media memengaruhi kinerja unggah media.
8. Skor *Performance* yang tinggi pada desktop (95%) dan mobile (86%), serta skor *Accessibility* dan *Best Practices* maksimal, menunjukkan bahwa sistem telah dirancang inklusif dan efisien, meskipun masih terdapat ruang optimasi untuk kinerja perangkat mobile.

## 5.2 Saran

Berdasarkan capaian dan potensi pengembangan lebih lanjut, beberapa saran dapat diajukan:

1. Mengembangkan modul analitik yang lebih canggih untuk melacak pola kunjungan, koleksi yang paling diminati, dan demografi pengunjung, yang dapat memberikan *insight* berharga bagi manajemen museum.
2. Meskipun *PostgreSQL* skalabel, untuk volume data yang sangat besar di masa depan, pertimbangkan strategi *sharding* atau *clustering* basis data untuk memastikan kinerja tetap optimal.
3. Menerapkan otentikasi dua faktor (2FA) untuk akun admin dan *superadmin* guna menambah lapisan keamanan, serta melakukan *penetration testing* secara berkala untuk mengidentifikasi dan mengatasi potensi kerentanan.
4. Menambahkan fitur-fitur edukasi interaktif, seperti kuis, narasi audio, atau *timeline* sejarah yang terkait dengan koleksi, untuk meningkatkan nilai pendidikan sistem.
5. Membangun fitur berbagi koleksi ke media sosial secara langsung dari aplikasi untuk meningkatkan visibilitas museum dan jangkauan.

6. Menerapkan mekanisme mitigasi jika token autentikasi diretas, seperti penggunaan *refresh token*, masa berlaku *access token* yang singkat, serta fitur revokasi atau *blacklist token* untuk menonaktifkan token yang tidak sah.
7. Lakukan optimalisasi kinerja unggah media dengan menerapkan teknik *client-side compression* untuk video berukuran besar. Selain itu, tingkatkan performa pada perangkat seluler dengan mengimplementasikan *Progressive Web App* (PWA) atau *Server-Side Rendering* (SSR) untuk mengurangi waktu pemuatan dan memberikan pengalaman pengguna yang lebih baik.

## DAFTAR PUSTAKA

- [1] M. Muhaemin, "Strategi Digital Curator Dalam Mengelola Arsip Dan Koleksi Digital: Sebuah Kajian Literatur," *Bibliotika: Jurnal Kajian Perpustakaan dan Informasi*, vol. 8, no. 1, pp. 64–76, 2024.
- [2] B. Wicaksono and S. Sunarmi, "Digital Aset Management Sebagai Strategi Keberlangsungan Museum Prabu Gesaun Ulun di Era Keterbukaan Informasi Publik," *Avant Garde*, vol. 6, no. 1, pp. 1–15, 2024.
- [3] E. Y. Putra, C. Haerani, R. Sidomulyo, V. Rondonuwu, C. Golung, and A. Stenly, "Digitalisasi Museum Berbasis Web Dengan AR Dan QR Code (Studi Kasus Pada Museum Sulawesi Utara)," *Jurnal Teknik Elektro dan Komputer TRIAC*, vol. 11, no. 1, pp. 24–30, 2024.
- [4] Y. L. N. Sitepu, M. P. A. Tarigan, and C. R. M. Siregar, "Sistem Penyimpanan Kemuseumman," *Jurnal Sadewa: Publikasi Ilmu Pendidikan, Pembelajaran dan Ilmu Sosial*, vol. 2, no. 3, pp. 101–108, 2024.
- [5] F. Hidayat and R. Aulia, "Pengelolaan Koleksi Di Museum Musik Indonesia Sebagai Upaya Pelestarian Warisan Budaya," *Fihris Jurnal Ilmu Perpustakaan*, vol. 7, no. 1, pp. 56–70, 2021.
- [6] J. J. Medina, J. M. Maley, S. Sannapareddy, N. N. Medina, C. M. Gilman, and J. E. McCormack, "A rapid and cost-effective pipeline for digitization of museum specimens with 3D photogrammetry," *PLOS ONE*, vol. 15, no. 8, Art. no. e0236417, 2020.
- [7] D. Puspasari, "Designing Mobile Application for Museum Enthusiasts," in *Proc. Indonesia Heritage Tourism Forum (IHTF)*, 2019, pp. 3–7, 2020.
- [8] C. Meier, I. S. Berriel, and F. P. Nava, "Creation of a Virtual Museum for the Dissemination of 3D Models of Historical Clothing," *Sustainability*, vol. 13, no. 22, Nov. 2021.
- [9] N. Istiawan and Nuralia, "Perancangan Sistem Informasi Manajemen

- Koleksi Museum Berbasis Web (Studi Kasus Museum Negeri Provinsi Lampung)," *Jurnal Informatika dan Rekayasa Perangkat Lunak*, vol. 2, no. 1, pp. 102–109, Mar. 2021.
- [10] T. Keep, "Digitization of Museum Collections and the Hellenic Museum Digitization Project," *Iris | Journal of the Classical Association of Victoria*, vol. 32–33, pp. 17–18, 2020.
- [11] Y. Bunari and A. Fikri, "Rancang Bangun Aplikasi Museum Digital Berbasis Android," *Jurnal Inovtek Polbeng - Seri Informatika*, vol. 7, no. 2, pp. 205–210, 2022.
- [12] R. Angeloni, "Digitization and Virtual Experience of Museum Collections: The Virtual Tour of the Civic Art Gallery of Ancona," *SCIRES-IT*, vol. 12, no. 2, pp. 29–42, 2022.
- [13] L. Zhao, J. Zhang, H. Jing, J. Wu, and Y. Huang, "A Blockchain-Based Cryptographic Interaction Method of Digital Museum Collections," *Journal of Cultural Heritage*, vol. 59, pp. 69–82, Nov. 2023.
- [14] S. N. Mohammed and H. K. Metwaly, "Digitization and the Collection Sustainability: Report on the Grand Egyptian Museum Project, Egypt," *Studies in Digital Heritage*, vol. 7, no. 2, pp. 161–174, Dec. 2023.
- [15] Z. Gao, T. Braud, and V. Guljajeva, "VR-driven museum opportunities: digitized archives in the age of the metaverse," *Artnodes*, no. 32, pp. 1–10, Jul. 2023.
- [16] S. G. Prasuteja, A. Wibowo, D. F. Racma, R. Y. Widiastuti, and Suyudi, "Website-based Collection Inventory Information System Design at Wayang Sendang Mas Banyumas Museum," *Journal of Intelligent Systems and Information Technology*, vol. 1, no. 2, pp. 94–103, Aug. 2024.
- [17] D. C. Blackburn, D. M. Boyer, J. A. Gray, J. Winchester, J. M. Bates, S. L. Baumgart, et al., "Increasing the impact of vertebrate scientific collections through 3D imaging: The openVertebrate (oVert) Thematic Collections Network," *BioScience*, vol. 74, no. 3, pp. 169–186, 2024.
- [18] H. Mahboub, H. Sadok, A. Chehri, and R. Saadane, "Measuring the Digital Transformation: A Key Performance Indicators Literature Review," *Procedia Computer Science*, vol. 225, pp. 4570–4579, 2023.

- [19] S. Styliani, L. Fotis, K. Kostas, and P. Petros, "Virtual museums, a survey and some issues for consideration," *Journal of Cultural Heritage*, vol. 10, no. 4, pp. 520–528, 2020.
- [20] J. Li, X. Zheng, I. Watanabe, and Y. Ochiai, "A systematic review of digital transformation technologies in museum exhibition," *Computers in Human Behavior*, 108407, 2024.
- [21] A. Auliahadi, "Problem Manajemen Pengembangan Koleksi di Museum Kerinci," *Shaut Al-Maktabah: Jurnal Perpustakaan, Arsip dan Dokumentasi*, vol. 12, no. 1, pp. 85–100, 2020.
- [22] M. I. Hossain, "Software Development Life Cycle (SDLC) Methodologies for Information Systems Project Management," *Int. J. Multidiscip. Res.*, vol. 5, no. 5, pp. 1–13, Sep.–Oct. 2023.
- [23] G. Singh, M. Javed, and B. K. Dhaliwal, "Full Stack Web Development: Vision, Challenges and Future Scope," *International Research Journal of Engineering and Technology (IRJET)*, vol. 9, no. 4, Apr. 2022.
- [24] S. bin Uzayr, *Mastering Visual Studio Code: A Beginner's Guide*. CRC Press, 2022.
- [25] P. P. Kore, M. J. Lohar, M. T. Surve, and S. Jadhav, "API Testing Using Postman Tool," *International Journal for Research in Applied Science and Engineering Technology*, vol. 10, no. 12, pp. 841–843, 2022.
- [26] Z. Toufie and B. Kabaso, "The Next Evolution of Web Browser Execution Environment Performance," in *Proc. 2023 Int. Conf. Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, pp. 1–7, 2023.
- [27] N. A. Alhazmy, Z. N. Chandra, P. Atmadiputra, and Y. Triyana, "Building a Comprehensive Content Management System with NPM, Vue.js, Node.js, PostgreSQL, and Strapi," *Liaison Journal of Engineering*, vol. 3, 2023.
- [28] B. Sutara and S. S. Gunawan, "Comparative Analysis Of Rest API Performance Between Express.js Framework And Hapi.js Using Apache JMeter," *Jurnal Riset Teknik Informatika*, vol. 1, no. 1, pp. 19–26, 2024.
- [29] M. F. S. Lazuardy and D. Anggraini, "Modern Front End Web Architectures with React.js and Next.js," *Research Journal of Advanced Engineering and*

- Science*, vol. 7, no. 1, pp. 132–141, 2022.
- [30] P. Filip and L. Čegan, "Comparison of MySQL and MongoDB with Focus on Performance," in *Proc. 2020 Int. Conf. Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, pp. 184–187, 2020.
- [31] M. Levoy and P. Hanrahan, "Light Field Rendering," in *Seminal Graphics Papers: Pushing the Boundaries*, vol. 2, pp. 441–452, 2023.
- [32] D. Vinaykumar *et al.*, "Efficient File Sharing System Utilizing MongoDB and Node.js," *International Journal of Computational Learning & Intelligence*, vol. 3, no. 1, pp. 177–182, 2024.
- [33] A. Sharma, P. Singh, and R. A. Choudhury, "Providing Authentication using JSON Web Tokens for Enhancing User Security," *International Journal of Research Publication and Reviews*, vol. 5, no. 4, pp. 5309–5312, 2024.