

**IMPLEMENTASI ARSITEKTUR RESTFUL WEB
SERVICE MENGGUNAKAN FRAMEWORK
EXPRESS PADA APLIKASI ARSIP
MEDIA BERBASIS NUXT**

SKRIPSI

Untuk memenuhi syarat memperoleh gelar sarjana Strata 1
Program Studi Teknik Informatika



Oleh:

Dede Ariansya	172175
Welni Welce Solarbesain	172168

**PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS DIPA MAKASSAR
2023**

**IMPLEMENTASI ARSITEKTUR RESTFUL WEB
SERVICE MENGGUNAKAN FRAMEWORK
EXPRESS PADA APLIKASI ARSIP
MEDIA BERBASIS NUXT**

SKRIPSI

Untuk memenuhi syarat memperoleh gelar sarjana Strata 1
Program Studi Teknik Informatika

Dede Ariansya	172175
Welni Welce Solarbesain	172168

Telah disetujui untuk dipertahankan

Pembimbing I

Annah, S.Kom., MT.
NIDN. 0907087903

Pembimbing II,

Hasyrif S Y, S.Kom., MT.
NIDN. 0916068301

PENGESAHAN NASKAH SKRIPSI

Judul : Implementasi Arsitektur Restful Web Service Menggunakan Framework Express Pada Aplikasi Arsip Media Berbasis Nuxt
Mahasiswa 1 : Dede Ariansya
Nim 1 : 172175
Mahasiswa 2 : Welni Welce Solarbesain
Nim 2 : 172168
Tanggal Ujian : 03 Maret 2023

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Annah, S.Kom., MT.
NIDN. 0907087903

Hasyrif S Y, S.Kom., MT.
NIDN. 0916068301

Penguji I,

Penguji II,

Salman, S.Kom., MT.
NIDN. 0025027801

Sri Wahyuni, S.Kom., MT.
NIDN. 0931039002

Mengetahui
Ketua Program Studi Teknik Informatika
Universitas Dipa Makassar

Ir. Irsal, MT.
NIDN. 9990216745

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa atas segala nikmat, karunia dan limpahan rahmat-Nya yang telah memberikan kekuatan kepada penulis sehingga skripsi ini dapat diselesaikan, yang merupakan salah satu persyaratan dalam penyelesaian pendidikan Strata Satu di Universitas Dipa Makassar.

Berbagai kendala dan keterbatasan ditemui dalam penyusunan skripsi ini dari tahap persiapan hingga penyelesaiannya, namun kendala dan kesulitan tersebut dapat diatasi berkat bimbingan dan kerjasama dari berbagai pemangku kepentingan.

Oleh Karena itu perkenankanlah penulis dengan segala kerendahan hati menyampaikan ucapan terima kasih yang tulus dan penghargaan yang tak terhingga kepada:

1. Dr. Jhony W. Soetikno, SE,MM. selaku Rektor Universitas Dipa Makassar.
2. Ir. Irsal, MT. selaku Ketua Program Studi Teknik Informatika Universitas Dipa Makassar.
3. Annah, S.Kom., MT. selaku Pembimbing I, yang telah membimbing penulis dalam penyelesaian skripsi ini.
4. Hasyrif S Y, S.Kom., MT. selaku Pembimbing II, yang telah banyak memberikan bimbingan dan arahan penulis dalam penyelesaian skripsi ini.
5. Dosen Universitas Dipa Makassar yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis.

6. Kedua orang tua tercinta yang tak bosan-bosannya memberikan nasehat dan dukungan yang tidak dapat kami nilai dalam bentuk apapun. Semoga Tuhan selalu senantiasa melimpahkan kesehatan dan kesejahteraan bagi beliau, Amin.
7. Untuk semua teman-teman tanpa terkecuali yang tidak dapat disebutkan namanya, terimakasih untuk setiap bantuan yang telah kalian berikan kepada penulis.

Penulis menyadari bahwa skripsi ini masih banyak terdapat kekurangan. Oleh karena itu penulis sangat mengharapkan kritik dan saran yang membangun, dan semoga skripsi ini dapat bermanfaat bagi yang membacanya. Akhirnya teriring do'a dan harapan semoga segala bantuan yang telah diberikan baik materil maupun moril mendapat imbalan disisi Tuhan yang maha esa dan bermanfaat bagi kita semua. Amin

Makassar, Maret 2023

P e n u l i s

ABSTRAK

Dinas Kesehatan Provinsi Sulawesi Selatan merupakan unsur pelaksana urusan pemerintahan dalam bidang Kesehatan dan obat-obatan Provinsi Sulawesi Selatan. Ada banyak kegiatan yang biasa dilakukan oleh Dinas Kesehatan Provinsi Sulawesi Selatan salah satunya adalah membuat dokumentasi pada saat diadakannya kegiatan yang mana dokumentasi tersebut diambil dalam bentuk gambar atau video yang nantinya akan diarsipkan oleh tiap-tiap pegawai pada bidang yang melakukan kegiatan. Penelitian ini mengimplementasikan arsitektur restful web service, menggunakan framework express dan framework nuxt. Penelitian ini menghasilkan aplikasi yang dapat digunakan untuk membuat album yang dapat menampung dokumen dokumentasi yang dapat diakses oleh pegawai dengan mudah dan cepat.

Kata kunci: Arsitektur Restful Web Service, Express JS, Nuxt JS.

ABSTRACT

The South Sulawesi Provincial Health Office is the executor of government affairs in the health and medicine sector of the South Sulawesi Province. There are many activities that are usually carried out by the Health Office of the Province of South Sulawesi, one of which is making documentation at the time the activity is held where the documentation is taken in the form of pictures or videos which will be archived by each employee in the field carrying out the activity. This study implements a restful web service architecture, using the express framework and the nuxt framework. This research produces an application that can be used to create albums that can accommodate documentation documents that can be accessed by employees easily and quickly.

Keywords: Restful Web Service Architecture, Express JS, Nuxt JS.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR.....	iv
ABSTRAK	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	3
1.5 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	6
2.1 Kerangka Pikir.....	6
2.2 Landasan Teori.....	7
2.2.1 HTTP	7
2.2.2 JSON	7

2.2.3 API.....	8
2.2.4 RESTful API	9
2.2.5 JavaScript	9
2.2.6 Node.JS.....	10
2.2.7 Express.JS.....	11
2.2.8 MYSQL.....	11
2.2.9 Vue.JS.....	12
2.2.10 Nuxt.JS	14
2.2.11 <i>Unified Modeling Language (UML)</i>	16
2.3 Penelitian Terkait	20
BAB III METODE PENELITIAN	22
3.1 Waktu dan Tempat Penelitian	22
3.2 Jenis Penelitian.....	22
3.3 Metode Pengumpulan Data	22
3.4 Bahan dan Alat Penelitian	23
3.4.1 Bahan Penelitian.....	23
3.4.2 Alat Penelitian	23
3.5 Metode Pengujian.....	24
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	25
4.1 Perancangan Solusi	25

4.1.1	<i>Use Case Diagram</i>	25
4.1.2	<i>Class Diagram</i>	26
4.1.3	<i>Activity Diagram</i>	27
4.1.4	<i>Sequence Diagram</i>	30
4.1.5	Arsitektur RESTful API.....	36
4.1.6	Rancangan Endpoint API.....	37
4.1.7	Rancangan Tabel	39
4.1.8	Rancangan <i>Interface</i>	40
4.2	Analisis dan Validasi Data	45
4.2.1	Analisis.....	45
4.2.2	Validasi Data	46
4.3	Hasil dan Pembahasan.....	47
4.3.1	Hasil dan Pembahasan Implementasi MYSQL	47
4.3.2	Hasil dan pembahasan implementasi Nuxt JS.....	49
4.3.3	Hasil dan Pembahasan Implementasi Express JS.....	51
4.3.4	Hasil dan Pembahasan Implementasi Node JS.....	51
4.4	Pengujian Sistem	52
4.4.1	Pengujian Halaman Login	52
4.4.2	Pengujian Halaman Peran	54
4.4.3	Pengujian Halaman User	58

4.4.4 Pengujian Halaman Album	64
BAB V KESIMPULAN DAN SARAN	70
5.1 Kesimpulan.....	70
5.2 Saran.....	70
DAFTAR PUSTAKA	71

LAMPIRAN

RIWAYAT HIDUP

DAFTAR GAMBAR

Gambar 2.1 Kerangka Berpikir.....	6
Gambar 2.2 <i>lifecycle hook</i> Vue.JS	13
Gambar 4.1 <i>Use Case Diagram</i>	26
Gambar 4.2 <i>Class Diagram</i>	26
Gambar 4.3 <i>Activity Diagram Login</i>	27
Gambar 4.4 <i>Activity Diagram Kelola Peran</i>	28
Gambar 4.5 <i>Activity Diagram Kelola Pengguna</i>	28
Gambar 4.6 <i>Activity Diagram Kelola Album</i>	29
Gambar 4.7 <i>Activity Diagram Akses Album</i>	29
Gambar 4.8 <i>Sequence diagram login</i>	30
Gambar 4.9 <i>Sequence diagram menampilkan daftar peran</i>	30
Gambar 4.10 <i>Sequence diagram membuat peran</i>	30
Gambar 4.11 <i>Sequence diagram update peran</i>	31
Gambar 4.12 <i>Sequence diagram hapus peran</i>	31
Gambar 4.13 <i>Sequence diagram menampilkan daftar pengguna</i>	31
Gambar 4.14 <i>Sequence diagram membuat pengguna</i>	32
Gambar 4.15 <i>Sequence diagram update pengguna</i>	32
Gambar 4.16 <i>Sequence diagram hapus pengguna</i>	32
Gambar 4.17 <i>Sequence diagram memberikan akses album pada pengguna</i>	33
Gambar 4.18 Sequence diagram hapus akses album pengguna.....	33
Gambar 4.19 <i>Sequence diagram menampilkan daftar album</i>	33
Gambar 4.20 <i>Sequence diagram menampilkan detail album</i>	34

Gambar 4.21 <i>Sequence</i> diagram membuat album.....	34
Gambar 4.22 <i>Sequence</i> diagram <i>update</i> album.....	34
Gambar 4.23 <i>Sequence</i> diagram hapus album	35
Gambar 4.24 <i>Sequence</i> diagram <i>upload</i> media.....	35
Gambar 4.25 <i>Sequence</i> diagram hapus media.....	35
Gambar 4.26 <i>Sequence Diagram</i> Akses Album	35
Gambar 4.27 Rancangan halaman login	41
Gambar 4.28 Rancangan halaman <i>home</i>	41
Gambar 4.29 Rancangan halaman detail album user	41
Gambar 4.30 Rancangan <i>modal preview media</i>	42
Gambar 4.31 Rancangan halaman <i>dashboard</i>	42
Gambar 4.32 Rancangan halaman daftar peran	42
Gambar 4.33 Rancangan modal buat dan edit peran.....	43
Gambar 4.34 Rancangan halaman daftar user.....	43
Gambar 4.35 Rancangan modal buat dan edit user	43
Gambar 4.36 Rancangan modal <i>give</i> dan <i>remove</i> album user	44
Gambar 4.37 Rancangan halaman daftar album	44
Gambar 4.38 Rancangan modal buat album	44
Gambar 4.39 Rancangan halaman detail album.....	45
Gambar 4.40 Rancangan modal <i>preview media</i>	45
Gambar 4.41 CLI instalasi <i>package</i> prisma	47
Gambar 4.42 CLI menampilkan hasil implementasi MYSQL.	48
Gambar 4.43 Kode membuat skema tabel User.....	48

DAFTAR TABEL

Tabel 2.1 Simbol <i>use case</i> diagram.....	16
Tabel 2.2 Simbol <i>Activity</i> Diagram.....	18
Tabel 2.3 Simbol <i>Sequence</i> Diagram	19
Tabel 2.4 Simbol <i>Class</i> Diagram	20
Tabel 3.1 Bahan Penelitian	23
Tabel 3.2 Perangkat lunak yang digunakan	23
Tabel 4.1 Rancangan Endpoint API.....	37
Tabel 4.2 Skema tabel Role	39
Tabel 4.3 Skema tabel User	39
Tabel 4.4 Skema tabel Album.....	39
Tabel 4.5 Skema tabel Media.....	40
Tabel 4.6 Skema tabel UsersOnAlbums	40
Tabel 4.7 Daftar nama kegiatan dan jumlah dokumen	46
Tabel 4.8 Daftan pegawai yang dapat mengakses album	46
Tabel 4.9 Hasil pembuatan skema tabel User	48
Tabel 4.10 Pengujian <i>backend</i> fungsi login.....	52
Tabel 4.11 Pengujian <i>frontend</i> fungsi login.....	53
Tabel 4.12 Pengujian <i>backend</i> fungsi buat peran.....	54
Tabel 4.13 Pengujian <i>frontend</i> fungsi buat peran.	55
Tabel 4.14 Pengujian <i>backend</i> fungsi edit peran.	56
Tabel 4.15 Pengujian <i>frontend</i> fungsi edit peran.	56
Tabel 4.16 Pengujian <i>backend</i> fungsi hapus peran.	57

Tabel 4.17 Pengujian <i>frontend</i> fungsi hapus peran.....	57
Tabel 4.18 Pengujian <i>backend</i> fungsi buat user.....	58
Tabel 4.19 Pengujian <i>frontend</i> fungsi buat user.....	59
Tabel 4.20 Pengujian <i>backend</i> fungsi edit user.....	60
Tabel 4.21 Pengujian <i>frontend</i> fungsi edit user.	60
Tabel 4.22 Pengujian <i>backend</i> fungsi berikan dan hapus album user.....	61
Tabel 4.23 Pengujian <i>frontend</i> fungsi berikan dan hapus album user.	62
Tabel 4.24 Pengujian <i>backend</i> fungsi hapus user.	63
Tabel 4.25 Pengujian <i>frontend</i> fungsi hapus user.	63
Tabel 4.26 Pengujian <i>backend</i> fungsi buat album.	64
Tabel 4.27 Pengujian <i>frontend</i> fungsi buat album.	64
Tabel 4.28 Pengujian <i>backend</i> fungsi edit informasi album.	65
Tabel 4.29 Pengujian <i>frontend</i> fungsi edit informasi album.....	66
Tabel 4.30 Pengujian <i>backend</i> fungsi upload media.....	67
Tabel 4.31 Pengujian <i>frontend</i> fungsi upload media.	67
Tabel 4.32 Pengujian <i>backend</i> fungsi hapus media.	68
Tabel 4.33 Pengujian <i>frontend</i> fungsi hapus media.	68
Tabel 4.34 Pengujian <i>backend</i> fungsi hapus album.	69
Tabel 4.35 Pengujian <i>frontend</i> fungsi hapus album.....	69

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dinas Kesehatan Provinsi Sulawesi Selatan merupakan unsur pelaksana urusan pemerintahan dalam bidang Kesehatan dan obat-obatan Provinsi Sulawesi Selatan. Ada banyak kegiatan yang biasa dilakukan oleh Dinas Kesehatan Provinsi Sulawesi Selatan salah satunya adalah membuat dokumentasi pada saat diadakannya sebuah kegiatan yang mana dokumentasi tersebut diambil dalam bentuk gambar atau video yang nantinya akan diarsipkan oleh tiap-tiap pegawai pada bidang yang melakukan kegiatan. Contohnya jika yang melakukan kegiatan di bidang Sekretariat maka yang bertugas untuk melakukan arsip terhadap gambar atau video dokumentasi kegiatan adalah sub bagian umum.

Saat ini file dokumentasi disimpan pada laptop milik pegawai di tiap-tiap bidang yang bertanggung jawab untuk menyimpan file dokumentasi. Yang mana, ukuran dari file-file dokumentasi ini akan bertambah besar seiring berjalannya waktu dan membuat penanggung jawab harus selalu meningkatkan ukuran media penyimpanan dari laptop yang digunakan. Dengan besarnya file-file dokumentasi dan disimpan tidak pada satu tempat yang sama melainkan berbeda di tiap-tiap bidang. Hal ini berdampak pada sulitnya pegawai untuk mendapatkan file dokumentasi yang dibutuhkan dengan cepat karena file-file dokumentasi yang dibutuhkan tersebar ke berbagai penanggung jawab yang berbeda. Ditambah lagi untuk mendapatkan file dokumentasi, pegawai yang membutuhkannya perlu

membuat janji dengan penanggung jawab yang bersangkutan dikarenakan jika penanggung jawab tidak hadir maka tidak ada pilihan lain selain menunggu sampai penanggung jawab untuk hadir karena file dokumentasi yang diperlukan tersimpan pada laptop pribadi milik penanggung jawab.

Maka dari itu perlu adanya sebuah aplikasi yang dapat membuat album agar file-file dokumentasi ini tidak terpisah, dapat disimpan pada satu tempat yang sama, dapat diakses dengan cepat dan bersifat *online* serta aplikasi dibangun dengan memisahkan bagian *frontend* dan *backend*. “*Hands-On Nuxt.js Web Development: Build Universal and Static-Generated Vue.js Applications Using Nuxt.js*” (Kok, 2020), yang mana pada buku ini menjelaskan tentang bagaimana cara membangun *frontend* aplikasi dengan menggunakan *framework* Nuxt dan *backend* aplikasi yang mengimplementasikan arsitektur RESTful API menggunakan *framework* Express. Aplikasi yang dibangun dengan mengimplementasikan arsitektur RESTful API memiliki beberapa manfaat, yaitu: dapat menskalakan aplikasi secara efisien, pemisahan total klien-server, dan independen terhadap teknologi yang digunakan. Dengan demikian performa dari aplikasi tidak akan menurun meski dengan data-data yang tersimpan pada aplikasi terus bertambah besar.

Berdasarkan latar belakang yang ada maka penulis memilih judul *Implementasi Arsitektur Restful Web Service Menggunakan Framework Express Pada Aplikasi Arsip Media Berbasis NUXT*. Diharapkan dengan adanya aplikasi tersebut dapat membantu Dinas Kesehatan Provinsi Sulawesi Selatan dalam melakukan arsip media dokumentasi kegiatan dan memudahkan pegawai pada saat ingin mengakses ataupun berbagi file-file dokumentasi kepada sesama pegawai.

1.2 Rumusan Masalah

Berdasarkan pada latar belakang masalah yang telah dikemukakan di atas maka rumusan permasalahan dalam penelitian ini adalah:

1. Bagaimana pengarsipan media dapat diimplementasikan dalam sebuah aplikasi.
2. Bagaimana penerapan *backend* aplikasi dengan arsitektur RESTful web *service* menggunakan *framework* express yang dapat dikonsumsi *frontend*.
3. Bagaimana penerapan *frontend* aplikasi menggunakan *framework* nuxt yang dapat terhubung dengan *backend*.

1.3 Tujuan Penelitian

1. Merancang aplikasi arsip media berbasis web yang dapat digunakan untuk menyimpan dan berbagi media secara *online*.
2. Merancang *backend* aplikasi dengan arsitektur RESTful web *service* menggunakan *framework* express yang dapat menerima permintaan dari *frontend* dan kemudian memberikan jawaban yang sesuai.
3. Merancang *frontend* aplikasi menggunakan *framework* nuxt yang dapat mengkonsumsi data dari jawaban atas permintaan yang dikirimkan kepada *backend*.

1.4 Manfaat Penelitian

1. Terhadap akademik, dapat menjadi acuan dalam melakukan penelitian selanjutnya yang relevan dan searah dengan penelitian ini.

2. Terhadap tempat penelitian, dari hasil penelitian ini diharapkan dengan adanya aplikasi tersebut dapat membantu pegawai Dinas Kesehatan Provinsi Sulawesi Selatan dalam melakukan arsip media dokumentasi kegiatan dan memudahkan pegawai pada saat ingin mengakses ataupun berbagi media-media dokumentasi kepada sesama pegawai.
3. Terhadap penulis, sebagai syarat untuk mendapatkan gelar sarjana pada jurusan Teknik Informatika Universitas Dipa Makassar.

1.5 Sistematika Penulisan

Sistematika penulisan mengenai pembahasan masalah dan penyelesaiannya akan diuraikan sebagai berikut :

BAB I : PENDAHULUAN

Bab ini membahas tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, dan manfaat penelitian, serta sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Bab ini membahas landasan teori yang digunakan meliputi kerangka pikir dan landasan teori.

BAB III : METODOLOGI PENELITIAN

Bab ini membahas mengenai waktu dan tempat penelitian, jenis penelitian, teknik pengumpulan data, alat dan bahan penelitian, alat desain dan pengujian sistem, serta tahap dan jadwal penelitian.

BAB IV : HASIL PENELITIAN DAN PEMBAHASAN

Bab ini membahas mengenai hasil penelitian serta pembahasan dari hasil penelitian yang terkait.

BAB V : KESIMPULAN DAN SARAN

Bab ini membahas kesimpulan dan saran.

DAFTAR PUSTAKA

BAB II

TINJAUAN PUSTAKA

2.1 Kerangka Pikir

Kerangka pikir penelitian ini dibuat dalam bentuk skema seperti pada Gambar 2.1 dibawa ini.

Dalam setiap kegiatan Dinas Kesehatan Provinsi Sulawesi Selatan yang diadakan oleh bidang tertentu, selalu ada pegawai yang bertugas mengambil gambar atau video untuk keperluan arsip.



Selama ini media dokumentasi disimpan pada staf yang berbeda di tiap-tiap bidang. Hal ini berdampak pada sulitnya pegawai untuk mendapatkan media dengan cepat karena media dokumentasi yang dibutuhkan tersebar ke berbagai penanggung jawab.



Membangun aplikasi yang bisa dikelompokkan berdasarkan album agar media-media dokumentasi ini tidak terpisah, dapat disimpan pada satu tempat yang sama, dapat diakses dengan cepat dan bersifat *online*.



Diharapkan dengan adanya aplikasi tersebut dapat memudahkan pegawai dalam mengakses media-media yang telah diarsipkan dengan cepat dan sesuai dengan kebutuhan.

Gambar 2.1 Kerangka Pikir

2.2 Landasan Teori

2.2.1 HTTP

“*Hypertext Transfer Protocol* (HTTP) adalah protokol jaringan lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif, dan hypermedia. Penggunaannya terutama dalam mengambil sumber daya yang saling berhubungan melalui tautan yang disebut dokumen hypertext. Dokumen hypertext kemudian membentuk World Wide Web pada tahun 1990 oleh fisikawan Inggris Tim Berners-Lee. Sampai saat ini, protokol HTTP memiliki dua versi utama. HTTP/1.0 menggunakan koneksi terpisah untuk setiap dokumen, dan HTTP/1.1 memungkinkan transaksi menggunakan koneksi yang sama, membuang waktu untuk mempertahankan koneksi. Namun, baik HTTP/1.0 dan HTTP/1.1 memiliki kerentanan keamanan. Tidak ada versi HTTP yang menjamin keamanan komunikasi data antara pengguna dan situs web. Hal ini berbahaya bila diterapkan pada website e-commerce, bank, website resmi pemerintah atau lembaga atau organisasi. Masalah utama dengan protokol HTTP adalah bahwa proses pengiriman permintaan HTTP dan tanggapan HTTP berjalan tanpa keamanan apa pun. Jadi siapa pun yang memiliki akses ke jaringan dapat mencegat informasi yang dikirim dan bahkan merusak data” (Prayama dkk., 2021)

2.2.2 JSON

“ (JSON) adalah format pertukaran data sederhana, mudah dibaca dan ditulis, mudah dikompilasi dan dihasilkan. Format ini sebagian didasarkan pada bahasa pemrograman JavaScript. JSON merupakan format tekstual yang merupakan

bahasa pemrograman agnostik karena menggunakan bahasa yang biasa digunakan oleh programmer. Properti ini membuat JSON menjadi bahasa pertukaran data yang ideal. JSON memiliki beberapa keunggulan dibandingkan XML. Manfaatnya antara lain: (2) Kecepatan pertukaran data JSON lebih baik daripada XML karena JSON membutuhkan ukuran karakter yang lebih kecil dari XML untuk data yang sama. (3) JSON menggunakan JavaScript eval(), sehingga proses parsing untuk browser JSON lebih mudah daripada XML. Mengurai dokumen XML saat sedang diurai oleh XMLHttpRequest” (Somya & Nathanael, 2019)

2.2.3 API

“*Application Programming Interface* (API) adalah konsep fungsi yang dapat dipanggil oleh program lain. API bertindak sebagai perekat yang menyatukan berbagai aplikasi pada berbagai jenis platform, umumnya dikenal sebagai API publik. API publik banyak digunakan, tetapi orang yang ingin menemukannya, yaitu programmer, harus mencari langsung melalui berbagai metode seperti mesin pencari publik, dokumentasi repositori, dan artikel web. Pengguna kesulitan menemukan tautan API publik karena mereka tidak memiliki sistem khusus untuk mengumpulkan publik API. Solusi untuk masalah ini dapat diselesaikan dengan membuat kerangka kerja web dengan antarmuka mesin pencari yang menyediakan pencarian khusus publik API. Hal ini memudahkan pengguna untuk menemukan API publik. Layanan web adalah API yang dibuat untuk mendukung interaksi antara dua atau lebih aplikasi yang berbeda melalui jaringan. *Representational State Transfer* (REST) adalah salah satu aturannya.” (Muri dkk., 2019)

2.2.4 RESTful API

“Salah satu jenis web service adalah REST atau RESTful (Representational State Transfer). REST sendiri memungkinkan untuk mengakses dan memanipulasi teks yang diwakili oleh layanan web untuk persyaratan sistem. API layanan web yang menggunakan REST disebut RESTful API. Karena REST, tidak seperti jenis layanan web lainnya, memiliki arsitektur, tidak ada standar formal untuk mewakili RESTful API. Dengan berbagai implementasi, REST menemukan notasi yang umum digunakan seperti HTTP, URI, JSON, dan XML. REST API telah terbukti lebih cepat untuk transfer data daripada metode serupa lainnya, dalam hal ini SOAP (Simple Object Access Protocol). Tidak seperti SOAP, RESTful API tidak memiliki notasi standar formal. Ini karena SOAP adalah protokol sedangkan REST adalah arsitektur. Sepanjang jalan, pengembangan untuk mengembangkan RESTful API mengalami masalah dengan penamaan objek. Notasi objek yang ada seharusnya membuat penamaan lebih mudah bagi pengembang, tetapi juga harus dipahami oleh komputer. Untuk alasan ini, pengembang menggunakan JSON (JavaScript Object Notation) sebagai representasi dari layanan web REST dalam aplikasi mereka. Menggunakan JSON untuk mendukung pembuatan aplikasi seluler berbasis REST API juga dianggap lebih kuat untuk pengguna seluler.”
(Amin dkk., 2017)

2.2.5 JavaScript

“Sebagian besar situs web menggunakan JavaScript untuk meningkatkan kegunaan dan fungsionalitas aplikasi web. Seiring dengan Hypertext Markup Language (HTML) dan Cascading Style Sheets (CSS), bahasa pemrograman

JavaScript adalah salah satu dari tiga teknologi dasar pengembangan web. Penggunaan JavaScript telah menjadi standar di antara semua pengembang web karena mudah digunakan dan memungkinkan Anda membuat halaman web yang dinamis dan interaktif. Menurut penelitian, 97,7% dari semua situs web menggunakan JavaScript sebagai bahasa pemrograman sisi klien mereka. Kode JavaScript ditafsirkan oleh browser web pengguna dan dieksekusi pada prosesor pengguna, bukan server web. Ini memungkinkan Anda untuk berinteraksi dengan Model Objek Dokumen (DOM) halaman Web dan menambahkan perilaku situs klien ke halaman HTML Anda. Contoh penggunaan ini termasuk animasi objek, validasi input pengguna, dan komunikasi asinkron. Selain lingkungan berbasis web, JavaScript digunakan di lingkungan seperti dokumen Portable Document Format (PDF), browser khusus situs, dan widget desktop.” (Alazab dkk., 2022)

2.2.6 Node.JS

“Dari prototipe pertama JavaScript pada tahun 1995 dan hingga 2009, JavaScript sebagian besar digunakan untuk aplikasi web sisi klien, tetapi ini berubah dengan diperkenalkannya Node.js. Node.js menyediakan lingkungan runtime yang memungkinkan server web diimplementasikan dalam JavaScript. Node.js mengandalkan mesin JavaScript V8 dan menggabungkannya dengan loop peristiwa dan API I/O tingkat rendah yang ditulis dalam C++. Pilihan desain ini memungkinkan interaksi dengan mesin yang mendasarinya, sistem filenya, sumber daya sistem operasi, dan database, sedangkan JavaScript di browser dijalankan dalam kotak pasir. Node.js sejak diperkenalkan pada tahun 2009 meningkat secara signifikan dalam popularitas, dan dalam survei Stack Overflow dari tahun 2019,

Node.js adalah framework, library, atau alat yang paling umum digunakan, dengan sekitar 50% dari 58.543 responden menggunakan Node.js. Bagian penting dari ekosistem Node.js adalah penggunaan kembali paket ekstensif yang difasilitasi oleh manajer paket npm. Informasi meta aplikasi. B. Dependensi yang dimiliki aplikasi Anda disediakan melalui file package.json.” (Benjamin dkk., 2019)

2.2.7 Express.JS

“Express.JS adalah kerangka kerja Node.JS yang membantu Anda dengan mudah membangun aplikasi berbasis Node.JS menggunakan pola desain yang dapat disesuaikan dan sangat fleksibel. Selain itu, Express.JS juga merupakan framework yang sangat ringan, cocok untuk membangun aplikasi web dan API.” (Rachmat, 2017)

2.2.8 MYSQL

"MySQL adalah perangkat lunak sistem manajemen basis data atau DBMS, multi-utas dan multi-pengguna, dengan sekitar 6 juta instalasi di seluruh dunia". Kami menyediakannya sebagai perangkat lunak gratis dibawah Lisensi Publik (GPL), tetapi jual di bawah lisensi komersial jika tidak sesuai dengan penggunaan GPL. Tidak seperti Apache, yang merupakan perangkat lunak yang dikembangkan untuk umum yang kode sumbernya dilindungi hak cipta oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh perusahaan perdagangan Swedia MySQL AB. MySQL AB mempertahankan hak cipta pada sebagian besar kode sumbernya. Dua orang yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson dan Michael "Monty" Widenius.” (Solichin, 2017)

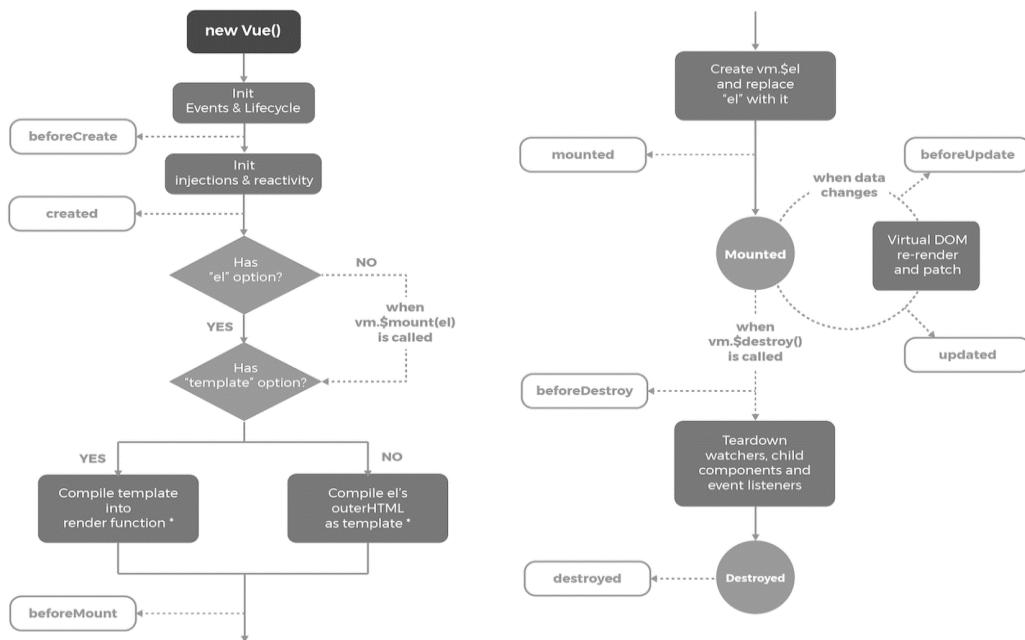
Berikut adalah beberapa manfaat MySQL:

- a. Gratis (bebas di download)
- b. Stabilitas dan ketahanan
- c. Fleksibilitas melalui berbagai pemrograman
- d. Keamanan yang baik
- e. Dukungan komunitas besar-besaran
- f. Manajemen database yang mudah.
- g. Mendukung Transaksi
- h. Pengembangan perangkat lunak sangat cepat.

2.2.9 Vue.JS

“Vue js adalah kerangka kerja JavaScript yang baik untuk pengembangan aplikasi dibandingkan dengan kerangka kerja JavaScript lain seperti Angular dan React. dalam tiga kerangka. Secara sintaksis, Angular paling sulit dipelajari karena menggunakan TypeScript. Pengguna baru harus mempelajari tidak hanya kerangka kerja baru, tetapi juga superset baru JavaScript. React dan Vue js dekat dalam hal kurva belajar. Sintaks Vue adalah HTML yang valid, mirip dengan HTML, CSS, dan Javascript yang disematkan di halaman web tradisional, sehingga dapat dianggap lebih sederhana daripada React. Bereaksi menggunakan sintaks JSX. Sintaks JSX lebih mudah dipelajari bagi mereka yang memiliki dasar yang kuat dalam HTML dan JavaScript, tetapi bisa sedikit lebih sulit digunakan daripada sintaks Vue. Vue juga mengungguli Angular dan React di semua pengujian, dan 5x lebih cepat dari Angular dan 2x lebih cepat dari React secara keseluruhan. React sekitar 2,5x lebih cepat dari Angular.” (Adivia & Ari, 2020)

Vue JS adalah kerangka kerja untuk mengelola dan menyederhanakan pengembangan aplikasi web. Komponen adalah salah satu fitur Vue yang paling kuat. Untuk aplikasi besar, seluruh aplikasi harus dibagi menjadi komponen yang lebih kecil, mandiri, dan dapat digunakan kembali untuk membuat pengembangan lebih mudah dikelola.



Gambar 2.2 *lifecycle hook* Vue.JS

Pada gambar 2.2 menunjukkan bagaimana cara Vue JS bekerja atau yang dikenal juga sebagai *lifecycle hook*. *Lifecycle hook* adalah bagian yang sangat penting untuk diketahui, karena mereka menentukan langkah dimana Vue JS mengeksekusi kode Anda. Berikut ini adalah arti dari gambar 2.2:

- beforeCreate*: dieksekusi pertama kali pada saat sebuah komponen sebelum semuanya diinisiasi, ketika data dan event belum diinisiasi oleh Vue
- created*: dieksekusi oleh vue ketika data dan event telah selesai di-render, tetapi template belum di-render oleh vue.

- c. beforeMount: dieksekusi oleh Vue sebelum template pada sebuah component di-render tetapi setelah template di compile oleh vue untuk mengakses DOM.
- d. mounted: dieksekusi oleh vue setelah template di-render.
- e. beforeUpdate: dieksekusi oleh vue ketika stage data sudah diubah pada sebuah component, tepat sebelum DOM merender ulang template.
- f. updated: dieksekusi oleh Vue ketika data berhasil diubah dan DOM berhasil dirender kembali.
- g. beforeDestroy: dieksekusi tepat sebelum dihapus. Komponen tetap ada dan berfungsi sepenuhnya.
- h. destroyed: dieksekusi saat sudah dihapus, tidak ada yang tersisa di komponen. Segala sesuatu yang melekat padanya telah dihapus.

2.2.10 Nuxt.JS

“Nuxt adalah kerangka kerja pengembangan web Node.js tingkat tinggi untuk membuat aplikasi Vue yang dapat dikembangkan dan digunakan dalam dua mode berbeda: universal (SSR) atau aplikasi satu halaman (SPA). Selanjutnya, Anda dapat menggunakan SSR dan SPA di Nuxt sebagai aplikasi yang dibuat secara statis. Meskipun Anda dapat memilih mode SPA, kekuatan penuh Nuxt terletak pada mode universal atau rendering sisi server (SSR) untuk membangun aplikasi universal. Aplikasi universal digunakan untuk mendeskripsikan kode JavaScript yang dapat dijalankan baik di klien maupun di sisi server. Tetapi jika Anda ingin mengembangkan SPA klasik (atau standar/tradisional), yang hanya dijalankan di sisi klien, Anda mungkin ingin mempertimbangkan untuk menggunakan vanilla Vue.” (Lau, 2020)

Nuxt dibuat di atas Vue, dilengkapi dengan beberapa fitur tambahan seperti data asinkron, middleware, tata letak, modul, dan plugin yang menjalankan aplikasi Anda di sisi server terlebih dahulu, lalu di sisi klien. Ini berarti aplikasi umumnya merender lebih cepat daripada aplikasi sisi server (atau beberapa halaman) tradisional.

Nuxt sudah diinstal sebelumnya dengan paket-paket berikut sehingga Anda tidak perlu menginstalnya, yang akan Anda lakukan di aplikasi Vue standar:

- a. Vue
- b. Vue Router
- c. Vuex
- d. Vue Server Renderer
- e. Vue Meta

Selain itu, Nuxt menggunakan webpack dan Babel untuk mengkompilasi dan menggabungkan kode Anda dengan pemuat webpack berikut:

- a. Vue loader
- b. Babel loader

Singkatnya, webpack adalah bundler modul yang menggabungkan semua skrip, gaya, aset, dan gambar di aplikasi JavaScript Anda, sementara Babel adalah kompiler JavaScript yang mengkompilasi atau mentransplasikan JavaScript generasi berikutnya (ES2015+) ke JavaScript yang kompatibel dengan browser (ES5) sehingga Anda dapat menjalankan kode Anda di browser saat ini. Webpack menggunakan apa yang mereka sebut loader untuk melakukan praproses file Anda saat Anda mengimportnya melalui pernyataan impor JavaScript atau metode

require. Anda dapat menulis pemuat Anda tetapi Anda tidak perlu melakukannya saat mengkompilasi kode Anda dalam file Vue karena telah dibuat untuk Anda oleh komunitas Babel dan tim Vue.

2.2.11 *Unified Modeling Language (UML)*

Menurut (Muslihudin dkk., 2016), UML (*Unified Modeling Language*) yang berarti bahwa bahasa pemodelan UML standar bukan hanya diagram, tetapi juga mewakili konteks. *Unified modeling language* (UML) termasuk simbol dan tata bahasa yang menjelaskan bagaimana simbol ini digunakan.

Dengan mempelajari simbol dan tata bahasa, siapa pun dapat memahami gambaran materi model sebelumnya. Beberapa bagian:

1. *Use Case Diagram*

Menurut Munawar (2018), “*Use case* diagram adalah jenis diagram di UML yang menggambarkan interaksi antara sistem dan aktornya. *Use case* diagram bekerja dengan menggambarkan interaksi tipikal antara pengguna sistem dengan sistemnya sendiri melalui cerita bagaimana sistem tersebut digunakan.”.

Selain itu, kasus penggunaan naratif memberikan informasi standar yang diperlukan untuk menganalisis, mendesain, dan membuat kode fitur ini. Jenis *use case* ini disebut cerita *use casenarrative*.

Tabel 2.1 Simbol *use case* diagram

No	Gambar	Nama	Keterangan
1		Actor	Menspesifikasi kan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case.

No	Gambar	Nama	Keterangan
2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent).
3		Generalization	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada diatasnya objek induk (<i>ancestor</i>).
4		Include	Menspesifikasi bahwa <i>use case</i> sumber secara Eksplisit
5		Extend	Mengspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang Diberikan
6		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya
7		Sistem	Mengspesifikasikan paket yang menampilkan sistem secara terbatas.
8		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu Actor
9		Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Sumber: Munawar (2018:09)

2. Activity Diagram

Menurut Munawar (2018), *Activity* diagram adalah bagian penting dari UML yang menggambarkan aspek dinamis dari suatu sistem. *Activity* diagram memiliki peran yang mirip dengan *flowchart*, tetapi perbedaannya dengan *flowchart* adalah *Activity* diagram dapat mendukung perilaku paralel, sedangkan *flowchart* tidak bisa.

Activity diagram adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan alur kerja dalam banyak kasus. *Activity* diagram memiliki peran yang mirip dengan *flowchart*, tetapi perbedaannya dengan *flowchart* adalah *activity* diagram dapat mendukung perilaku paralel, sedangkan *flowchart* tidak bisa.

Tabel 2.2 Simbol *Activity* Diagram

No	Gambar	Nama	Keterangan
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari satu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali
4		Activity Final Node	Bagaimana objek dibentuk dan dihancurkan
5		Fork Node	Percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk mengabungkan dua kegiatan paralel menjadi satu

Sumber: Munawar (2018:09)

3. Sequence Diagram

Menurut Munawar (2018), *Sequence diagram* adalah salah satu jenis diagram pada UML yang menjelaskan interaksi obyek yang berdasarkan urutan waktu. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini di dalam *use case*.

Sequence Diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu.

Tabel 2.3 Simbol *Sequence Diagram*

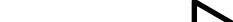
No	Gambar	Nama	Keterangan
1		<i>LifeLine</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah <i>object</i> .
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

Sumber: Munawar (2018:09)

4. Class Diagram

Menurut Sri Mulyani (2016), *Class diagram* adalah diagram yang digunakan untuk mempresentasikan kelas, komponen-komponen kelas dan hubungan-hubungan antar kelas. *Class diagram* menggambarkan struktur dan deskripsi class, package, dan objek beserta hubungan atau satu sama lain seperti *containment*, pewarisan, asosiasi dan lain-lain. Kelas memiliki tugas bagian utama yaitu *attribute*, *operation*, dan *name*.

Tabel 2.4 Simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1		Asosiasi	Relasi antara kelas dengan makna umum
2		Asosiasi Berarah	Relasi antar kelas dengan makna kelas yang digunakan oleh kelas lain
3		Generalisasi	Relasi antar kelas dengan generalisasi spesialisasi.
4		Kebergantungan	Relasi antar kelas dengan makna kebergantungan suatu kelas.
5		Agregasi	Relasi antar kelas dengan makna semua bagian

Sumber : Sri Mulyani (2016:249)

2.3 Penelitian Terkait

Dalam penyusunan skripsi ini, terdapat beberapa penilitian terkait yang digunakan oleh penyusun, sebagai berikut:

1. JSON Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service oleh Rohmat Gunawan dan Alam Rahmatulloh. Jurnal Edukasidan Penelitian Informatika Vol. 5 No. 1, 2019. Yang mana pada penelitian ini menyimpan nama pengguna dan kata sandi pada token JWT. Sedangkan pada penelitian kami, kami hanya menyimpan id pengguna sebagai identitas pada token JWT yang diotentikasi oleh sistem.

2. Penerapan Teknologi Single Page Application (SPA) pada Sistem Surat Disposisi Online (Sudion) di Universitas Muhammadiyah Prof Dr Hamka oleh Muhammad Fathan Aulia dan Arry Avorizano. Jurnal Teknik Informatika dan Komputer Vol. 1 No. 1, 2022. Yang mana pada penelitian ini menggunakan *framework* turbolinks untuk mempercepat pengoperasian browser dengan mencegah loop saat memuat sumber daya. Sedangkan pada penelitian kami, kami akan menggunakan *framework* Vue.js yang mana memiliki fungsi yang sedikit mirip dengan *framework* turbolinks namun lebih efisien dalam membuat website yang dinamis karena dalam pembuatannya menggunakan JavaScript.
3. Implementasi Autentikasi dari sisi Backend pada Arsitektur Microservices Menggunakan Express Js oleh Regita Lisgiani dan Sigit Nurmajid. Jurnal Infotronik Vol. 7 No. 1, 2022. Yang mana pada penelitian ini membuat sebuah server yang fungsinya khusus untuk mengelola otentikasi pengguna yang bisa diakses oleh frontend dan server lain. Sedangkan pada penelitian kami, kami hanya membuat satu buah server yang nantinya akan menelolah otentikasi, otorisasi, dan menejemen basis data.

BAB III

METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian dilakukan di Dinas Kesehatan Provinsi Sulawesi Selatan. Jl. Perintis Kemerdekaan Km. 11 Tamalanrea Tlp. (0411) 581718 Fax. (0411) 586451. Penelitian ini berlangsung selama 3 bulan mulai dari bulan September 2022 sampai dengan November 2022.

3.2 Jenis Penelitian

Dalam menyelesaikan skripsi ini kami menggunakan jenis penelitian lapangan (*Field Research*), yaitu penelitian dilakukan dengan cara mengamati aktivitas-aktivitas pengolahan data pada objek yang akan diteliti.

3.3 Metode Pengumpulan Data

Pada kegiatan penelitian ini, Penulis menggunakan beberapa metode yang dijadikan sebagai cara pengumpulan data yang dibutuhkan, yaitu:

1. Observasi

Penelitian dilakukan dengan cara meninjau langsung lokasi penelitian dan mengumpulkan data-data yang dibutuhkan sebagai variabel utama.

2. Wawancara

Penelitian dilakukan dengan cara tanya jawab dengan salah satu pegawai yang bertanggung jawab untuk melakukan arsip terhadap media-media dokumentasi.

3.4 Bahan dan Alat Penelitian

3.4.1 Bahan Penelitian

Tabel 3.1 Bahan Penelitian

No.	Bahan penelitian	Keterangan
1.	Gambar kegiatan	File jpg,jpeg,png
2.	Vidio kegiatan	File mp4,mov

3.4.2 Alat Penelitian

1. Spesifikasi laptop yang digunakan yaitu:
 - a. Processor Intel core i5
 - b. Harddisk 1048 Gb
 - c. RAM DDR4 8Gb
2. Perangkat lunak yang digunakan yaitu:

Tabel 3.2 Perangkat lunak yang digunakan

No.	Perangkat Lunak	Unit	Spesifikasi
1.	Sistem operasi	1	<i>windows 10</i>
2.	Bahasa Pemrograman	1	<i>Javascript</i>
3.	Database	1	<i>MySQL</i>
4.	Editor	1	<i>Visual Studio Code</i>

3.5 Metode Pengujian

Penulis menggunakan salah satu metode pengujian perangkat lunak yang mendalam, metode pengujian *black box*, atau yang biasa dikenal dengan aliran logika, untuk menguji jalur logika perangkat lunak dengan menyediakan kasus uji yang beroperasi di bawah serangkaian kondisi atau *loop* tertentu.

Dengan menggunakan metode *black box*, penulis dapat menemukan kategori kesalahan berikut:

1. Fungsi-fungsi yang tidak benar atau salah
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau *database eksternal*
4. Kesalahan kinerja
5. Instalisasi dan kesalahan terminasi

Langkah-langkah pengujian dalam *Black box*:

1. Buat *test case* dari aplikasi.
2. Uji coba data setiap *form* dari aplikasi.
3. Jika hasil pengujian sesuai dengan yang diharapkan, maka dapat disimpulkan bahwa aplikasi berfungsi dengan baik (tidak ada kesalahan fungsional).

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

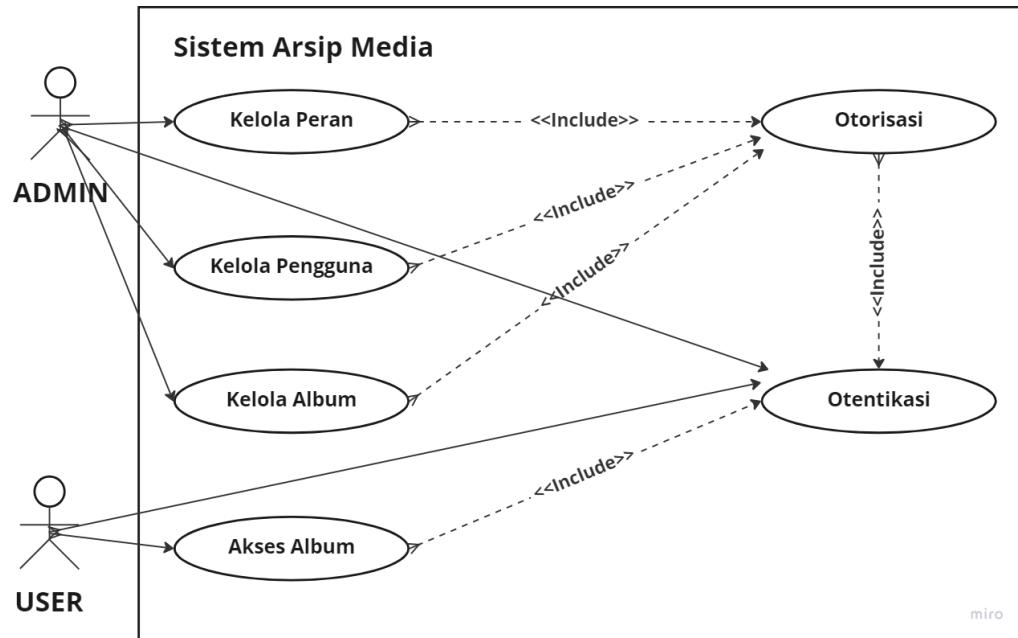
4.1 Perancangan Solusi

Rancangan desain harus dapat mengatasi permasalahan awal tanpa menimbulkan permasalahan lain. Rancangan merupakan suatu sistem kegiatan yang dilakukan untuk mendesain suatu sistem yang mempunyai tahapan-tahapan kerja yang tersusun secara logis, dimulai dari pengumpulan data yang diperlukan guna pelaksanaan perancangan tersebut. Langkah selanjutnya adalah menganalisis data yang telah dikumpulkan guna menentukan batasan-batasan sistem, kemudian melangkah lebih jauh lagi yakni merancang sistem tersebut.

Rancangan *Input* merupakan awal dimulainya proses informasi. Bahan baku informasi meliputi data dari transaksi yang dilakukan oleh organisasi. Formulir adalah alat penting untuk mengontrol alur kerja dan digunakan untuk mengumpulkan data saat itu terjadi. Ini sering disebut sebagai dokumen sumber. Rancangan *Output* merupakan produk dari sistem informasi yang dapat dilihat.

4.1.1 Use Case Diagram

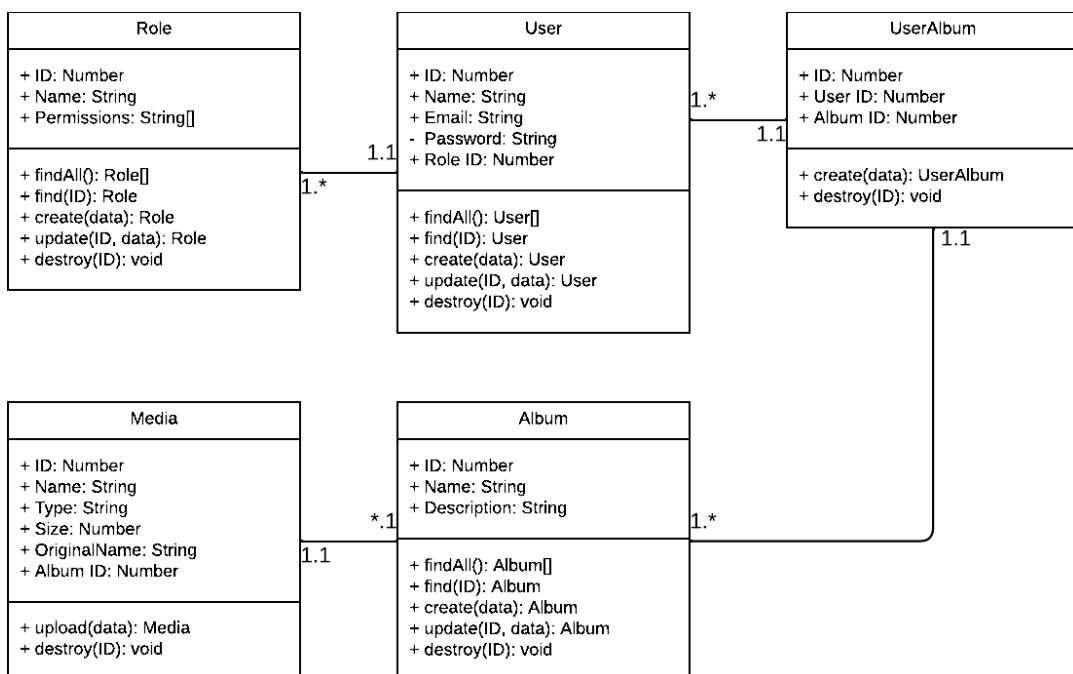
Pada gambar 4.1 *Use Case Diagram* di bawah menunjukkan aplikasi yang digunakan oleh 2 pengguna yaitu admin dan user. Admin dapat mengelola data user, data peran, data album dan memberikan akses album kepada user. Sedangkan user hanya dapat mengakses album yang telah diberikan izin oleh admin.



Gambar 4.1 Use Case Diagram

4.1.2 Class Diagram

Bentuk perancangan *class diagram* dari aplikasi arsip media dapat dilihat pada gambar 4.2 berikut.

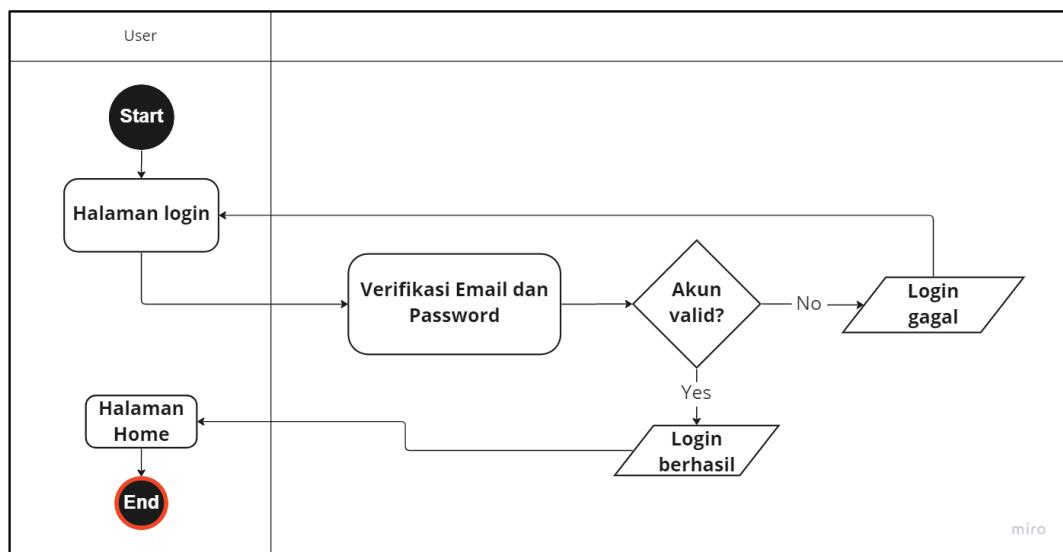


Gambar 4.2 Class Diagram

4.1.3 Activity Diagram

Activity diagram digunakan untuk menggambarkan rangkaian aliran dari aktivitas. *Activity* diagram juga digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti *use case* atau interaksi.

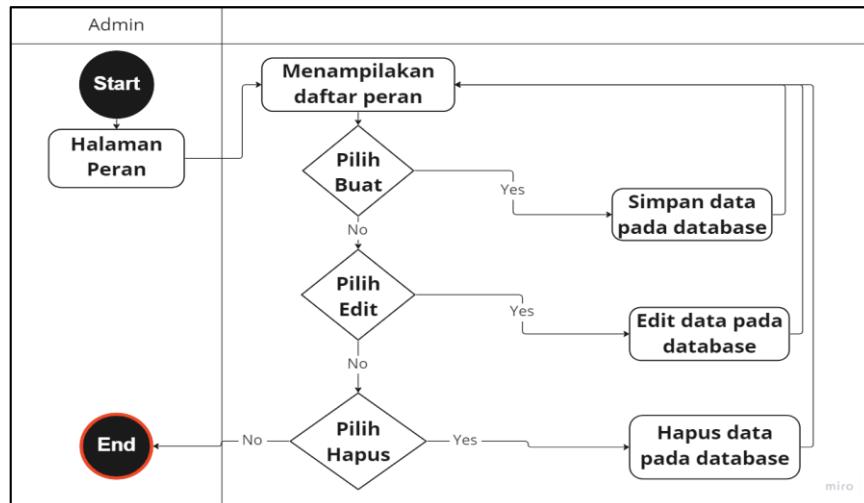
1. Activity Diagram Login



Gambar 4.3 Activity Diagram Login

Pada gambar 4.3 diatas merupakan aktivitas untuk melakukan login dengan cara menginput *email* dan *password* klik tombol masuk dan jika login berhasil aplikasi menampilkan halaman *home* namun jika login gagal aplikasi akan menampilkan kembali halaman login.

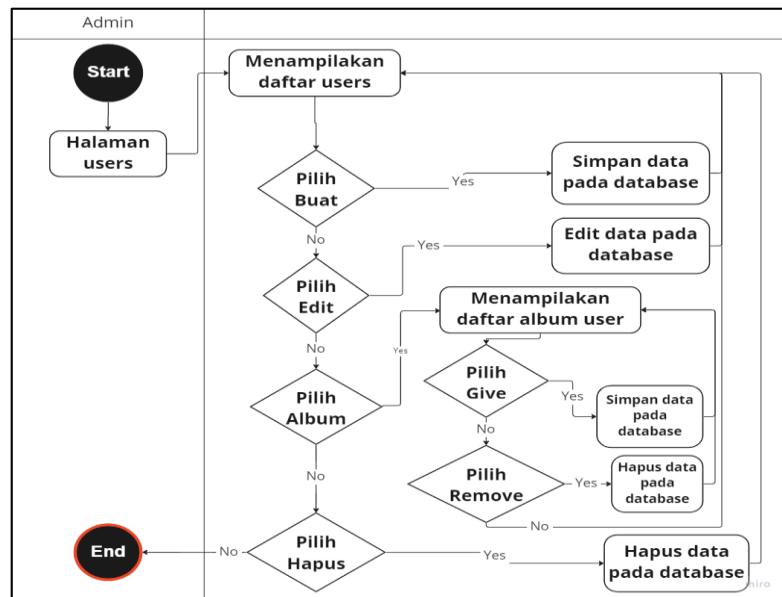
2. Activity Diagram Kelola Peran



Gambar 4.4 Activity Diagram Kelola Peran

Pada gambar 4.4 diatas merupakan aktivitas untuk menampilkan daftar peran, membuat peran, mengedit peran dan menghapus peran.

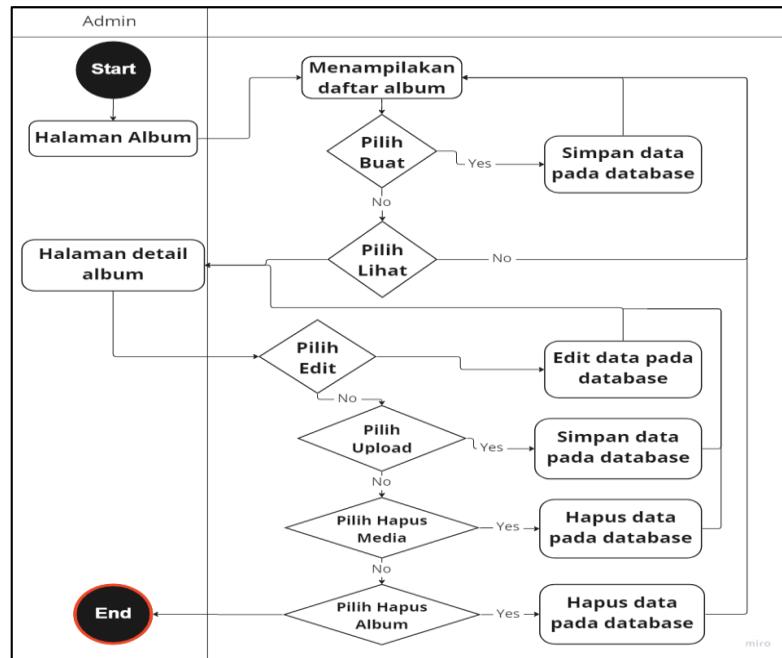
3. Activity Diagram Kelola Pengguna



Gambar 4.5 Activity Diagram Kelola Pengguna

Pada gambar 4.5 diatas merupakan aktivitas untuk menampilkan daftar user, membuat user, mengedit user, menghapus user dan memberikan akses album serta menghapus akses album.

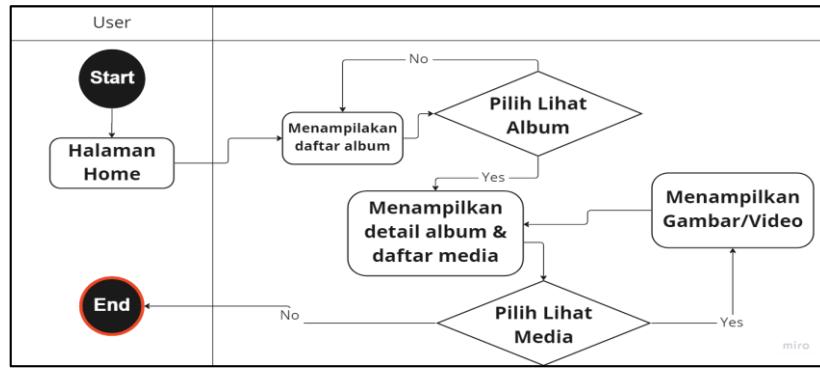
4. Activity Diagram Kelola Album



Gambar 4.6 Activity Diagram Kelola Album

Pada gambar 4.6 diatas merupakan aktivitas untuk menampilkan daftar album, membuat album, mengedit album, menghapus album dan menampilkan daftar media album serta menghapus media.

5. Activity Diagram Akses Album

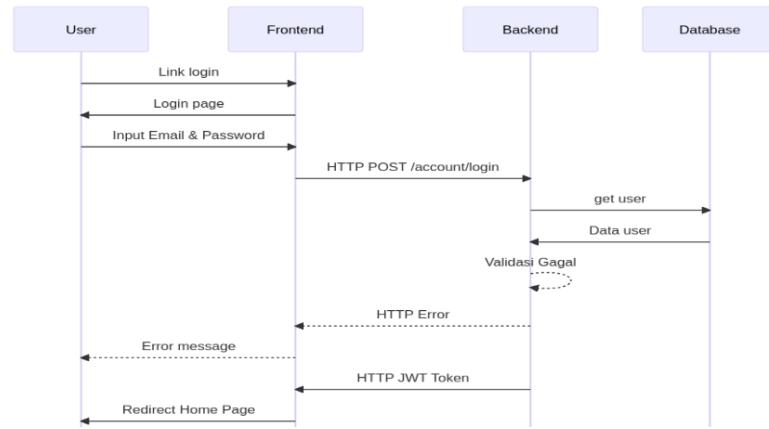


Gambar 4.7 Activity Diagram Akses Album

Pada gambar 4.7 diatas merupakan aktivitas untuk menampilkan daftar album yang bisa diakses oleh user.

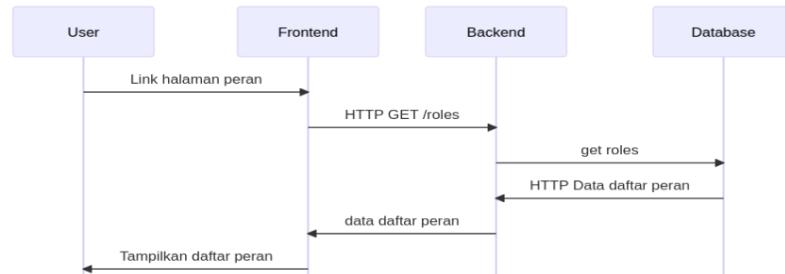
4.1.4 Sequence Diagram

1. Sequence Diagram Login

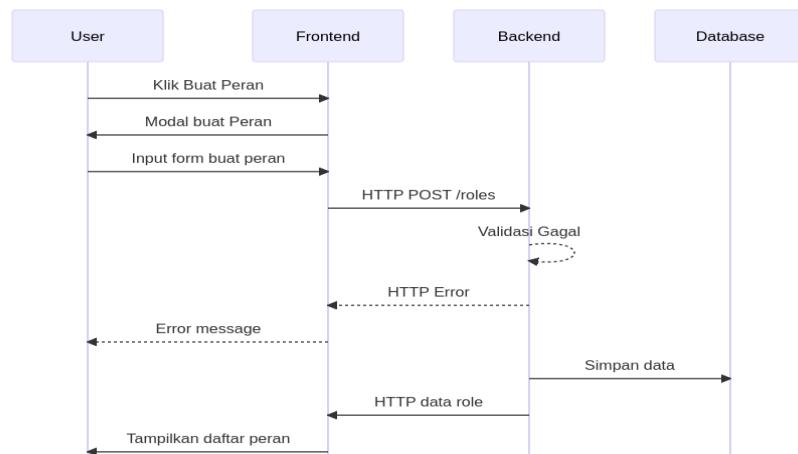


Gambar 4.8 Sequence diagram login

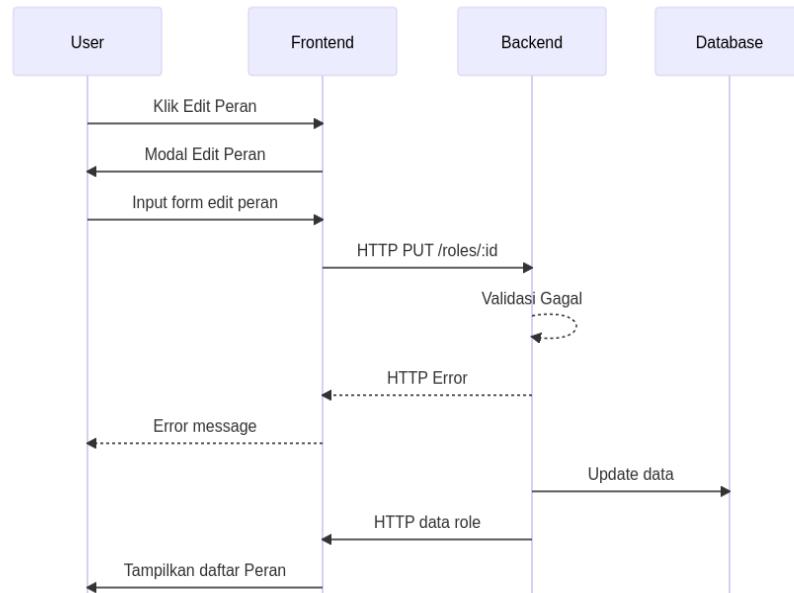
2. Sequence Diagram Kelola Peran



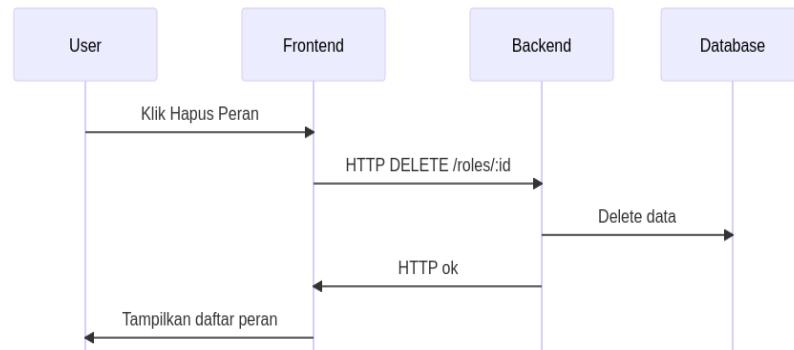
Gambar 4.9 Sequence diagram menampilkan daftar peran



Gambar 4.10 Sequence diagram membuat peran

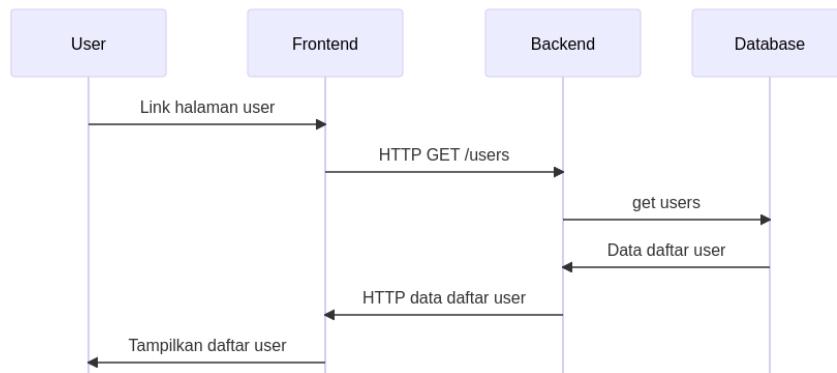


Gambar 4.11 Sequence diagram update peran

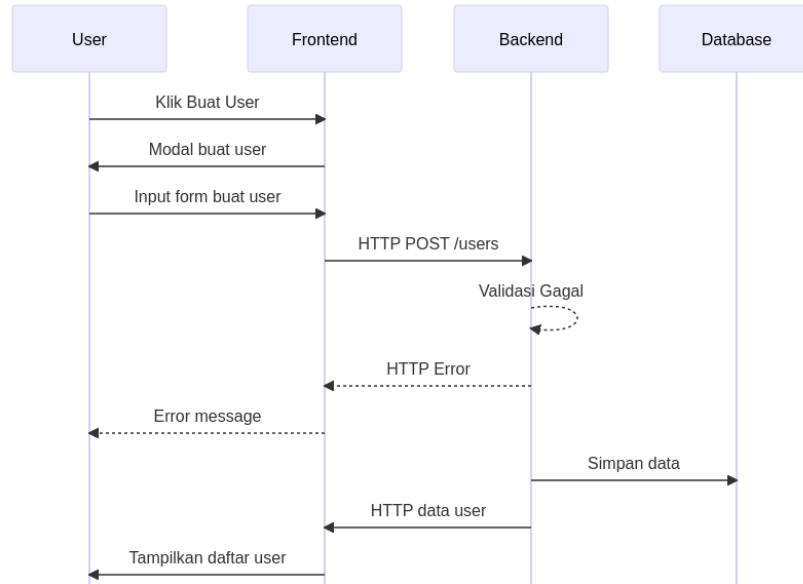


Gambar 4.12 Sequence diagram hapus peran

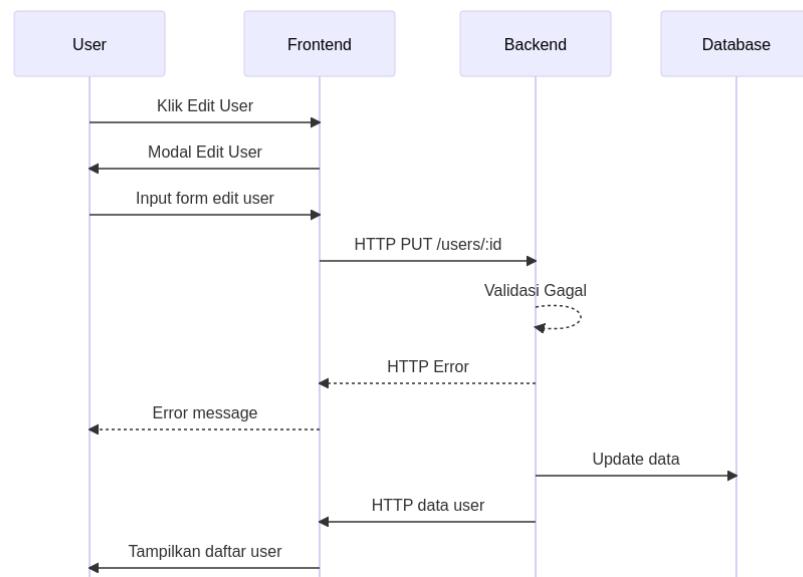
3. Sequence Diagram Kelola Pengguna



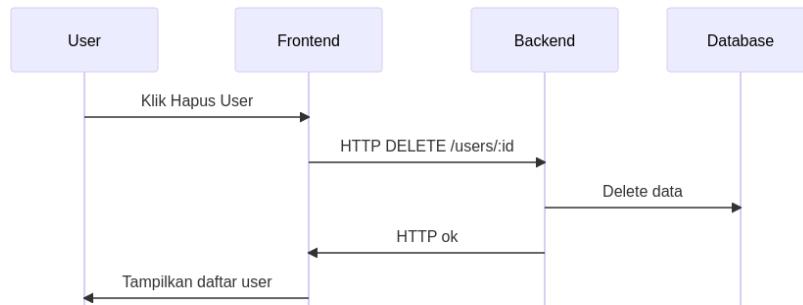
Gambar 4.13 Sequence diagram menampilkan daftar pengguna



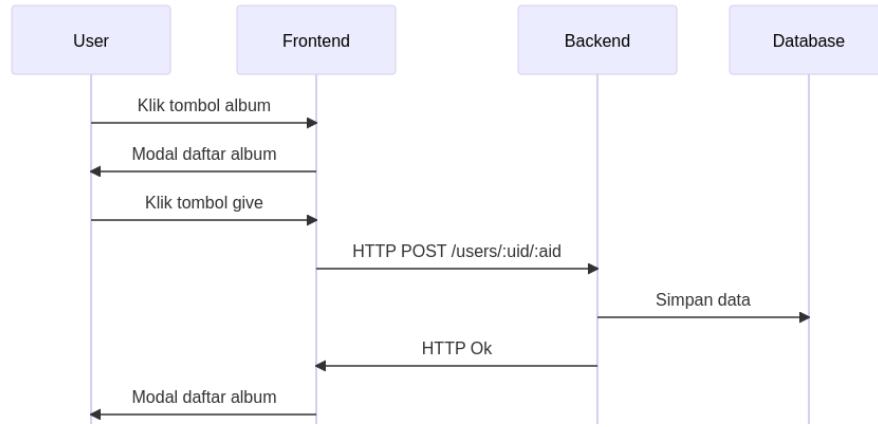
Gambar 4.14 Sequence diagram membuat pengguna



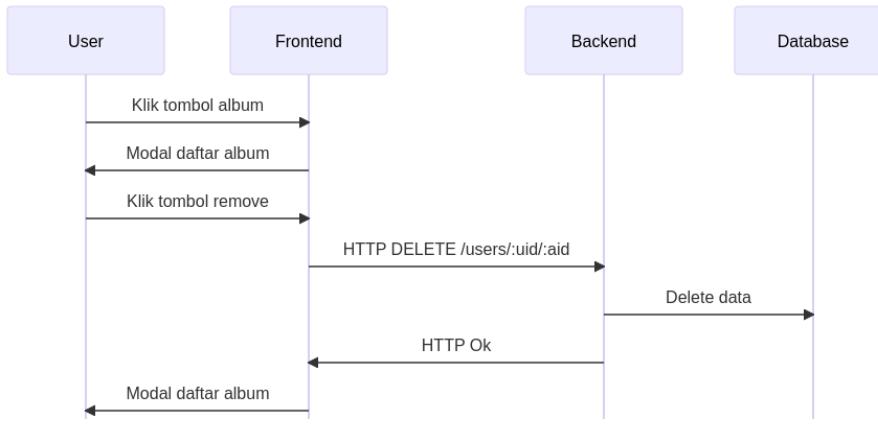
Gambar 4.15 Sequence diagram update pengguna



Gambar 4.16 Sequence diagram hapus pengguna

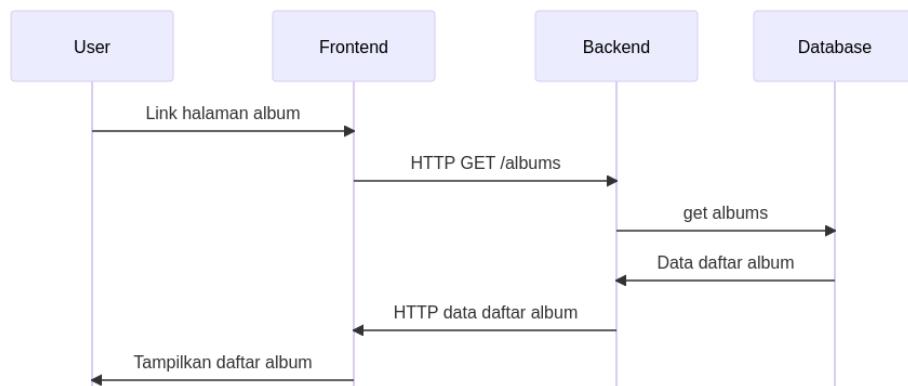


Gambar 4.17 Sequence diagram memberikan akses album pada pengguna

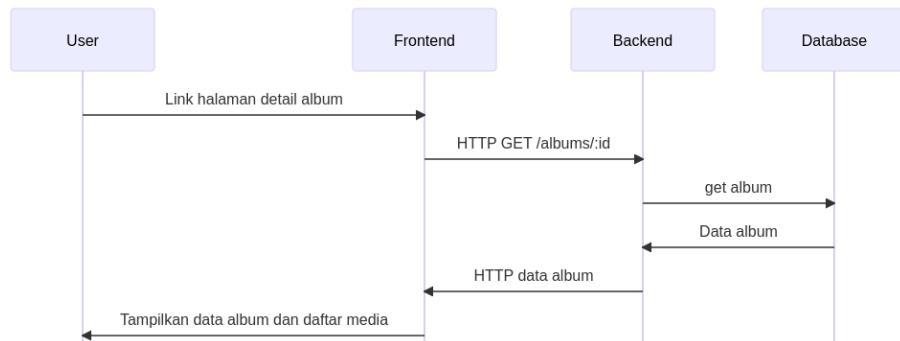


Gambar 4.18 Sequence diagram hapus akses album pengguna

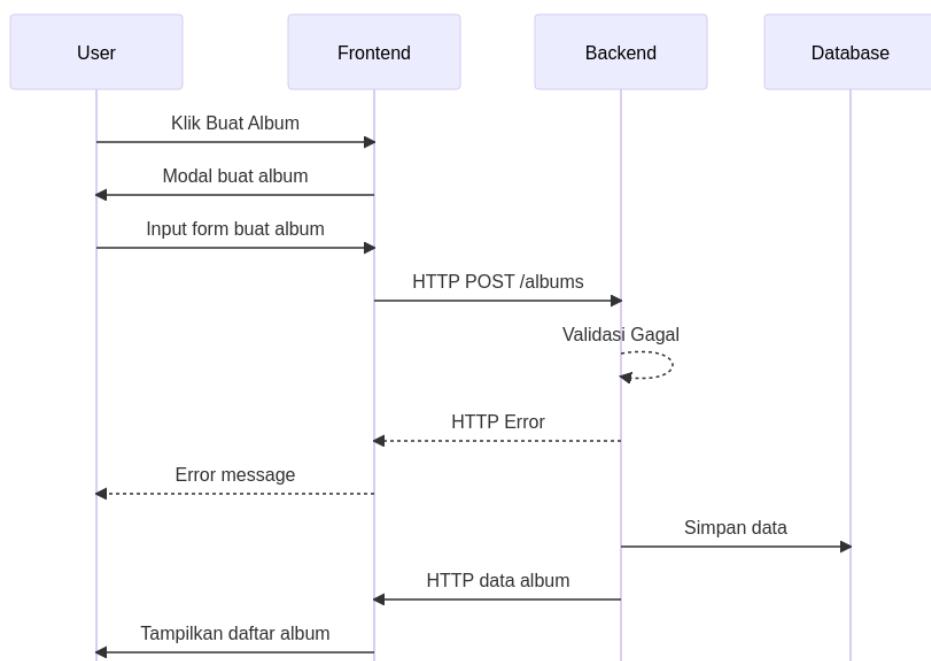
4. Sequence Diagram Kelola Album



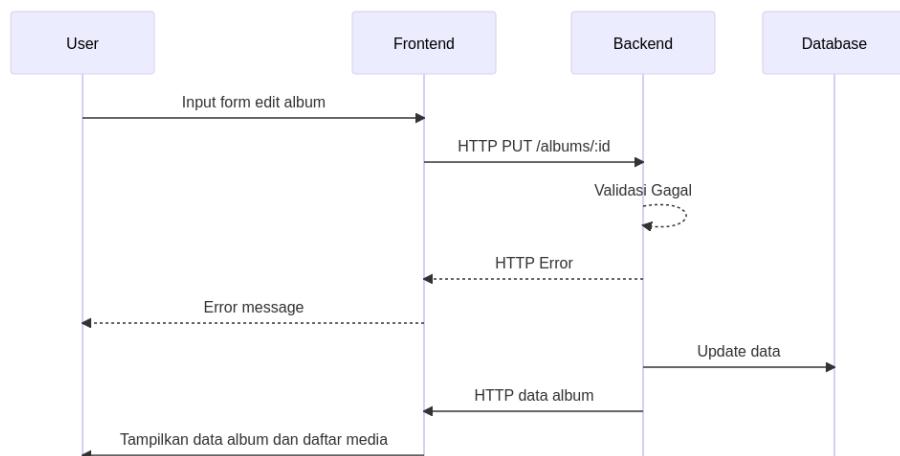
Gambar 4.19 Sequence diagram menampilkan daftar album



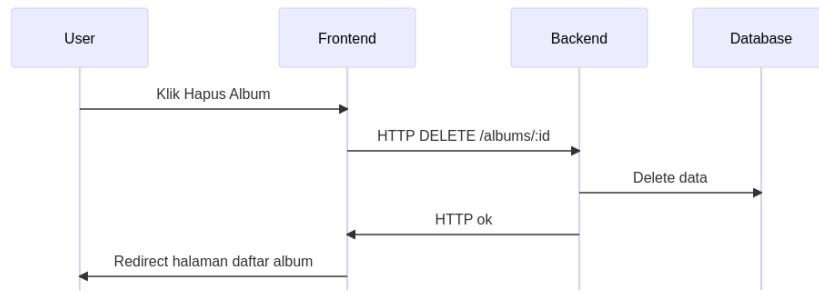
Gambar 4.20 Sequence diagram menampilkan detail album



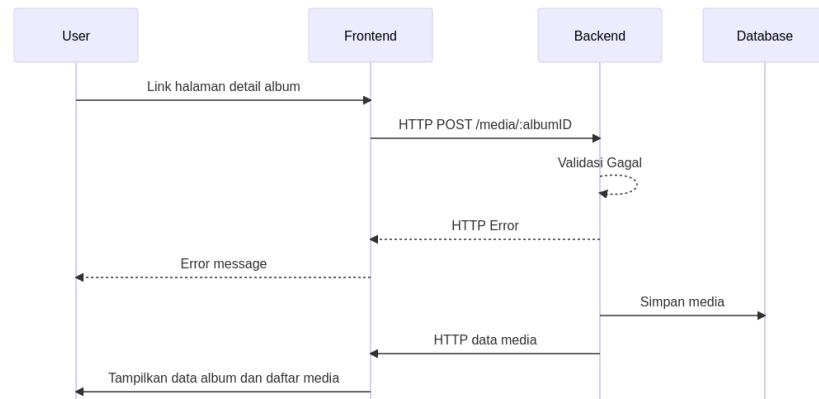
Gambar 4.21 Sequence diagram membuat album



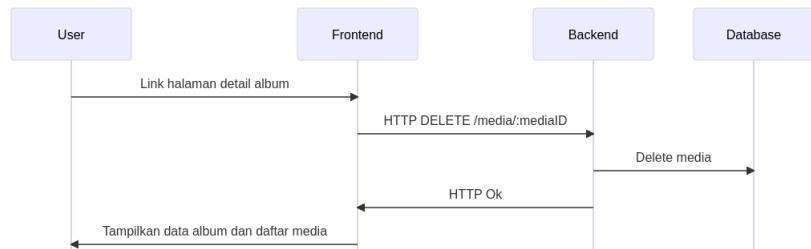
Gambar 4.22 Sequence diagram update album



Gambar 4.23 Sequence diagram hapus album

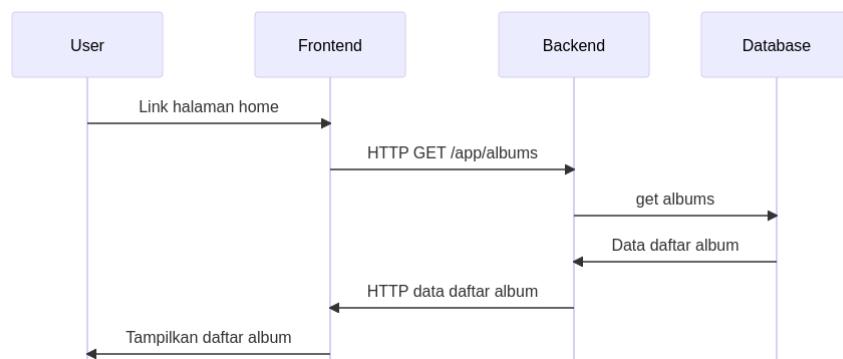


Gambar 4.24 Sequence diagram upload media



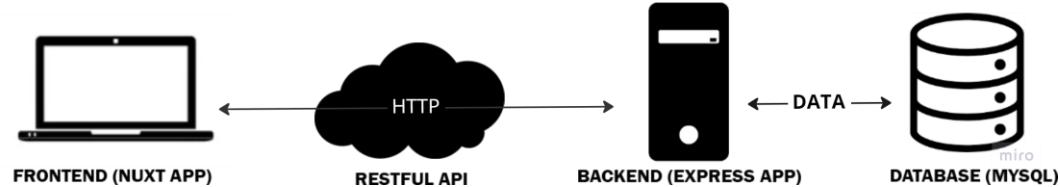
Gambar 4.25 Sequence diagram hapus media

5. Sequence Diagram Akses Album



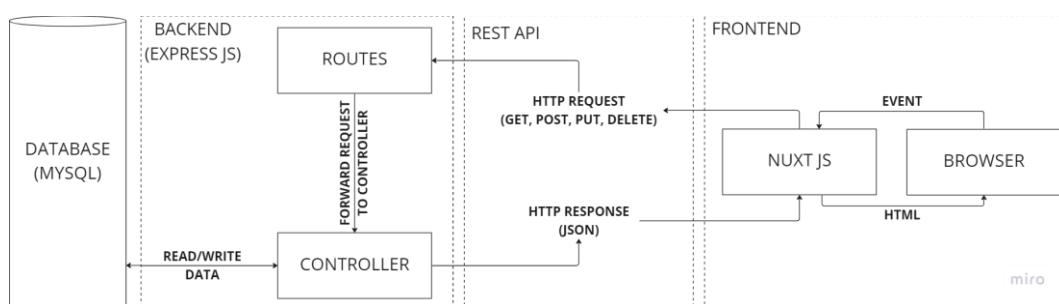
Gambar 4.26 Sequence Diagram Akses Album

4.1.5 Arsitektur RESTful API



Gambar 4.27 Gambaran umum sistem

Berdasarkan Gambar 4.27 diatas, dapat dilihat bahwa untuk memperoleh data user atau album dari *database* arsip media, *frontend* arsip media harus mengirimkan *request* ke *backend* arsip media dengan menggunakan metode HTTP. *Request* kemudian diterima oleh REST API yang kemudian melakukan *forward* ke *backend* dan *backend* akan melakukan tindakan sesuai dengan jenis *request* dari *frontend*. *Backend* kemudian mengambil data yang sesuai dari *database*, data yang telah didapatkan kemudian diproses oleh *backend* dan melakukan *response* data ke *frontend* dengan format JSON.



Gambar 4.28 Design pattern

- Router*: *router* berfungsi sebagai *middleware*. *Router* mendefinisikan bagaimana *response* terhadap *request*. *Router* mendefinisikan *endpoint* dari tiap-tiap fungsi yang ada pada *controller* arsip media dan juga menentukan metode *request* yang digunakan, baik metode GET, POST, PUT, DELETE.

- b. *Controller*: *controller* berfungsi sebagai pengeksekusi *query*, yang kemudian berinteraksi dengan *database* arsip media. *Controller* juga berfungsi untuk mengambil data dari *request* yang dikirimkan ke api oleh *frontend* arsip media, baik melalui *parameter* ataupun *body* dari *request*, dan mengembalikan *response* sesuai dengan hasil *query* pada fungsi didalamnya dalam format JSON.
- c. Nuxt js: Nuxt js berfungsi untuk mentriger setiap *event* yang terjadi pada *browser*, kemudian melakukan aksi sesuai dengan event yang terjadi. Aksi yang dilakukan bisa berupa pengambilan data dari *backend* menggunakan metode HTTP, memperbarui html pada *browser* ataupun melakukan *redirect* halaman.

4.1.6 Rancangan Endpoint API

Tabel 4.1 Rancangan Endpoint API

NO	Name	Method	Path	Body	Auth
1.	Login	POST	/auth/login	email: string password: string	No
2.	Get profile	GET	/account/profile		Yes
3.	Get all permissions	GET	/permissions		Yes
4.	Get all roles	GET	/roles		Yes
5.	Create role	POST	/roles	name: string permissions: number[]	Yes
6.	Update role	PUT	/roles/:RID	name: string permissions: number[]	Yes
7.	Delete role	DELETE	/roles/:RID		Yes
8.	Get all users	GET	/users		Yes
9.	Create user	POST	/users	name: string	Yes

NO	Name	Method	Path	Body	Auth
				email: string password: string roleId: number	
10.	Update user	PUT	/users/:UID	name: string email: string password: string roleId: number	Yes
11.	Delete user	DELETE	/users/:UID		Yes
12.	Give album	POST	/users/:UID/:AID		Yes
13.	Remove album	DELETE	/users/:UID/:AID		Yes
14.	Get all albums	GET	/albums		Yes
15.	Get album	GET	/albums/:AID		Yes
16.	Create album	POST	/albums	name: string description: string	Yes
17.	Update album	PUT	/albums/:AID	name: string description: string	Yes
18.	Delete album	DELETE	/albums/:AID		Yes
19.	Upload media	POST	/media/:AID	file: file	Yes
20.	Delete media	DELETE	/media/:MID		Yes
21.	Akses media	GET	/media/:AID/:MID		Yes
22.	Download media	GET	/media/download/:MID		Yes
23.	Get all user albums	GET	/app/albums		Yes
24.	Get user album	GET	/app/albums/:AID		Yes
25.	Akses media	GET	/app/albums/media/:name		Yes

4.1.7 Rancangan Tabel

Tabel 4.2 Skema tabel Role

No	Nama Field	Tipe data	Ukuran
1	id	Int	
2	name	Varchar	191
3	permissions	Varchar	255
4	updatedAt	Datetime	
5	createdAt	Datetime	

Tabel 4.3 Skema tabel User

No	Nama Field	Tipe data	Ukuran
1	id	Int	
2	name	Varchar	191
3	email	Varchar	191
4	password	Varchar	191
5	roleId	Int	
6	updatedAt	Datetime	
7	createdAt	Datetime	

Tabel 4.4 Skema tabel Album

No	Nama Field	Tipe data	Ukuran
1	id	Int	
2	name	Varchar	191
3	description	Text	
4	updatedAt	Datetime	

5	createdAt	Datetime	
---	-----------	----------	--

Tabel 4.5 Skema tabel Media

No	Nama Field	Tipe data	Ukuran
1	id	Int	
2	name	Varchar	191
3	originalName	Varchar	191
4	type	Varchar	8
5	size	Int	
6	albumId	Int	
7	updatedAt	Datetime	
8	createdAt	Datetime	

Tabel 4.6 Skema tabel UsersOnAlbums

No	Nama Field	Tipe data	Ukuran
1	userId	Int	
2	albumId	Int	

4.1.8 Rancangan *Interface*

1. Halaman Login

The diagram shows a login form within a large rectangular frame. At the top center is the word "LOGO". Below it is a form field labeled "Email address" with a corresponding input box. Below that is another form field labeled "Password" with a corresponding input box. At the bottom center of the form is a "LOGIN" button.

Gambar 4.29 Rancangan halaman login

2. Halaman Home

The diagram shows a home page layout. At the top left is the word "LOGO". At the top right is a "LOGOUT" button. On the left side is a sidebar with a user profile section showing "Nama User" and "email@email.com" with a "ROLE" button. Below this are menu items: "Home" (which is bolded), "Dashboard", "Role", "User", and "Album". The main content area is titled "Daftar album" and contains a table with two rows of data:

ID	Nama	Total Gambar	Total Video	Dibuat	Aksi
1	Nama Album 1	3	2	1 hari yang lalu	<button>Lihat</button>
2	Nama Album 2	4	1	2 hari yang lalu	<button>Lihat</button>

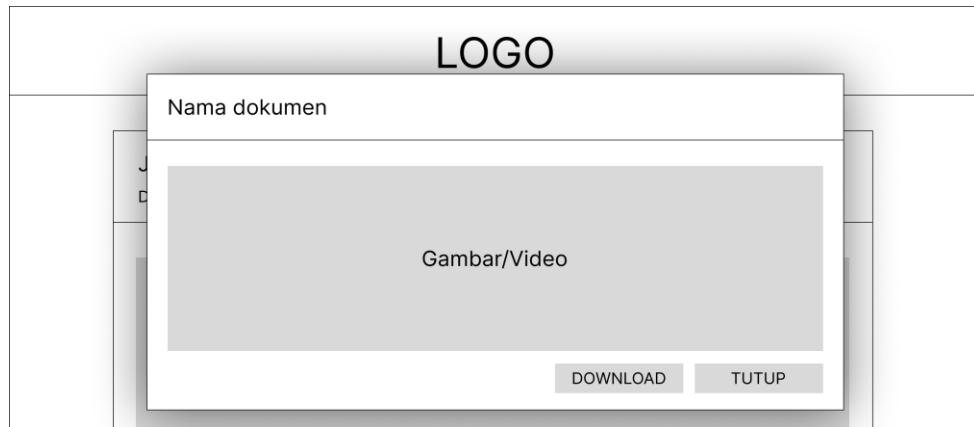
Gambar 4.30 Rancangan halaman *home*

3. Halaman Detail Album User

The diagram shows a detail page for an album. At the top center is the word "LOGO". Below it is a form field titled "Judul album" with a corresponding input box. Below that is a form field titled "Deskripsi album" with a corresponding input box. The main content area contains a table with three rows of data:

Jenis	Ukuran	Nama	Aksi
Image	450.5 kB	Nama file 1.jpg	<button>Lihat</button>
Video	120.9 MB	Nama file 2.mp4	<button>Lihat</button>

Gambar 4.31 Rancangan halaman detail album user



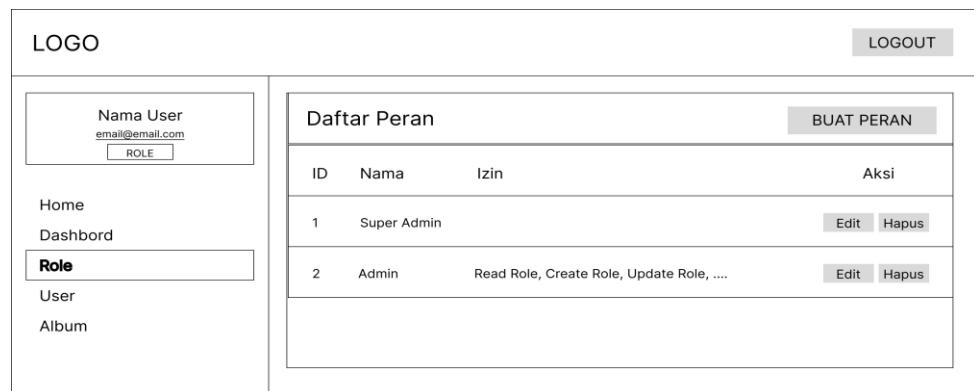
Gambar 4.32 Rancangan *modal preview media*

4. Halaman Dashboard



Gambar 4.33 Rancangan halaman *dashboard*

5. Halaman Peran



Gambar 4.34 Rancangan halaman daftar peran

Logo

Buat Peran / Edit Peran

Nama Peran

Izin

- Create User
- Edit User
- Delete User
- Create Role
- Edit Role

BUAT/SIMPAN BATAL

LOGOUT

PERAN

Home
Dashboard
Role
User
Album

Gambar 4.35 Rancangan modal buat dan edit peran

6. Halaman User

Logo

Daftar User

ID	Nama	Email	Peran	Aksi
1	Super Admin	super@admin.com	Super Admin	Edit Album Hapus

BUAT USER

LOGOUT

Nama User
email@email.com
ROLE

Home
Dashboard
Role
User
Album

Gambar 4.36 Rancangan halaman daftar user

Logo

Buat User / Edit User

Nama

Alamat email

Password

Peran

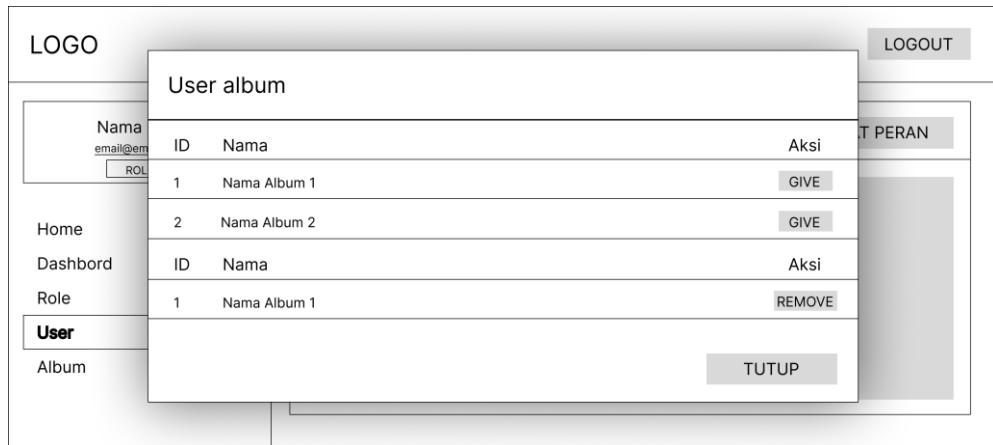
BUAT/SIMPAN BATAL

LOGOUT

PERAN

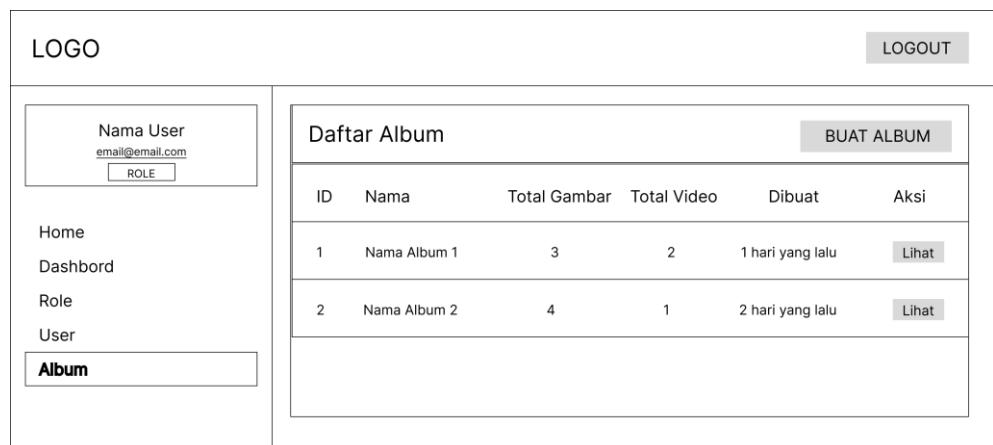
Home
Dashboard
Role
User
Album

Gambar 4.37 Rancangan modal buat dan edit user

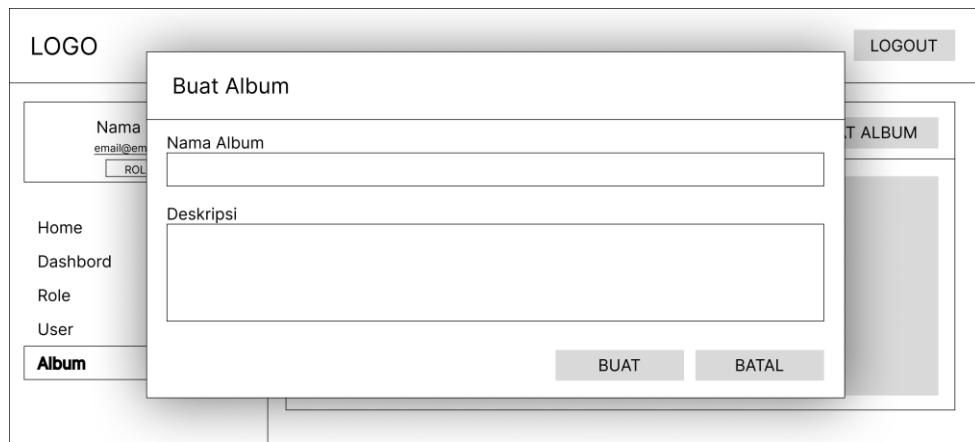


Gambar 4.38 Rancangan modal *give* dan *remove* album user

7. Halaman Album



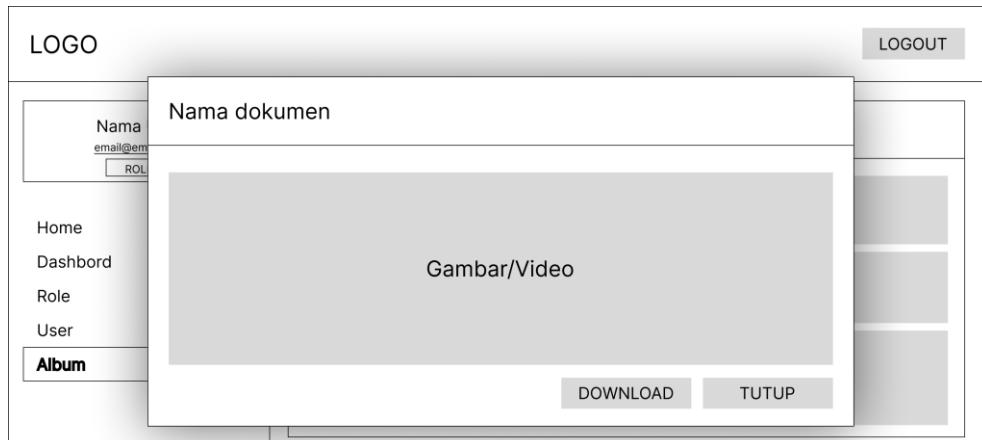
Gambar 4.39 Rancangan halaman daftar album



Gambar 4.40 Rancangan modal buat album



Gambar 4.41 Rancangan halaman detail album

Gambar 4.42 Rancangan modal *preview media*

4.2 Analisis dan Validasi Data

4.2.1 Analisis

Pada penelitian ini aplikasi yang dibangun akan terbagi menjadi dua bagian yang saling berkomunikasi menggunakan protokol HTTP yaitu Backend dan Frontend. Backend akan terhubung langsung dengan database dan akan mengelola data yang dikirimkan oleh Frontend untuk dapat melakukan tugas yang diberikan dan memberikan jawaban yang dibutuhkan oleh Frontend. Sedangkan Frontend akan merender UI aplikasi, dan juga menerima input dan perintah pengguna, dan

juga membuat permintaan http ke backend berdasarkan apa yang dipesan pengguna dan mendapatkan response serta menampilkan data yang diterima.

4.2.2 Validasi Data

Dalam penelitian ini tentunya diambil data penelitian sebagai masukan dan objek penelitian untuk dianalisis. Data yang dikumpulkan berupa 26 dokumen dokumentasi yang dilengkapi dengan masing-masing nama kegiatan.

Tabel 4.7 Daftar nama kegiatan dan jumlah dokumen

No	Nama Kegiatan	Jumlah dokumen
1	Peringatan Hari Ibu, 22 Desember 2022, Kementerian Kesehatan mengadakan Gerakan Nasional Bumil Sehat melalui kampanye #bumilsehat.	10
2	Pertemuan Tim Pokja Penurunan Angka Kematian Ibu (AKI) dan Angka Kematian Bayi (AKB) Provinsi Sulawesi Selatan.	7
3	Pertemuan evaluasi tenaga gizi pendamping desa TK. Provinsi Sulawesi selatan	9

Setelah mengumpulkan data dokumentasi, kami mengumpulkan 2 data pegawai yang nantinya bisa mengakses dokumen-dokumen gambar pada tabel 4.7 diatas.

Tabel 4.8 Daftan pegawai yang dapat mengakses album

No	Nama	Alamat email
1	Andi Tenri Musharifa, Skm	anditenrimusharifa@gmail.com
2	Ramlah Purnama Putri,S.Kom	Putriadzra83@gmail.com

4.3 Hasil dan Pembahasan

Hasil dan implementasi dari empat teknologi yang digunakan yaitu MYSQL, Express JS, Nuxt JS, dan Node JS dalam mengembangkan aplikasi arsip media. Implementasi tersebut akan dijelaskan dengan hasil dan pembahasan masing-masing. Tentang hasil dan pembahasan implementasi MERN adalah sebagai berikut.

4.3.1 Hasil dan Pembahasan Implementasi MYSQL

Pada implementasi MYSQL menggunakan versi 8.0.31, implementasi ini dibantu dengan melakukan instalasi package pendukung menggunakan NPM. Package tersebut yaitu prisma, Adapun untuk CLI (Command Line Interface) yang digunakan untuk instalasi dapat dilihat pada gambar 4.1 berikut

```
npm install prisma --save-dev
npx prisma init --datasource-provider mysql
```

Gambar 4.43 CLI instalasi package prisma

Package prisma akan membantu pengembang dalam beberapa hal diantaranya seperti menghubungkan backend ke server atau database MYSQL, membuat model tabel database dan lain sebagainya. Untuk implementasi model database dapat dilihat pada sub bab Perancangan tabel diatas.

1. Hasil

Untuk hasil implementasi dapat dilihat setelah program dijalankan, dalam kasus ini hasil implementasi dapat dilihat melalui terminal atau command prompt.

```
mysql -u root -p
use arsip_media;
show tables;
```

Gambar 4.44 CLI menampilkan hasil implementasi MYSQL.

Pada gambar 4.42 diatas menampilkan urutan CLI untuk menampilkan hasil dari implmentasi MYSQL. Pada perintah diatas akan menampilkan nama-nama dari tabel yang dibuat secara otomatis oleh prisma, yang mana nama basis data dari penelitian ini adalah arsip_media.

2. Pembahasan

Penggunaan prisma untuk terhubung dengan basis data dalam membangun aplikasi yang memiliki banyak data yang saling berelasi sangatlah mempermudah programmer dalam membangun aplikasi, pasalnya proses untuk membuat database, membuat skema tabel dan mengupdate skema tabel dapat dibuat secara otomatis.

```
model User {
    id      Int      @id @default(autoincrement())
    name   String
    email  String    @unique
    password String
}
```

Gambar 4.45 Kode membuat skema tabel User

Pada gambar 4.43 diatas adalah contoh kode untuk membuat skema dari tabel User yang mana pada gambar diatas akan menghasilkan skema seperti pada tabel 4.9 dibawah.

Tabel 4.9 Hasil pembuatan skema tabel User

No	Nama Field	Tipe data	Ukuran
1	id	Int	
2	name	Varchar	191
3	email	Varchar	191
4	password	Varchar	191

4.3.2 Hasil dan pembahasan implementasi Nuxt JS

Mengimplementasikan Nuxt JS untuk pengembangan *frontend* Arsip Media tentunya tidak lepas dari kenyamanan dan performa yang diinginkan. Pengembangan web berbasis SPA dikatakan lebih cepat dari pada pengembangan web multi-halaman konvensional. Ini karena SPA hanya akan memuat halaman web satu kali, dan jika ada perubahan data, hanya data yang relevan yang akan dimuat tanpa memuat sumber daya lain yang tidak diperlukan. Tentu saja, ini menghemat transaksi data seperti permintaan dan tanggapan yang dilakukan ke server untuk mengurangi *bandwidth*. Selain itu, kemudahan mengimplementasikan Nuxt JS berguna karena Anda dapat menambahkan paket lain menggunakan NPM.

1. Hasil

Untuk hasil dari implementasi Nuxt JS pada pengembangan frontend aplikasi Arsip Media dapat dilihat secara lengkap melalui link <https://skripsi.dedeard.my.id>.

2. Pembahasan

Nuxt JS adalah *framework* tingkat tinggi yang dibangun di atas Vue. Sederhananya, Nuxt.js adalah *framework* yang akan memudahkan dalam membangun aplikasi Vue.js mulai dari *development* dan *rendering*. Nuxt bekerja untuk menghilangkan sebagian besar kerumitan dan kompleksitas yang melibatkan berurusan dengan hal-hal seperti data asinkron, *middleware*, dan perutean. Cara kerjanya sangat mirip dengan Angular Universal. Selain itu, Nuxt JS memiliki fitur yang dapat membantu Anda membuat halaman web statis dengan membuat perintah. Kemampuan Nuxt luar biasa dan menyediakan alat bangun konfigurasi statis yang sama dengan milik Jekyll.

Nuxt.js menyertakan hal-hal seperti vue-router. Ini mirip dengan Vue.js 2.0. Nuxt.js juga menyertakan Vuex saat menggunakan opsi store. Selain itu, Nuxt memiliki opsi rendering server Vue dan vue-meta. Ini bagus karena menghilangkan beban secara manual termasuk dan mengkonfigurasi berbagai perpustakaan yang diperlukan untuk mengembangkan aplikasi vue.js yang disediakan server. Nuxt.js melakukan semua ini langsung dari kotaknya, sambil mempertahankan ukuran total min 57kb + gzip (60kb dengan vuex). Nuxt.js juga menggunakan webpack dengan vue-loader dan babel-loader untuk pengemasan, pemecahan kode, dan minifikasi. Kami akan menjelaskan lebih detail cara kerja Nuxt di bawah ini.

Inilah yang terjadi ketika pengguna mengunjungi aplikasi Nuxt.js atau membuka salah satu halamannya melalui <nuxt-link>:

- a. Ketika pengguna awalnya mengunjungi aplikasi, jika tindakan nuxtserverinit didefinisikan di toko, nuxt.js akan menyebutnya dan memperbarui toko.
- b. Ini kemudian akan menjalankan middleware yang ada untuk halaman yang dikunjungi. Nuxt pertama-tama memeriksa file nuxt.config.js untuk middleware global, kemudian memeriksa file tata letak yang sesuai (untuk halaman yang diminta) dan terakhir memeriksa halaman dan elemen turunannya untuk menemukan *middleware*. *Middleware* lebih disukai dalam urutan ini.
- c. Jika rute yang dikunjungi adalah rute dinamis dan ada metode *autentikasi* untuk mengotentikasi rute.
- d. Selanjutnya, nuxt.js memanggil metode ASYNCDATA dan mengambil untuk memuat data sebelum merender halaman. Metode ASYNCDATA digunakan

untuk mengambil data dan menampilkan sisi server data, sedangkan metode Fetch digunakan untuk mengisi penyimpanan sebelum merender halaman.

- e. Pada langkah terakhir, halaman (berisi semua data yang relevan) ditampilkan.

4.3.3 Hasil dan Pembahasan Implementasi Express JS

Menerapkan Express JS di web Arsip Media sebagai kerangka kerja yang digunakan untuk membangun *backend* adalah pilihan yang tepat. Selain menggunakan prisma untuk terkoneksi dengan basis data yang mana prisma juga menggunakan data berbasis JSON, sangat cocok dipadukan dengan bahasa pemrograman JavaScript, hal ini juga dikarenakan performa dan kemudahan yang dibawa oleh Express JS.

Hasil yang didapat dari implementasi Express JS tentunya berupa API yang telah terintegrasi dengan basis data MYSQL. Dengan API ini sisi *frontend* akan dapat terhubung ke bagian *backend* menggunakan metode HTTP. API yang telah dibuat dalam pengembangan aplikasi ini dapat diakses pada <https://kkl.dedeard.my.id/>.

4.3.4 Hasil dan Pembahasan Implementasi Node JS

Implementasi Node JS dapat dipahami sebagai inti dari implementasi Nuxt JS dan Node JS sebelumnya. Memang, implementasi keduanya dimungkinkan karena Node JS adalah lingkungan untuk menjalankan dan menginstal keduanya bersama dengan pustaka pendukung lainnya. Selain itu, bahkan implementasi MYSQL memerlukan instalasi pustaka dan juga memerlukan Node JS. Oleh karena

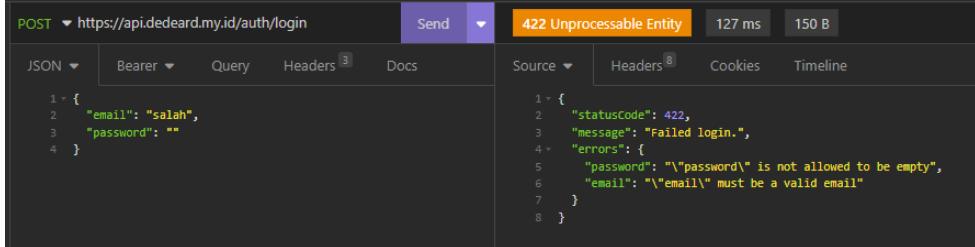
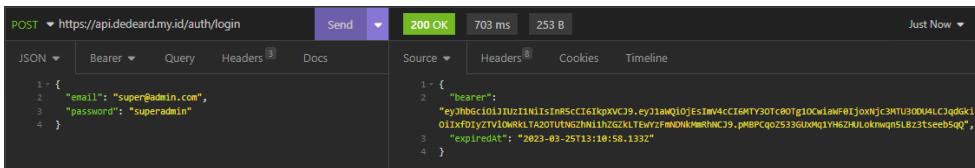
itu, dapat disimpulkan bahwa kemudahan dan kecepatan pengembangan Arsip Media adalah dampak dari penerapan Nuxt JS, Express JS dan MYSQL dengan Node JS sebagai lingkungan pengembangan untuk menjalankan, menginstal, men-debug, dll.

4.4 Pengujian Sistem

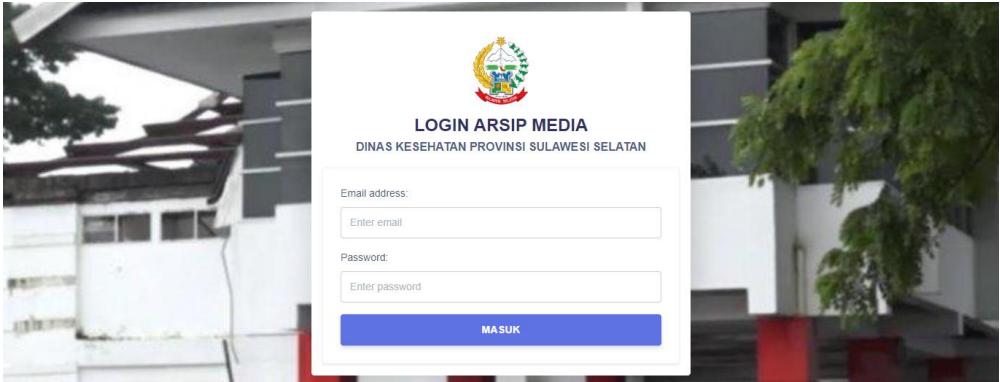
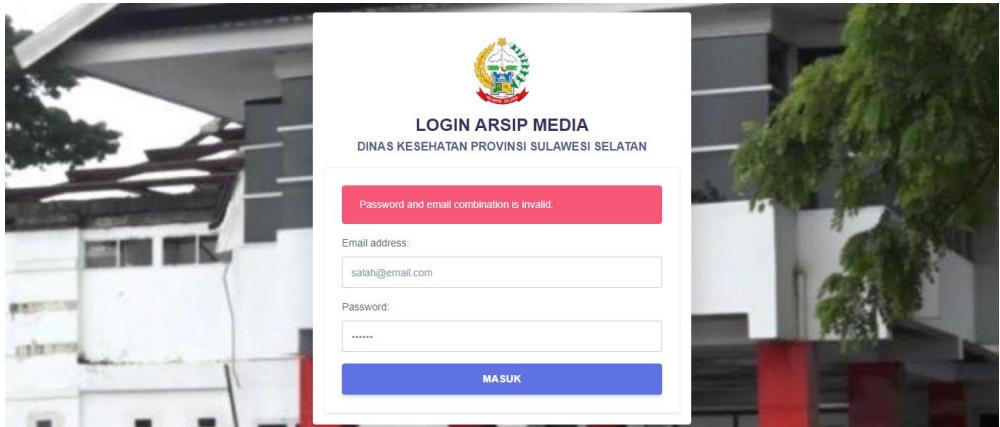
Pengujian sistem dilakukan dengan metode pengujian langsung berbasis teknik black box dengan menguji fungsionalitas aplikasi, tombol-tombol dan relevansi hasil aplikasi.

4.4.1 Pengujian Halaman Login

Tabel 4.10 Pengujian *backend* fungsi login.

<i>Test Factor</i>	Hasil	Keterangan
Melakukan <i>request</i> dengan metode POST dengan <i>body</i> berupa email dan password.	OK	GAGAL: <i>response</i> berupa pesan error. BERHASIL: <i>response</i> berupa data token otentifikasi.
<i>Screen Shot</i>		
 <pre> POST https://api.dedeard.my.id/auth/login Send ▾ 422 Unprocessable Entity 127 ms 150 B Source ▾ Headers [8] Cookies Timeline 1 = { 2 "statusCode": 422, 3 "message": "Failed login.", 4 "errors": [5 "password": "\"password\" is not allowed to be empty", 6 "email": "\"email\" must be a valid email" 7] 8 } </pre>	 <pre> POST https://api.dedeard.my.id/auth/login Send ▾ 200 OK 703 ms 253 B Just Now ▾ Source ▾ Headers [8] Cookies Timeline 1 = { 2 "bearer": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJiaWQlOjEsIMV4CCIGMTY3OTc0OTg1OCwiAiF0IjoxNjC3MTU3ODU4LCJqdGkiOlxrbDiyTzI0mRkLTz20tUtgNZHNUlhGZKLTwFzFmNDkM8RNJCJ9.pMBPcqZ533GUxq1YH6ZHULoknwqn5LBz3tseeBsqQ", 3 "expiredAt": "2023-03-25T13:10:58.133Z" 4 } </pre>	

Tabel 4.11 Pengujian *frontend* fungsi login.

Test Factor	Hasil	Keterangan
User menginput email dan password kemudian mengklik tombol login.	OK	GAGAL: Aplikasi akan menampilkan pesan error. BERHASIL: Halam akan beralih ke halaman <i>dashboard</i> . Jika user adalah seorang admin maka akan muncul menu Home, Dasbor, Peran, User dan Album . Jika User Bukan Seorang Admin maka hanya akan muncul menu Home dan Dasbor .
Screen Shot		
 		

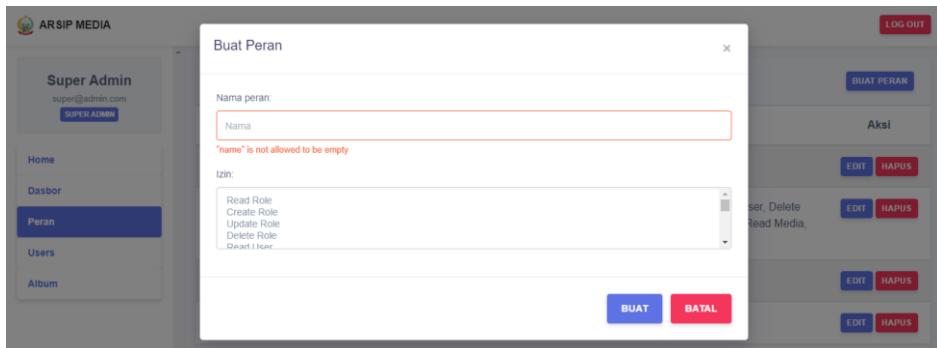
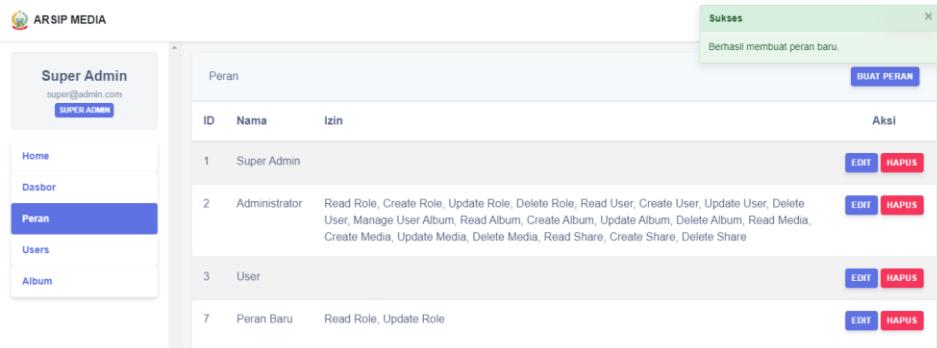
4.4.2 Pengujian Halaman Peran

Tabel 4.12 Pengujian *backend* fungsi buat peran

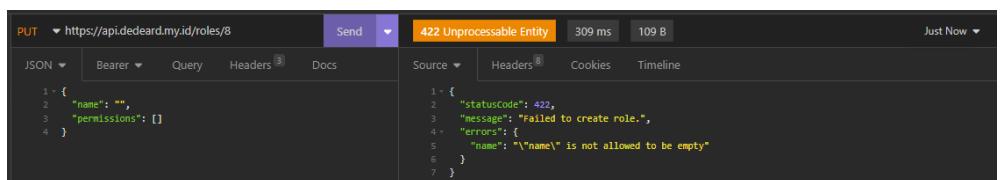
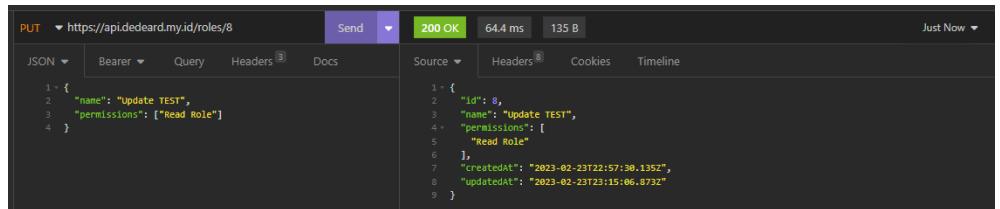
Test Factor	Hasil	Keterangan
Melakukan <i>request</i> dengan metode POST dengan <i>body</i> berupa <i>name</i> dan <i>permissions</i> .	OK	GAGAL: <i>response</i> berupa pesan error. BERHASIL: <i>response</i> berupa data peran baru.
Screen Shot		
<pre> POST https://api.dedead.my.id/roles/ Send 422 Unprocessable Entity 44.2 ms 109 B Just Now Source Headers Cookies Timeline 1: { 2: "status": 422, 3: "message": "Failed to create role.", 4: "errors": [5: { 6: "name": "\\"name\\\" is not allowed to be empty" 7: } </pre>		

Test Factor	Hasil	Keterangan
Melakukan <i>request</i> dengan metode POST dengan <i>body</i> berupa <i>name</i> dan <i>permissions</i> .	OK	GAGAL: <i>response</i> berupa pesan error. BERHASIL: <i>response</i> berupa data peran baru.
Screen Shot		
<pre> POST https://api.dedead.my.id/roles/ Send 200 OK 48.8 ms 117 B Just Now Source Headers Cookies Timeline 1: { 2: "id": 8, 3: "name": "Test", 4: "permissions": [], 5: "createdAt": "2023-02-23T22:57:30.135Z", 6: "updatedAt": "2023-02-23T22:57:30.135Z" 7: } </pre>		

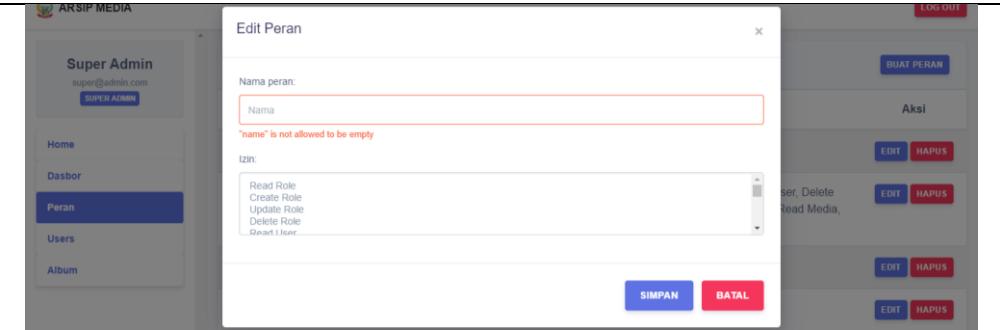
Tabel 4.13 Pengujian *frontend* fungsi buat peran.

Test Factor	Hasil	Keterangan
User mengklik Buat Peran maka akan muncul modal. kemudian user menginput nama dan izin peran.	OK	GAGAL: Aplikasi akan menampilkan pesan error. BERHASIL: Modal akan otomatis tertutup dan memunculkan data baru pada table daftar peran.
Screen Shot		
		
		
		

Tabel 4.14 Pengujian *backend* fungsi edit peran.

Test Factor	Hasil	Keterangan
Melakukan <i>request</i> dengan metode PUT dengan <i>body</i> berupa <i>name</i> dan <i>permissions</i> dan parameter berupa id peran.	OK	GAGAL: <i>response</i> berupa pesan error. BERHASIL: <i>response</i> berupa data peran baru.
Screen Shot		
		 

Tabel 4.15 Pengujian *frontend* fungsi edit peran.

Test Factor	Hasil	Keterangan
User mengklik Edit pada table peran maka akan muncul modal kemudian user mengedit nama atau izin peran.	ok	GAGAL: <i>response</i> berupa pesan error. BERHASIL: modal akan otomatis tertutup dan mengupdate data table peran.
Screen Shot		
		

The screenshot shows two parts of the ARSIP MEDIA backend interface. The top part is a modal titled 'Edit Peran' where a new role named 'test' is being created with permissions like 'Read Role', 'Create Role', and 'Update Role'. The bottom part is a table listing existing roles: Super Admin, Administrator, Normal User, and the newly created 'test' role.

ID	Nama	Izin	Aksi
1	Super Admin		EDIT HAPUS
2	Administrator	Read Role, Create Role, Update Role, Delete Role, Read User, Create User, Update User, Delete User, Manage User Album, Read Album, Create Album, Update Album, Delete Album, Read Media, Create Media, Update Media, Delete Media, Read Share, Create Share, Delete Share	EDIT HAPUS
3	Normal User		EDIT HAPUS
5	test	Read Role, Create Role, Update Role	EDIT HAPUS

Tabel 4.16 Pengujian *backend* fungsi hapus peran.

Test Factor	Hasil	Keterangan
Melakukan <i>request</i> dengan metode DELETE dengan parameter berupa id peran.	OK	BERHASIL: kode <i>response</i> berupa 204 yaitu no content.
Screen Shot		
<p>The screenshot shows a Postman request to the URL <code>https://api.dedearn.my.id/roles/8</code>. The response status is 204 No Content, indicating that the data was successfully deleted. The response body is empty, showing "No body returned for response".</p>		

Tabel 4.17 Pengujian *frontend* fungsi hapus peran.

Test Factor	Hasil	Keterangan
User mengklik Hapus pada table peran maka akan muncul modal konfirmasi hapus peran.	ok	Jika berhasil modal akan otomatis tertutup dan menghapus data pada table peran.
Screen Shot		

The screenshot shows a modal dialog box centered over a table of roles. The dialog contains a red 'X' icon at the top, followed by the text "Anda yakin ingin menghapus peran?", two buttons ("Ya, Hapus" and "Batal") at the bottom, and a background overlay of the table.

ID	Nama	Aksi
1	Super Admin	<button>EDIT</button> <button>HAPUS</button>
2	Administrator	<button>EDIT</button> <button>HAPUS</button>
3	Normal User	<button>EDIT</button> <button>HAPUS</button>
5	test	<button>EDIT</button> <button>HAPUS</button>

The screenshot shows the 'Peran' (Roles) page with a table listing roles and their permissions. A 'BUAT PERAN' button is visible at the top right.

ID	Nama	Izin	Aksi
1	Super Admin	Read Role, Create Role, Update Role, Delete Role, Read User, Create User, Update User, Delete User, Manage User Album, Read Album, Create Album, Update Album, Delete Album, Read Media, Create Media, Update Media, Delete Media, Read Share, Create Share, Delete Share	<button>EDIT</button> <button>HAPUS</button>
2	Administrator	Read Role, Create Role, Update Role, Delete Role, Read User, Create User, Update User, Delete User, Manage User Album, Read Album, Create Album, Update Album, Delete Album, Read Media, Create Media, Update Media, Delete Media, Read Share, Create Share, Delete Share	<button>EDIT</button> <button>HAPUS</button>
3	Normal User	Read Role, Create Role, Update Role, Delete Role, Read User, Create User, Update User, Delete User, Manage User Album, Read Album, Create Album, Update Album, Delete Album, Read Media, Create Media, Update Media, Delete Media, Read Share, Create Share, Delete Share	<button>EDIT</button> <button>HAPUS</button>

4.4.3 Pengujian Halaman User

Tabel 4.18 Pengujian *backend* fungsi buat user.

Test Factor	Hasil	Keterangan
Melakukan <i>request</i> dengan metode POST dengan body berupa data user.	OK	GAGAL: <i>response</i> berupa pesan error. BERHASIL: <i>response</i> berupa data user baru.
Screen Shot		
<p>The screenshot shows a Postman interface with a failed POST request to the URL <code>https://api.dedead.my.id/users</code>. The response status is 422 Unprocessable Entity, and the response body is a JSON object containing error details:</p> <pre> 1 - { 2 "statusCode": 422, 3 "message": "Failed create user.", 4 "errors": [5 "name": "\"name\" is required", 6 "password": "\"password\" is required", 7 "email": "\"email\" is required", 8 "roleId": "\"roleId\" is required" 9] 10 } </pre>		

```

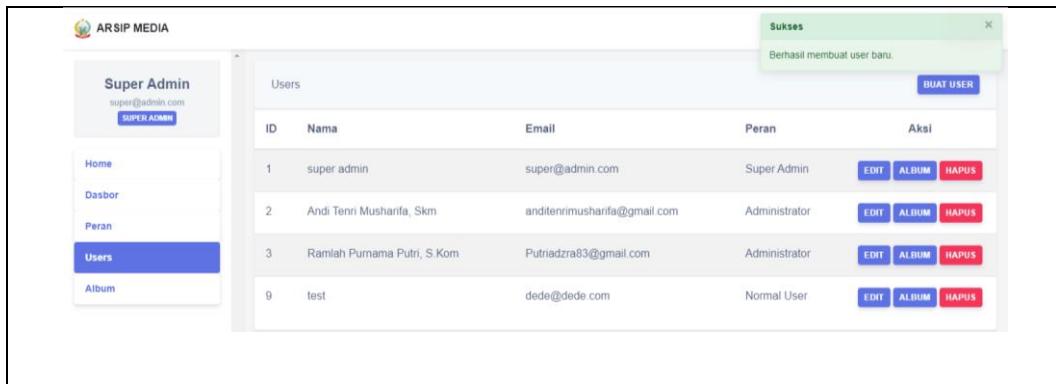
POST https://api.dedead.my.id/users
Send 200 OK 289 ms 564 B Just Now
Source Headers Cookies Timeline
1 < {
2   "name": "Contoh user",
3   "password": "password",
4   "email": "email@mail.com",
5   "roleId": 2
6 }
1 > {
2   "id": 12,
3   "name": "Contoh user",
4   "roleId": 2,
5   "email": "email@mail.com",
6   "createdAt": "2023-02-23T23:35:45.724Z",
7   "updatedAt": "2023-02-23T23:35:45.724Z",
8 >
9   "role": {
10     "id": 2,
11     "name": "Administrator",
12     "permissions": [ < 20 > ],
13     "createdAt": "2023-02-23T23:35:45.724Z",
14     "updatedAt": "2023-02-23T23:35:45.724Z"
15   }
16 }
17

```

Tabel 4.19 Pengujian *frontend* fungsi buat user.

<i>Test Factor</i>	Hasil	Keterangan
User mengklik Buat User maka akan muncul modal kemudian user menginput data user.	OK	GAGAL: Aplikasi akan menampilkan pesan error. BERHASIL: modal akan otomatis tertutup dan memunculkan data baru pada table user.

Screen Shot

Tabel 4.20 Pengujian *backend* fungsi edit user.

Test Factor	Hasil	Keterangan
Melakukan <i>request</i> dengan metode PUT dengan body berupa data user dan parameter berupa id user.	OK	GAGAL: <i>response</i> berupa pesan error. BERHASIL: <i>response</i> berupa data user yang baru.
<i>Screen Shot</i>		

Tabel 4.21 Pengujian *frontend* fungsi edit user.

Test Factor	Hasil	Keterangan
User mengklik Edit pada table user maka akan muncul modal kemudian user mengedit data user.	OK	GAGAL: <i>response</i> berupa pesan error. BERHASIL: modal akan otomatis tertutup dan mengupdate data table peran.

Screen Shot

ID	Nama	Email	Peran	Aksi
1	super admin	super@admin.com	Super Admin	EDIT ALBUM HAPUS
2	Andi Tenri Musharifa, Skm	anditenrimusharifa@gmail.com	Administrator	EDIT ALBUM HAPUS
3	Ramlah Purnama Putri, S.Kom	Putriadzra83@gmail.com	Administrator	EDIT ALBUM HAPUS
9	Test Edited	dede@dede.com	Normal User	EDIT ALBUM HAPUS

Tabel 4.22 Pengujian *backend* fungsi berikan dan hapus album user.

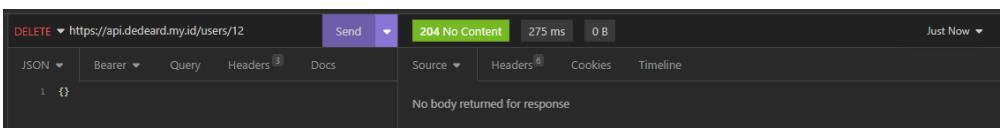
Test Factor	Hasil	Keterangan
Melakukan <i>request</i> dengan metode POST/DELETE dengan parameter berupa id user dan id album.	OK	BERHASIL: kode <i>response</i> berupa 204.

Screen Shot

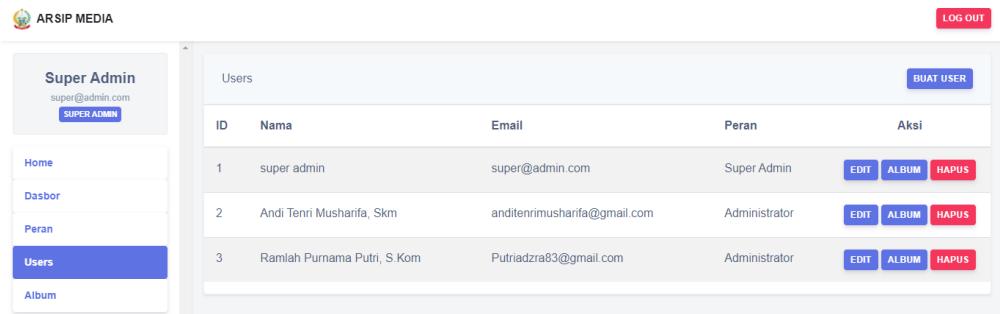
Tabel 4.23 Pengujian *frontend* fungsi berikan dan hapus album user.

Test Factor	Hasil	Keterangan
User mengklik Album pada table user maka akan muncul modal untuk meberikan atau menghapus album yang dapat diakses oleh user.	OK	BERHASIL: Jika berhasil table akan otomatis terupdate.
Screen Shot		

Tabel 4.24 Pengujian *backend* fungsi hapus user.

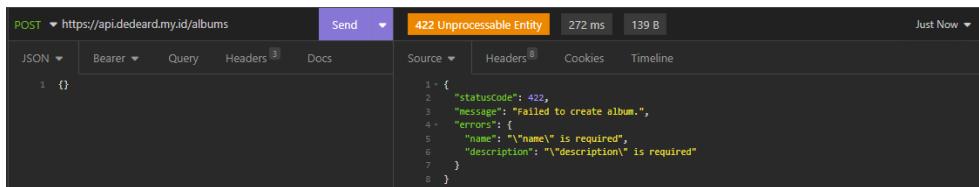
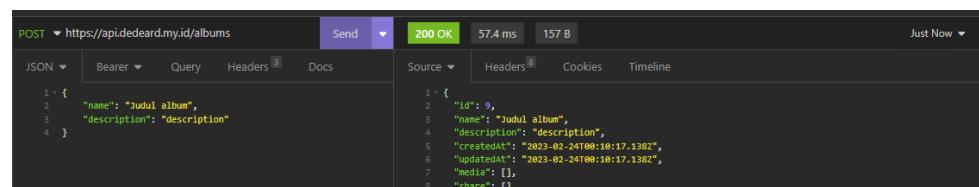
Test Factor	Hasil	Keterangan
Melakukan <i>request</i> dengan metode DELETE dengan parameter berupa id user.	OK	BERHASIL: kode <i>response</i> berupa 204 yaitu no content.
Screen Shot		
		

Tabel 4.25 Pengujian *frontend* fungsi hapus user.

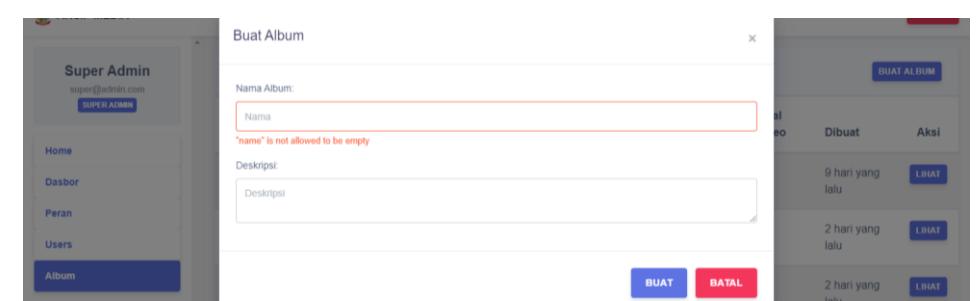
Test Factor	Hasil	Keterangan
User adalah admin kemudian mengklik Hapus pada table user maka akan muncul modal konfirmasi hapus user.	OK	BERHASIL: modal akan otomatis tertutup dan menghapus data pada table user.
Screen Shot		
		
		

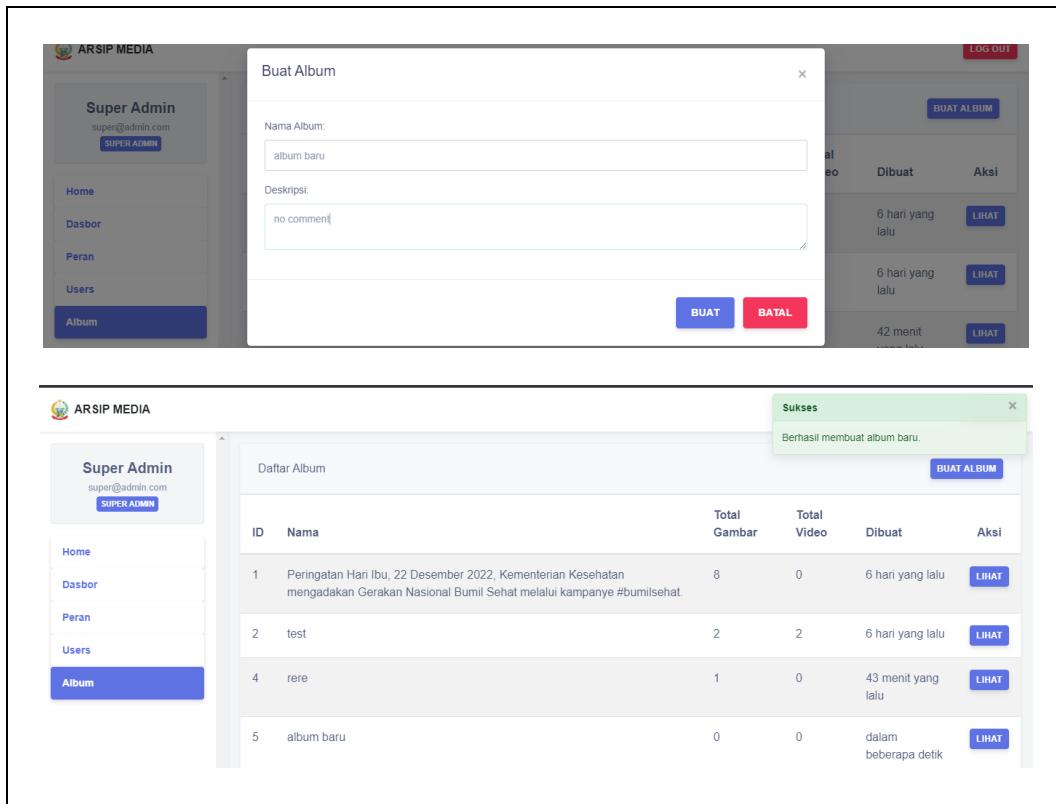
4.4.4 Pengujian Halaman Album

Tabel 4.26 Pengujian *backend* fungsi buat album.

<i>Test Factor</i>	Hasil	Keterangan
Melakukan <i>request</i> dengan metode POST dengan body berupa data album.	OK	GAGAL: <i>response</i> berupa pesan error. BERHASIL: <i>response</i> berupa data album baru.
Screen Shot		
		

Tabel 4.27 Pengujian *frontend* fungsi buat album.

<i>Test Factor</i>	Hasil	Keterangan
User mengklik Buat Album maka akan muncul modal kemudian user menginput data album.	OK	GAGAL: Aplikasi akan menampilkan pesan error. BERHASIL: modal akan otomatis tertutup dan memunculkan data baru pada table album.
Screen Shot		
		

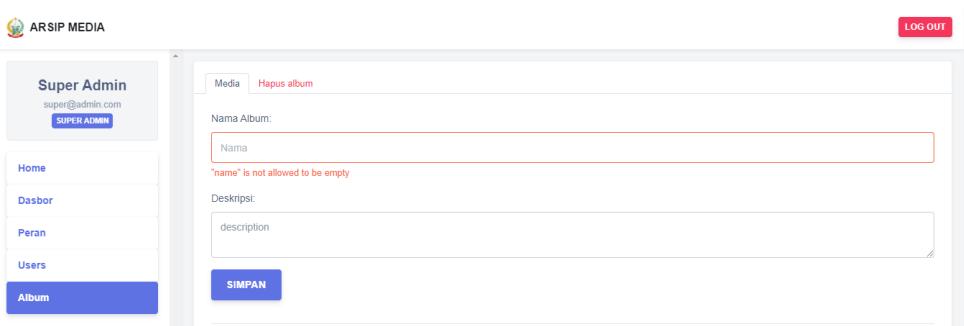
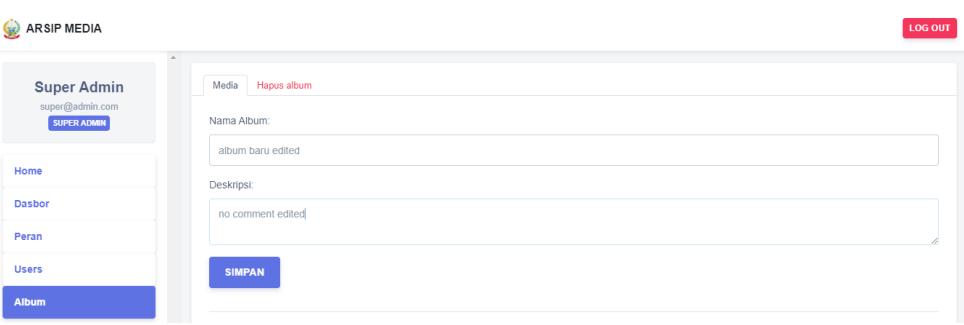
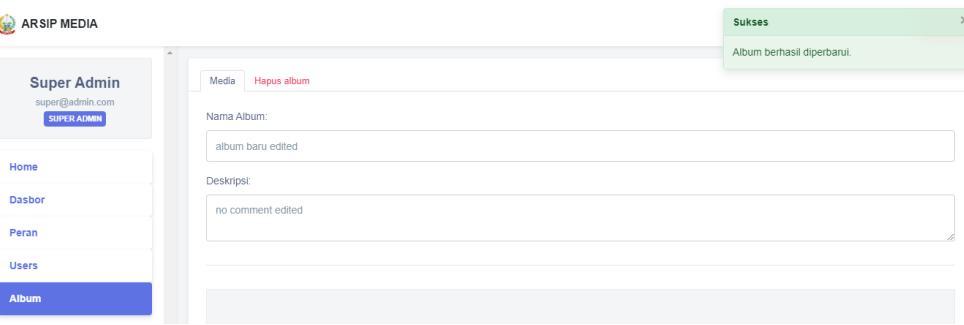


Tabel 4.28 Pengujian *backend* fungsi edit informasi album.

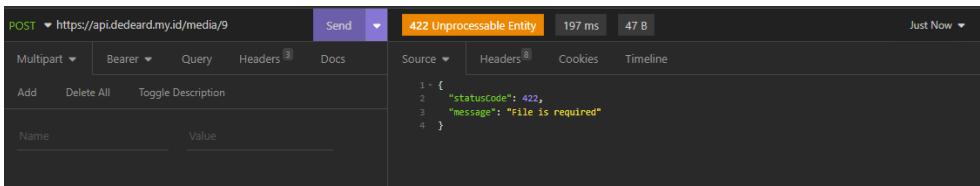
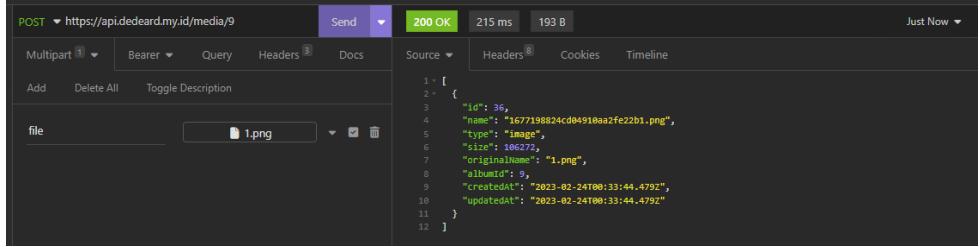
<i>Test Factor</i>	Hasil	Keterangan
Melakukan <i>request</i> dengan metode PUT dengan body berupa data album dan parameter berupa id album.	OK	GAGAL: <i>response</i> berupa pesan error. BERHASIL: <i>response</i> berupa data user yang baru.

Screen Shot

Tabel 4.29 Pengujian *frontend* fungsi edit informasi album.

Test Factor	Hasil	Keterangan
User mengklik Lihat pada table album. Kemudian mengedit data album. Kemudian mengklik tombol Simpan .	OK	GAGAL: <i>response</i> berupa pesan error. Berhasil: makan akan memunculkan notifikasi berhasil menyimpan data.
Screen Shot		
		 <p>This screenshot shows the 'ARSIP MEDIA' application interface. On the left, there's a sidebar with 'Super Admin' logged in. The 'Album' menu item is highlighted in blue. In the main content area, there's a form titled 'Media' with a sub-tab 'Hapus album'. The 'Nama Album:' field is empty and has a red border, with an error message below it: "'name' is not allowed to be empty". Below the field is a 'Deskripsi:' field containing 'description'. At the bottom is a blue 'SIMPAN' button.</p>
		 <p>This screenshot shows the same application interface after the user has edited the album name. The 'Nama Album:' field now contains 'album baru edited'. The 'Deskripsi:' field remains empty. The 'SIMPAN' button is visible at the bottom.</p>
		 <p>This screenshot shows the application after the data has been saved. A green success toast notification in the top right corner says 'Sukses' and 'Album berhasil diperbarui.' (Album successfully updated). The main form fields are still visible below the toast.</p>

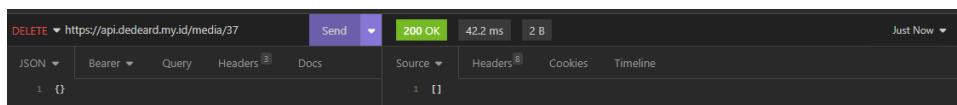
Tabel 4.30 Pengujian *backend* fungsi upload media.

Test Factor	Hasil	Keterangan
Melakukan <i>request</i> dengan metode POST dengan body berupa gambar dan parameter berupa id album.	OK	GAGAL: <i>response</i> berupa pesan error. BERHASIL: <i>response</i> berupa data media.
Screen Shot		
		 <pre> POST https://api.dedead.my.id/media/9 Send 422 Unprocessable Entity 197 ms 47 B Just Now Source Headers Cookies Timeline 1 + { 2 + "statusCode": 422, 3 + "message": "File is required" 4 + } </pre>
		 <pre> POST https://api.dedead.my.id/media/9 Send 200 OK 215 ms 193 B Just Now Source Headers Cookies Timeline 1 + [2 + { 3 + "id": 36, 4 + "name": "167198824cd04910aa2fe22b1.png", 5 + "type": "image", 6 + "size": 106272, 7 + "originalName": "1.png", 8 + "albumId": 9, 9 + "createdAt": "2023-02-24T00:33:44.479Z", 10 + "updatedAt": "2023-02-24T00:33:44.479Z" 11 + } 12 +] </pre>

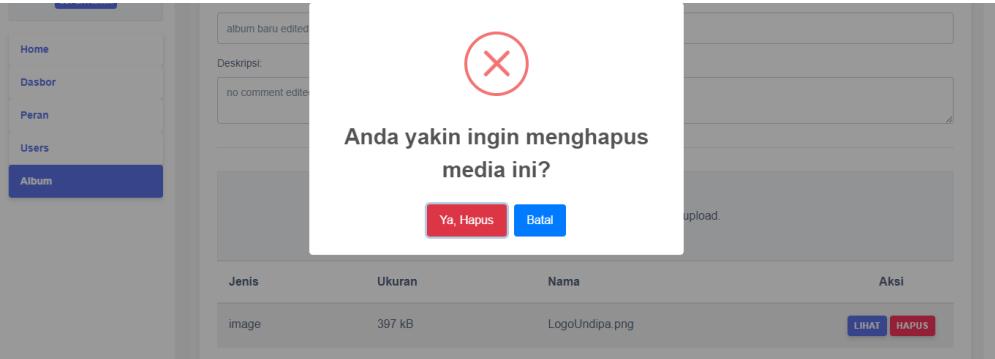
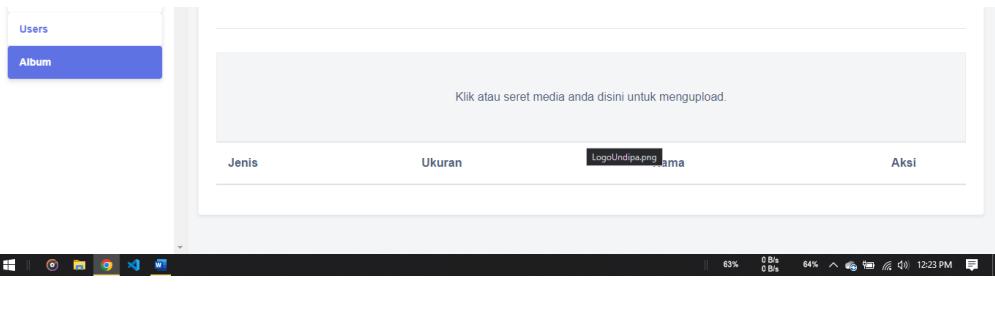
Tabel 4.31 Pengujian *frontend* fungsi upload media.

Test Factor	Hasil	Keterangan								
User mengklik Lihat pada table album. Kemudian mengklik atau menyeret media pada kotak upload media.	OK	GAGAL: <i>response</i> berupa pesan error. BERHASIL: makan akan menambahkan data baru pada table daftar media.								
Screen Shot										
		 <table border="1"> <thead> <tr> <th>Jenis</th> <th>Ukuran</th> <th>Nama</th> <th>Aksi</th> </tr> </thead> <tbody> <tr> <td>image</td> <td>397 kB</td> <td>LogoUndip.png</td> <td>LIHAT HAPUS</td> </tr> </tbody> </table>	Jenis	Ukuran	Nama	Aksi	image	397 kB	LogoUndip.png	LIHAT HAPUS
Jenis	Ukuran	Nama	Aksi							
image	397 kB	LogoUndip.png	LIHAT HAPUS							

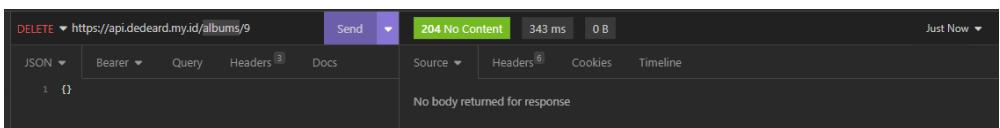
Tabel 4.32 Pengujian *backend* fungsi hapus media.

Test Factor	Hasil	Keterangan
Melakukan <i>request</i> dengan metode DELETE dengan parameter berupa id media.	OK	BERHASIL: kode <i>response</i> berupa 204 yaitu no content.
Screen Shot		
		

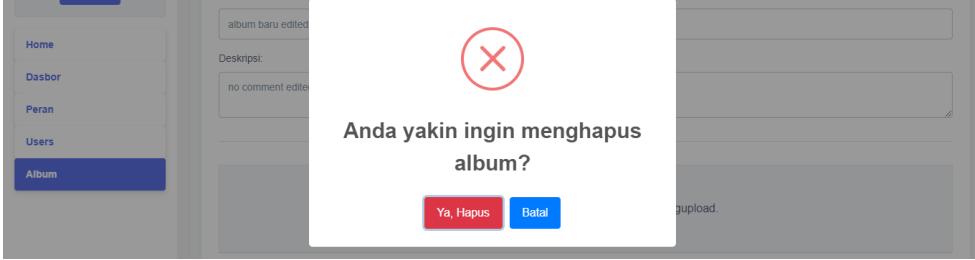
Tabel 4.33 Pengujian *frontend* fungsi hapus media.

Test Factor	Hasil	Keterangan
User mengklik Lihat pada table album. Kemudian pada table daftar media user mengklik hapus . maka akan muncul modal konfirmasi hapus media.	OK	Berhasil: tabel media akan otomatis terupdate.
Screen Shot		
		
		

Tabel 4.34 Pengujian *backend* fungsi hapus album.

Test Factor	Hasil	Keterangan
Melakukan <i>request</i> dengan metode DELETE dengan parameter berupa id album.	OK	BERHASIL: kode <i>response</i> berupa 204 yaitu no content.
Screen Shot		
		

Tabel 4.35 Pengujian *frontend* fungsi hapus album.

Test Factor	Hasil	Keterangan
User mengklik Lihat pada table album. Kemudian pada halaman detail album User mengklik tombol Hapus Album maka akan muncul modal konfirmasi hapus album.	OK	BERHASIL: halaman akan beralih ke halaman daftar album dan menghapus album beserta medianya.
Screen Shot		
		
		

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan pengujian beberapa kali maka kesimpulan yang dirangkum oleh peneliti dalam penelitian ini ialah:

1. Aplikasi arsip media dapat dibangun dengan menggunakan Express sebagai *backend*, Nuxt sebagai *frontend* dan Arsitektur RESTful API sebagai jembatan penghubung antara *backend* dan *frontend*.
2. Penerapan metode HTTP untuk melakukan transaksi data antara *backend* yang dibangun dengan mengimplementasikan Arsitektur RESTful API dengan *frontend* yang dibangun dengan Nuxt dapat dilakukan dengan baik.

5.2 Saran

Agar memperoleh hasil yang lebih baik kedepannya untuk aplikasi yang dibangun, maka peneliti memberikan saran sebagai berikut:

1. Terdapat beberapa kekurangan yang dapat diimplementasikan untuk pengembangan aplikasi kedepannya, seperti fitur multi upload dokumen sehingga membuat proses mengunggah dokumen lebih cepat.
2. Untuk pengembangan aplikasi selanjutnya dapat dikembangkan pada sistem operasi platform yang lain seperti android. Peneliti menyarankan agar skripsi ini dapat dijadikan sebagai bahan referensi untuk mengembangkan sistem yang lebih sempurna.

DAFTAR PUSTAKA

- Adivia, G. P., & Ari, S. (2020). *Pemanfaatan Vue Js Pada Fitur Pengaturan Tambak Dalam Aplikasi Budi Daya Tambak Jala*.
- Alazab, A., Khraisat, A., Alazab, M., & Singh, S. (2022). *Detection of Obfuscated Malicious JavaScript Code*.
- Amin, R., Dewi, E. M., & Herman, K. (2017). *Implementasi REST API pada Aplikasi Panduan Kepaskibraan Berbasis Android*.
- Benjamin, B. N., Behnaz, H., & François, G. (2019). *Nodest: feedback-driven static analysis of Node.js applications*.
- Kok, L. T. (2020). *Hands-On Nuxt.js Web Development : Build Universal and Static-Generated Vue.js Applications Using Nuxt.js*.
- Lau, T. K. (2020). *Hands-on Nuxt.js Web Development: Build universal and static-generated Vue.js applications using Nuxt.js*.
- Mulyani, S. (2016). *Metode Analisis dan Perancangan Sistem*.
- Munawar. (2018). *Analisis perancangan sistem berorientasi objek dengan UML (unified modeling language)*.
- Muri, M. F. A., Utomo, H. S., & Sayyidati, R. (2019). *Search Engine Get Application Programming Interface*.
- Muslihudin, M., Oktavfianto, & Pramesta, A. (2016). *Anilisis dan perancangan sistem informasi menggunakan model terstruktur dan UML*.
- Prayama, D., Yuhefizar, & Amelia Yolanda. (2021). *Protokol HTTPS, Apakah Benar-benar Aman?*
- Rachmat, F. (2017). *Pengembangan Sistem Informasi Geografis Berbasis Node.JS untuk Pemetaan Mesin dan Tracking Engineer dengan Pemanfaatan Geolocation pada PT IBM Indonesia*.
- Solichin, A. S. K. (2017). *Pemrograman Web dengan PHP dan MySQL*.
- Somya, R., & Nathanael, T. M. E. (2019). *Pengembangan Sistem Informasi Pelatihan Berbasis Web Menggunakan Teknologi Web Service Dan Framework Laravel*.