

IMPLEMENTASI ARSITEKTUR MEVN STACK PADA SISTEM MANAJEMEN ARSIP DIGITAL FOOTAGE FILM

Adhimas Dwi Wahyu Sejati^{1*}, Vidi Sandi²

^{1,2}Politeknik Pancasakti Global Tegal; Jl. Halmahera No 1 Tegal; 0283 - 4531418

Keywords:

Archive Management;
Google Drive API;
Express.js; Vue.js; Waterfall.

Correspondent Email:

dimaswhy173@gmail.com

Abstrak. Digital Agency Mengudara Komunikasi menghadapi tantangan dalam pengelolaan aset digital berupa footage film yang terus meningkat, di mana proses pengarsipan masih bergantung pada media penyimpanan fisik yang rentan terhadap kerusakan dan kurang mendukung kolaborasi tim. Penelitian ini bertujuan untuk mengembangkan sistem manajemen arsip digital berbasis web yang mampu meningkatkan keamanan data dan efisiensi akses informasi. Metode yang digunakan adalah metode Waterfall dengan implementasi teknologi MEVN Stack (MySQL, Express.js, Vue.js, Node.js) serta integrasi Google Drive API sebagai media penyimpanan berbasis cloud. Hasil penelitian menunjukkan bahwa sistem yang dibangun mampu mengotomatisasi proses penyimpanan dan pengelolaan arsip, mempermudah pencarian data berdasarkan kategori seperti Trailer, Teaser, dan Poster, serta meningkatkan aksesibilitas data secara real-time bagi tim. Dengan demikian, sistem ini efektif dalam mengurangi risiko kehilangan data akibat kerusakan media fisik dan meningkatkan efisiensi kerja dalam pengelolaan arsip digital.



Copyright © [JITET](#) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstract. Digital Agency Mengudara Komunikasi faces challenges in managing digital assets in the form of film footage, which continue to grow alongside increasing production. The current archiving process still relies on physical storage media, making it vulnerable to damage and limiting effective team collaboration. This study aims to develop a web-based digital archive management system to improve data security and accessibility. The research employs the Waterfall method and implements MEVN Stack technologies (MySQL, Express.js, Vue.js, Node.js), integrated with the Google Drive API as a cloud-based storage solution. The results show that the developed system is capable of automating the storage and management of digital archives, enabling efficient data retrieval based on categories such as Trailers, Teasers, and Posters, and improving real-time accessibility for team members. Therefore, the system effectively reduces the risk of data loss due to physical storage damage and enhances operational efficiency in managing digital archives.

1. PENDAHULUAN

Perkembangan industri kreatif mendorong kebutuhan pengelolaan data digital yang semakin cepat, aman, dan terstruktur, khususnya pada instansi atau agensi yang memproduksi konten visual dalam volume tinggi. Digital Agency Mengudara Komunikasi sebagai salah satu agensi yang mengelola berbagai aset visual masih menghadapi kendala

dalam proses pengarsipan footage film karena penyimpanan data masih bergantung pada media fisik seperti hardisk eksternal dan flashdisk. Kondisi tersebut menimbulkan risiko kehilangan data akibat kerusakan perangkat, keterbatasan aksesibilitas, serta rendahnya efisiensi dalam proses temu kembali arsip. Penyimpanan berbasis *cloud* dinilai lebih andal dibandingkan penyimpanan fisik karena

mampu meningkatkan keamanan data, ketersediaan akses, dan mendukung kolaborasi yang lebih baik [8].

Dalam konteks manajemen aset digital, pengelolaan arsip yang terstruktur berperan penting dalam mendukung efisiensi kerja organisasi. Arsip digital tidak hanya berfungsi sebagai tempat penyimpanan, tetapi juga sebagai sarana pengelompokan, pencarian, dan distribusi informasi secara sistematis [10]. Pada lingkungan produksi film atau konten visual, kebutuhan tersebut menjadi semakin penting karena aset yang dihasilkan memiliki ukuran file besar, jumlah yang terus bertambah, dan memerlukan pengelolaan berdasarkan kategori tertentu, seperti *trailer*, *teaser*, dan *poster*. Apabila proses pengarsipan masih dilakukan secara manual tanpa sistem terpusat, maka tim akan mengalami kesulitan dalam pencarian data, pengelompokan arsip, dan akses bersama antaranggota tim.

Beberapa penelitian sebelumnya telah membahas efisiensi arsitektur MEVN *Stack* dalam pengembangan aplikasi web, pemanfaatan *cloud storage* untuk meningkatkan keamanan data, serta pembangunan sistem informasi berbasis web dengan pendekatan *Waterfall* [7], [8], [15]. Namun, penelitian-penelitian tersebut umumnya belum secara spesifik mengintegrasikan kebutuhan pengelolaan arsip digital *footage* film dengan penyimpanan berbasis *cloud* dalam satu sistem yang terpusat. Dengan demikian, masih terdapat celah penelitian dalam pengembangan sistem yang tidak hanya mampu mengelola metadata arsip, tetapi juga menghubungkan proses penyimpanan *file* digital secara langsung ke layanan *cloud* agar lebih aman, mudah diakses, dan sesuai dengan kebutuhan operasional industri kreatif.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengembangkan sistem manajemen arsip digital *footage* film berbasis web menggunakan arsitektur MEVN *Stack* dan integrasi *Google Drive API*. Sistem yang dibangun diharapkan mampu mendukung proses pengunggahan *file*, pengelompokan arsip berdasarkan film dan kategori, serta mempermudah pencarian dan akses data secara terpusat. Kebaruan penelitian ini terletak pada penerapan integrasi antara teknologi web modern dan layanan *cloud storage* untuk

kebutuhan pengarsipan digital *footage* film, sehingga dapat menjadi solusi yang lebih efektif dalam meningkatkan keamanan data, efisiensi kerja, dan aksesibilitas informasi pada *Digital Agency* Mengudara Komunikasi.

2. TINJAUAN PUSTAKA

2.1. Sistem Manajemen Arsip Digital

Sistem manajemen arsip digital merupakan proses pengelolaan rekaman data dalam format elektronik yang mencakup penciptaan, penyimpanan, hingga pendistribusian. Efisiensi sebuah organisasi sangat bergantung pada bagaimana alur informasi dikelola secara sistematis [10]. Dalam industri kreatif, manajemen arsip digital berfungsi untuk meminimalisir redundansi data dan mempercepat waktu pencarian aset yang dibutuhkan kembali oleh tim produksi.

2.2. Cloud Storage dan Keunggulan Google Drive

Cloud storage adalah model penyimpanan data di mana data digital disimpan dalam kelompok logis di *server* yang dikelola oleh pihak ketiga. Migrasi dari penyimpanan fisik ke awan (*cloud*) memberikan jaminan keamanan data yang lebih tinggi melalui protokol enkripsi dan redundansi *server* [8]. *Google Drive* dipilih sebagai media penyimpanan dalam sistem ini karena keandalan infrastrukturnya serta kemudahan integrasi melalui *Application Programming Interface* (API) yang memungkinkan sistem eksternal untuk melakukan manipulasi *file* secara otomatis [11].

2.3. Framework Express.js

Express.js adalah kerangka kerja aplikasi *backend* minimalis dan fleksibel untuk *Node.js*. *Express.js* sangat optimal untuk menangani proses I/O (*Input/Output*) yang intensif seperti *upload file* besar karena sifatnya yang *non-blocking* [3], [12]. Hal ini sangat krusial dalam sistem manajemen *footage* film di mana ukuran *file* video biasanya mencapai ratusan *megabyte* hingga *gigabyte*.

2.4. Framework Vue.js

Vue.js adalah *framework* progresif yang digunakan untuk membangun antarmuka pengguna (*frontend*). *Vue.js* memungkinkan pengembang menciptakan *Single Page Application* (SPA) yang sangat responsif [4],

[13]. Dengan menggunakan *Vue.js*, sistem manajemen arsip ini dapat memberikan pengalaman pengguna yang mulus, di mana proses manajemen kategori dan pencarian *footage* dilakukan tanpa harus memuat ulang (*reload*) halaman secara keseluruhan.

2.5. Penelitian Terdahulu (State of the art)

Penelitian ini dilakukan dengan merujuk pada beberapa penelitian terdahulu untuk memetakan posisi penelitian dan menjamin keaslian karya ilmiah ini. Selain itu, artikel JITET yang membahas pengembangan *website* berbasis *Node.js* dan metode *Waterfall* menunjukkan bahwa kombinasi teknologi web modern dan pendekatan pengembangan terstruktur masih relevan untuk membangun sistem informasi berbasis web [15]. Berikut adalah ringkasan penelitian yang relevan:

Penelitian 1 Peneliti : Yoraeni dan Kurniawan [10] Fokus : Manajemen aset digital Hasil : Meningkatkan efisiensi kerja Perbedaan : Tidak mengembangkan sistem
Penelitian 2 Peneliti : Shodiq dkk. [8] Fokus : <i>Cloud storage</i> Hasil : Keamanan data lebih tinggi Perbedaan : Tidak fokus pada manajemen arsip
Penelitian 3 Peneliti : Raharjo [3] Fokus : Backend <i>Express.js</i> Hasil : Stabil untuk trafik tinggi Perbedaan : Tidak menangani <i>footage</i> video
Penelitian 4 Peneliti : Sharma [4] Fokus : Antarmuka <i>Vue.js</i> Hasil : Aplikasi lebih responsif Perbedaan : Tidak membahas arsip digital
Penelitian 5 Peneliti : Sumudra [15] Fokus : <i>Website</i> berbasis <i>Node.js</i> dan <i>Waterfall</i> Hasil : Sistem web responsif dan berjalan sesuai spesifikasi Perbedaan : Tidak membahas pengarsipan digital <i>footage</i>

3. METODE PENELITIAN

3.1 Metode Pengembangan Sistem (Metode Waterfall)

Sesuai dengan kebutuhan pengembangan yang terstruktur, penelitian ini menerapkan model *Waterfall* (Air Terjun). Metode ini merupakan pendekatan klasik yang sistematis dan sekuensial dalam membangun perangkat lunak [2]. Tahapan-tahapan yang dilakukan adalah sebagai berikut:

3.1.1 Analisis Kebutuhan

Melakukan identifikasi mendalam mengenai fitur-fitur yang dibutuhkan, seperti manajemen kategori *footage* (*Trailer*, *Teaser*, *Poster*), mekanisme *upload* otomatis ke *Google Drive*, dan hak akses pengguna.

3.1.2 Perancangan Sistem

Membuat cetak biru sistem yang meliputi perancangan basis data menggunakan *MySQL* [1], [14], serta perancangan alur kerja menggunakan diagram UML (*Use Case*, *Activity*, dan *Sequence Diagram*).

3.1.3 Implementasi

Menerjemahkan desain ke dalam bahasa pemrograman. Pada tahap ini, *backend* dibangun menggunakan *Express.js* [3], [12], *frontend* menggunakan *Vue.js* [4], [13], dan dilakukan integrasi API *Google Drive* melalui *Google Cloud Console* [11].

3.1.4 Pengujian

Melakukan verifikasi terhadap fungsionalitas sistem menggunakan metode *Blackbox Testing* untuk memastikan seluruh fitur berjalan sesuai harapan dan bebas dari *error* logika.

3.1.5 Pemeliharaan

Tahap akhir di mana sistem diserahkan kepada pengguna dan dilakukan perbaikan jika ditemukan kendala kecil saat operasional berlangsung.

3.2 Teknik Pengumpulan Data

Untuk mendapatkan data yang akurat dalam penyusunan sistem ini, penulis menggunakan beberapa teknik:

3.2.1 Observasi

Melakukan pengamatan langsung terhadap alur pengarsipan data video di *Digital Agency* Mengudara Komunikasi guna memahami kendala dalam penggunaan media penyimpanan fisik.

3.2.2 Wawancara

Melakukan tanya jawab dengan tim produksi dan manajer operasional mengenai spesifikasi kategori *footage* yang dibutuhkan serta kendala kolaborasi antar tim.

3.2.3 Studi Pustaka

Mengumpulkan referensi dari jurnal ilmiah, buku panduan pemrograman *Node.js*, *Express.js*, *Vue.js*, serta dokumentasi resmi *Google Drive API* dan *MySQL* [11]-[14].

3.3 Alat dan Bahan Penelitian

3.3.1 Perangkat Keras

Laptop dengan spesifikasi standar pengkodean (minimal RAM 8GB).

3.3.2 Perangkat Lunak

Visual Studio Code (Editor), *MySQL (Database)*, *Postman (API Testing)*, dan *Node.js Runtime*.

3.4 Kondisi Awal Sistem

Saat ini, proses pengarsipan pada *Digital Agency* Mengudara Komunikasi dilakukan secara manual dengan memindahkan file dari kartu memori ke *hardisk* eksternal. Selama masa observasi, ditemukan bahwa tim produksi seringkali harus melakukan pengecekan manual pada setiap *hardisk* eksternal untuk menemukan klip tertentu, yang secara signifikan membuang waktu produktif tim. Hal ini menyebabkan tim produksi seringkali menghabiskan waktu lebih dari 30 menit hanya untuk mencari satu klip *footage* tertentu di antara puluhan *hardisk*.

3.5 Analisis Kebutuhan Sistem

3.5.1 Kebutuhan Fungsional

a. Sistem Autentikasi

Admin harus melakukan login untuk mengakses fitur manajemen arsip.

b. Manajemen Data Master

Sistem dapat melakukan CRUD (*Create, Read, Update, Delete*) pada data Film dan Kategori (*Trailer, Teaser, Poster*).

c. Manajemen Pengguna

Admin utama dapat menambah atau menghapus akses anggota tim kantor.

d. Integrasi Google Drive

Sistem harus mampu mengunggah file langsung ke penyimpanan awan dan

mendapatkan *sharing link* atau *file ID* secara otomatis.

e. Pencarian dan Filter

Sistem menyediakan kolom pencarian berdasarkan nama film atau kategori untuk mempercepat akses data.

3.5.2 Kebutuhan Non-Fungsional

a. Responsivitas

Antarmuka harus bisa diakses dengan nyaman melalui perangkat *mobile* maupun *desktop (Vue.js & Tailwind CSS)*.

b. Ketersediaan

Data harus tersinkronisasi 24/7 antara *database* lokal dengan *Google Drive*.

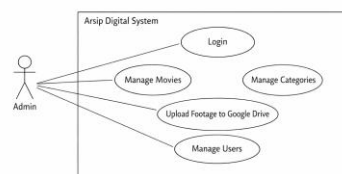
c. Efisiensi

Proses unggah harus memiliki indikator progres agar pengguna mengetahui status pengiriman file.

3.6 Perancangan Sistem dengan UML

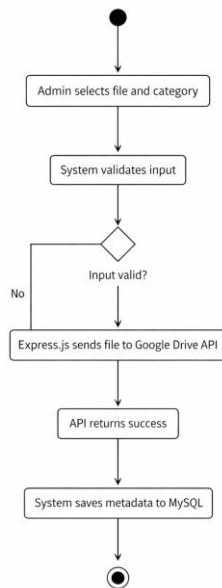
3.6.1 Use Case Diagram

Diagram ini menggambarkan hubungan antara **Aktor (Admin)** dengan fungsi-fungsi utama sistem. Admin berinteraksi dengan sistem untuk mengelola *database* film, kategori, dan user. Fitur utama yang menjadi inti adalah *Manage Footage*, di mana sistem secara otomatis akan memanggil layanan API *Google Drive* saat proses simpan data dilakukan.



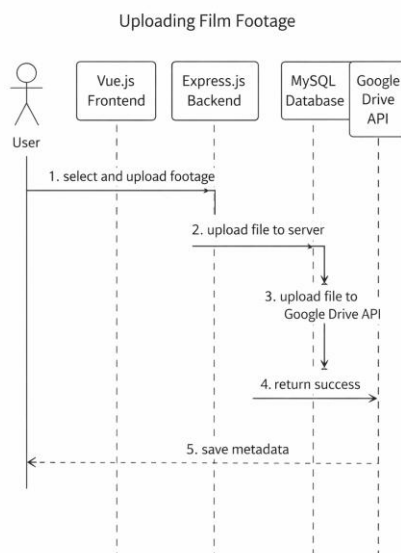
3.6.2 Activity Diagram

Diagram ini memvisualisasikan alur kerja dari sisi pengguna hingga ke sistem. Admin memilih menu *upload*, mengisi form (pilih film dan kategori), lalu memilih file. Sistem kemudian melakukan validasi; jika valid, *Express.js* akan mengirim file ke *Google Drive API*. Jika API mengembalikan respon sukses, data metadata file akan disimpan ke *MySQL* dan pengguna menerima pesan sukses di layar.



3.6.3 Sequence Diagram

Sequence Diagram ini memberikan gambaran teknis mengenai urutan pertukaran pesan antar komponen: **UI (Vue.js)**, **Router**, **Controller (Express.js)**, **GDrive Service**, dan **Database**. Diagram ini menunjukkan bagaimana sebuah file "mengalir" dari komputer lokal hingga menetap di *cloud storage* melalui *endpoint API* tertentu.



3.7 Perancangan Basis Data

Perancangan basis data pada sistem ini menggunakan *MySQL* dengan struktur yang ternormalisasi untuk menjamin integritas data. Berikut adalah spesifikasi tabel yang digunakan dalam basis data mk_film:

3.7.1 Tabel Categories

Tabel ini berfungsi untuk menyimpan kategori aset digital seperti *Trailer*, *Teaser*, atau *Poster*.

Field	Type	Null	Key	Extra
Id	Int	NO	PRI	Auto_increment
Name	Varchar(255)	NO		
Created_at	Datetime	YES		DEFAULT_GENERATED

3.7.2 Tabel Films

Tabel ini menyimpan data master proyek film atau iklan yang sedang diproduksi.

Field	Type	Null	Key	Extra
Id	Int	NO	PRI	Auto_increment
Title	Varchar(255)	NO	MUL	
Description	Text	YES		
Poster_gd_file_id	Varchar(255)	YES		
Poster_gd_link	Text	YES		

3.7.3 Tabel Users

Tabel ini digunakan untuk mengelola data administrator dan staf yang memiliki hak akses ke sistem.

Field	Type	Null	Key	Default	Extra
Id	Int	NO	PRI	NULL	Auto_increment
Name	Varchar(255)	NO		NULL	
Email	Varchar(255)	NO	UNI	NNULL	
Role	Varchar(50)	NO		Staff	

3.7.4 Tabel Footages

Tabel utama ini menyimpan relasi antara file video dengan proyek film serta menyimpan referensi ID dari *Google Drive*.

Field	Type	Null	Key	Extra
Id	Int	NO	PRI	Auto_increment
Film_id	Int	NO	MUL	Foreign Key ke Tabel Films
Category_id	Int	NO	MUL	Foreign Key ke Tabel Categories
File_name	Varchar(512)	NO		
Gd_file_id	Varchar(255)	YES		ID dari Google Drive API
File_size	Bigint	YES		

4. HASIL DAN PEMBAHASAN

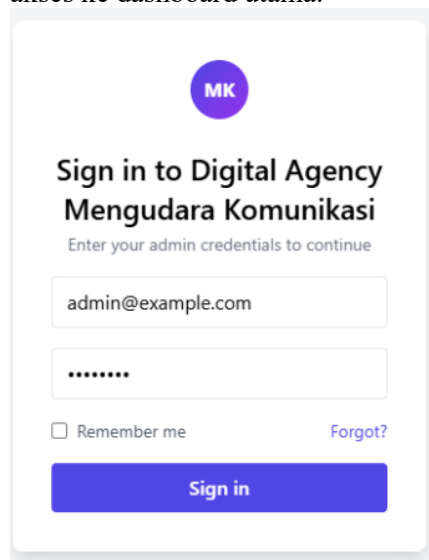
4.1 Implementasi Sistem

Implementasi sistem merupakan tahap realisasi dari rancangan yang telah disusun pada tahap sebelumnya ke dalam bentuk aplikasi berbasis web. Sistem dibangun menggunakan *Vue.js* pada sisi *frontend* untuk menghasilkan antarmuka yang responsif, *Express.js* pada sisi *backend* untuk menangani logika aplikasi dan proses pertukaran data, *MySQL* sebagai basis data utama, serta *Google Drive API* sebagai media penyimpanan *cloud*. Integrasi keempat komponen tersebut menghasilkan

sistem manajemen arsip digital yang mampu mendukung proses unggah, pengelompokan, pencarian, dan pengelolaan *footage* film secara terpusat.

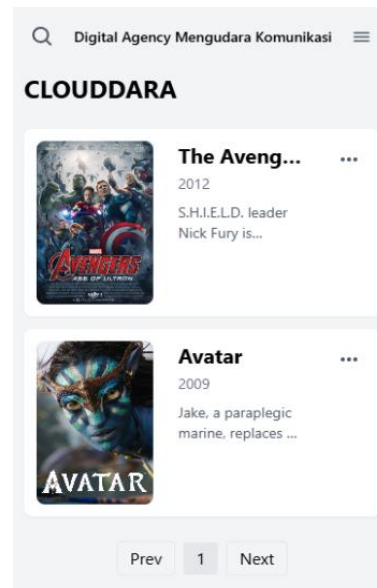
4.1.1 Antarmuka Halaman Login

Halaman *login* berfungsi sebagai gerbang awal untuk mengakses sistem. Pengguna diwajibkan memasukkan *email* dan kata sandi yang valid sebelum dapat masuk ke halaman utama. Penerapan autentikasi ini penting untuk membatasi akses hanya kepada pengguna yang memiliki hak otorisasi, sehingga keamanan data arsip dapat lebih terjaga. Dari sisi implementasi, proses validasi kredensial dilakukan pada *backend* sebelum sistem memberikan hak akses ke dashboard utama.



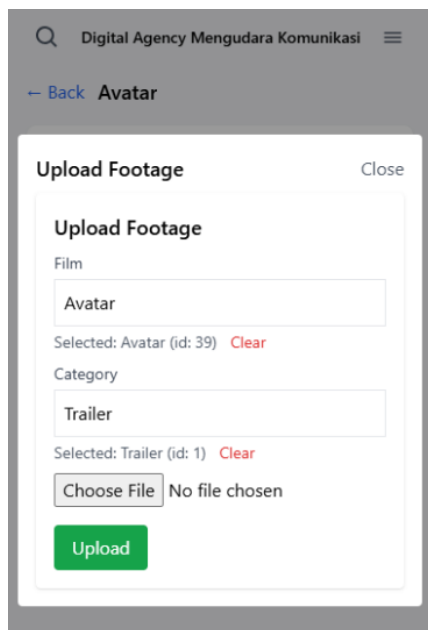
4.1.2 Antarmuka Dashboard Utama

Halaman *dashboard* utama berfungsi sebagai halaman awal setelah proses *login* berhasil dilakukan. Pada halaman ini, sistem menampilkan data film-film yang telah terdaftar sebagai bagian dari data utama dalam manajemen arsip digital. Keberadaan *dashboard* ini memudahkan admin untuk melihat daftar film yang tersedia secara cepat dan terpusat, sehingga proses pengelolaan arsip pada tahap berikutnya dapat dilakukan dengan lebih terarah. Dengan demikian, *dashboard* tidak hanya berperan sebagai halaman awal sistem, tetapi juga sebagai titik akses utama dalam pengelolaan data film.



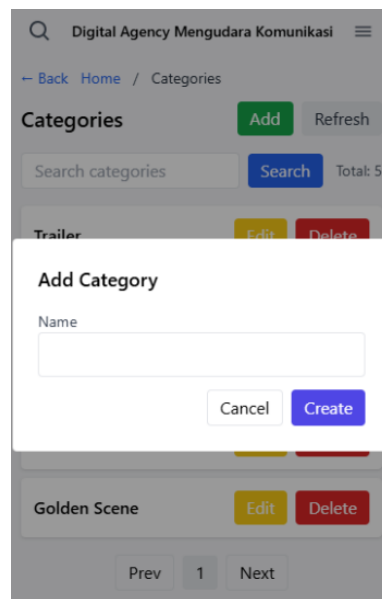
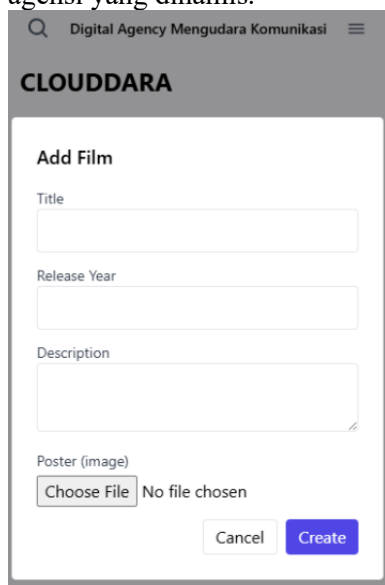
4.1.3 Antarmuka Form Upload dan Integrasi Google Drive

Fitur unggah merupakan inti dari sistem yang dibangun. Pada halaman ini, admin dapat memilih film, menentukan kategori arsip, lalu mengunggah file *footage* ke dalam sistem. Setelah pengguna menekan tombol *upload*, sistem akan memproses file melalui *backend* dan meneruskannya ke *Google Drive API*. Jika proses berhasil, sistem menyimpan metadata file, seperti nama file, kategori, relasi film, dan *file ID* dari *Google Drive* ke dalam basis data *MySQL*. Mekanisme ini menunjukkan bahwa sistem tidak hanya berfungsi sebagai katalog arsip, tetapi juga sebagai penghubung antara pengelolaan metadata lokal dan penyimpanan *file* berbasis *cloud*.



4.1.4 Antarmuka Manajemen Data Master

Sistem juga menyediakan halaman manajemen data master yang mencakup pengelolaan data film dan kategori. Melalui fitur ini, admin dapat menambah, mengubah, dan menghapus data sesuai kebutuhan. Ketersediaan fitur manajemen data master memberikan fleksibilitas yang tinggi karena penambahan proyek baru maupun kategori baru dapat dilakukan langsung melalui sistem tanpa perlu melakukan perubahan kode program. Dengan demikian, sistem menjadi lebih adaptif terhadap kebutuhan operasional agensi yang dinamis.



4.2 Hasil Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode *blackbox testing* yang berfokus pada pengujian fungsi tanpa meninjau struktur kode internal. Pengujian ini bertujuan untuk memastikan bahwa setiap masukan yang diberikan pengguna menghasilkan keluaran yang sesuai dengan kebutuhan sistem. Berdasarkan hasil pengujian, fitur-fitur utama seperti *login*, pengelolaan data film, pengelolaan kategori, unggah *file*, integrasi *Google Drive API*, dan pencarian arsip dapat berjalan sesuai dengan fungsi yang dirancang.

Hasil pengujian menunjukkan bahwa proses autentikasi mampu membatasi akses pengguna secara tepat, fitur CRUD pada data master dapat dijalankan tanpa kendala fungsional, dan proses unggah berhasil menyimpan file ke *Google Drive* sekaligus mencatat metadata *file* ke dalam basis data. Selain itu, fitur pencarian berdasarkan nama film atau kategori juga berjalan dengan baik sehingga mempermudah proses temu kembali arsip. Secara umum, seluruh fungsi inti sistem telah dapat digunakan untuk mendukung proses pengarsipan digital di *Digital Agency Mengudara Komunikasi*.

4.3 Pembahasan

Hasil implementasi menunjukkan bahwa sistem yang dibangun mampu menjawab permasalahan utama yang sebelumnya terjadi pada proses pengarsipan manual.

Sebelum adanya sistem, penyimpanan *footage* masih bergantung pada *hardisk* eksternal dan media fisik lain yang rawan rusak, sulit diakses bersama, dan menyulitkan proses pencarian arsip. Setelah sistem diterapkan, proses pengelolaan arsip menjadi lebih terstruktur karena setiap file disimpan berdasarkan relasi film dan kategori tertentu, serta dapat diakses kembali melalui mekanisme pencarian dalam aplikasi. Hal ini menunjukkan bahwa digitalisasi arsip tidak hanya berpengaruh pada keamanan penyimpanan, tetapi juga pada efisiensi alur kerja tim.

Dari sisi arsitektur, penggunaan MEVN *Stack* memberikan dukungan yang baik terhadap kebutuhan sistem. *Express.js* pada *backend* mampu menangani proses unggah *file* dan komunikasi dengan layanan eksternal secara efisien, sedangkan *Vue.js* membantu membentuk antarmuka yang responsif dan mudah digunakan. Temuan ini selaras dengan penelitian sebelumnya yang menyatakan bahwa arsitektur berbasis *JavaScript* modern efektif untuk pengembangan aplikasi web yang responsif dan terintegrasi [7], [15]. Selain itu, penggunaan *Google Drive API* sebagai media penyimpanan juga memperkuat aspek keamanan dan ketersediaan data, sejalan dengan penelitian yang menyebutkan bahwa *cloud storage* memberikan keunggulan dalam perlindungan data dan kemudahan akses dibandingkan media penyimpanan fisik [8], [10].

Jika dibandingkan dengan penelitian terdahulu, sistem yang dibangun pada penelitian ini memiliki kontribusi praktis pada integrasi antara pengelolaan metadata arsip dan penyimpanan *file footage* dalam satu *platform*. Penelitian sebelumnya lebih banyak membahas efisiensi arsitektur web, keamanan integrasi API, atau pemanfaatan *cloud storage* secara umum [6]-[10], [15], sedangkan penelitian ini menempatkan integrasi tersebut secara langsung pada kebutuhan pengarsipan digital *footage film*. Dengan demikian, kontribusi utama penelitian ini terletak pada penerapan solusi yang lebih kontekstual terhadap kebutuhan

industri kreatif, khususnya dalam pengelolaan aset digital berukuran besar.

Meskipun demikian, hasil pengujian juga menunjukkan bahwa performa sistem masih dipengaruhi oleh kestabilan koneksi internet, terutama pada saat unggah *file* berukuran besar. Ketergantungan terhadap *bandwidth* unggah menjadi salah satu keterbatasan implementasi, karena semakin besar ukuran *file* yang diproses maka semakin tinggi pula waktu tunggu yang dibutuhkan. Oleh sebab itu, meskipun sistem telah berhasil memenuhi kebutuhan fungsional, pengembangan lanjutan masih diperlukan, misalnya melalui optimasi *multi-upload*, *preview* video secara langsung, serta penambahan notifikasi proses unggah agar sistem semakin mendukung kebutuhan kerja tim secara menyeluruh.

Secara keseluruhan, hasil penelitian menunjukkan bahwa sistem manajemen arsip digital berbasis MEVN *Stack* dan *Google Drive API* mampu menjadi solusi yang layak untuk meningkatkan efisiensi, keamanan, dan aksesibilitas pengelolaan *footage* film. Sistem ini tidak hanya memindahkan proses penyimpanan dari media fisik ke *cloud*, tetapi juga menghadirkan mekanisme pengelolaan arsip yang lebih terstruktur, terpusat, dan mudah digunakan dalam mendukung aktivitas operasional *Digital Agency* Mengudara Komunikasi.

5. KESIMPULAN

5.1 Kesimpulan

Berdasarkan hasil penelitian, pengembangan, dan pengujian yang telah dilakukan pada Sistem Manajemen Arsip Digital *Footage* Film di *Digital Agency* Mengudara Komunikasi, maka dapat ditarik beberapa kesimpulan sebagai berikut:

- a. Sistem berhasil dibangun menggunakan arsitektur MEVN *Stack* (*MySQL*, *Express.js*, *Vue.js*, *Node.js*) yang terintegrasi secara langsung dengan *Google Drive API*, sehingga mampu memindahkan penyimpanan

aset dari media fisik ke penyimpanan awan (*cloud*).

- b. Metode pengembangan *Waterfall* yang digunakan terbukti efektif dalam memastikan setiap tahapan, mulai dari analisis kebutuhan hingga pengujian *Blackbox*, dilakukan secara terstruktur dan terdokumentasi dengan baik.
- c. Implementasi fitur pengkategorian (*Trailer, Teaser, Poster*) dan fitur pencarian telah berhasil mengatasi masalah inefisiensi pencarian data yang sebelumnya dialami oleh tim produksi.
- d. Pengujian fungsionalitas menunjukkan bahwa sistem dapat melakukan sinkronisasi data antara *database* lokal dengan *Google Drive* secara otomatis, yang meningkatkan keamanan data dari risiko kerusakan perangkat keras (*hardisk/flashdisk*).

5.2 Saran

Meskipun sistem ini telah berfungsi dengan baik sesuai dengan tujuan penelitian, terdapat beberapa saran untuk pengembangan lebih lanjut agar sistem menjadi lebih sempurna:

- a. Pengembangan Fitur Streaming
Diharapkan untuk pengembangan selanjutnya, sistem dapat memutar video (*preview*) secara langsung di dalam aplikasi tanpa harus mengunduhnya terlebih dahulu dari *Google Drive*.
- b. Sistem Notifikasi
Menambahkan fitur notifikasi melalui *email* atau *WhatsApp* ketika proses unggah file besar telah selesai dilakukan oleh sistem.
- c. Peningkatan Keamanan
Menambahkan fitur *Two-Factor Authentication* (2FA) untuk meningkatkan keamanan akses akun administrator mengingat data yang disimpan merupakan aset intelektual perusahaan yang berharga.
- d. Optimasi Multi-Upload
Menambahkan fitur unggah banyak file (*bulk upload*) secara sekaligus untuk lebih mengoptimalkan waktu kerja admin.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] N. Huda, Manajemen Basis Data Menggunakan MySQL. Yogyakarta, Indonesia: Andi Publisher, 2022.
- [2] R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach, 9th ed. New York, NY, USA: McGraw-Hill Education, 2021.
- [3] B. Raharjo, Pemrograman Web dengan Node.js dan Express.js. Bandung, Indonesia: Informatika, 2024.
- [4] A. Sharma, Vue.js 3 Design Patterns and Best Practices. Birmingham, U.K.: Packt Publishing, 2023.
- [5] Sutarman, Pengantar Teknologi Informasi. Jakarta, Indonesia: Bumi Aksara, 2023.
- [6] R. Pratama and M. Sari, "Penerapan OAuth 2.0 untuk Keamanan Integrasi API Pihak Ketiga pada Aplikasi Berbasis Web," Jurnal Teknik Informatika (JUTIF), vol. 5, no. 3, pp. 201-210, 2022.
- [7] B. Setiawan and T. Cahyono, "Efisiensi Arsitektur MEVN Stack dalam Pengembangan Aplikasi Single Page Application (SPA)," Jurnal Komputer dan Teknologi, vol. 15, no. 4, pp. 88-97, 2024.
- [8] A. Shodiq et al., "Implementasi Google Drive API Sebagai Media Penyimpanan Data Cadangan Otomatis," Jurnal Riset Teknologi Informasi (JRTI), vol. 10, no. 2, pp. 112-125, 2025.
- [9] S. Widodo, "Optimasi Pengolahan Data Footage Film Berbasis Web Menggunakan Framework Express.js," Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (JPTIIK), vol. 7, no. 2, pp. 34-42, 2023.
- [10] A. Yoraeni and D. Kurniawan, "Analisis Manajemen Aset Digital pada Industri Kreatif Menggunakan Media Penyimpanan Cloud," Jurnal Sistem Informasi, vol. 12, no. 1, pp. 45-58, 2023.
- [11] Google for Developers, "Google Drive API overview," 2026. [Online]. Available: <https://developers.google.com/workspace/drive/api/guides/about-sdk>. Accessed: Apr. 13, 2026.
- [12] Express.js, "Express 5.x API Reference," 2026. [Online]. Available: <https://expressjs.com/en/5x/api.html>. Accessed: Apr. 13, 2026.

- [13] Vue.js Core Team, "Introduction | Vue.js," 2026. [Online]. Available: <https://vuejs.org/guide/introduction>. Accessed: Apr. 13, 2026.
- [14] Oracle Corporation, "MySQL 8.4 Reference Manual," 2026. [Online]. Available: <https://dev.mysql.com/doc/en/>. Accessed: Apr. 13, 2026.
- [15] D. G. Sumudra, "Rancang Bangun Website Terpadu Menggunakan Framework ReactJS untuk Meningkatkan Tampilan yang Menarik dan Responsif pada Masjid Jami' At-Taqwa," Jurnal Informatika dan Teknik Elektro Terapan, vol. 13, no. 3S1, 2025, doi: 10.23960/jitet.v13i3S1.7561.