

Pengembangan Aplikasi Backend Berbasis Web Untuk Manajemen Konten Digital Signage Dan Jadwal Laboratorium Podcast

Muhammad Habib Yusuf¹

Teknologi Rekayasa Multimedia

Universitas Telkom

Bandung, Indonesia

habib Yusuf@student.telkomuniversity.
ac.id

Ady Purna Kurniawan²

Teknologi Rekayasa Multimedia

Universitas Telkom

Bandung, Indonesia

adypurnakurniawan@telkomuniversity.
ac.id

Rio Korio Utoro³

Teknologi Rekayasa Multimedia

Universitas Telkom

Bandung, Indonesia

korioutoro@telkomuniversity.ac.id

Abstrak — Pengembangan aplikasi backend berbasis web untuk manajemen konten digital signage dan jadwal laboratorium podcast dilakukan sebagai solusi atas kebutuhan sistem informasi terintegrasi di lingkungan Fakultas Ilmu Terapan Universitas Telkom. Aplikasi bernama "Podsync" ini menyediakan fitur utama berupa manajemen konten digital signage dengan operasi CRUD untuk link YouTube yang dilengkapi auto-generate QR code, integrasi Google Sheets API untuk sinkronisasi data peminjaman laboratorium secara real-time, sistem broadcast management untuk kontrol status siaran langsung on-air/off-air, serta dashboard analytics untuk visualisasi data penggunaan laboratorium dengan berbagai grafik dan metrik performa. Pengembangan dilakukan menggunakan metodologi Agile-Scrum selama empat sprint pada periode April hingga Juni 2025 dengan teknologi Node.js, Express.js, MySQL, dan React.js. Hasil implementasi menunjukkan aplikasi ini mampu meningkatkan efisiensi pengelolaan jadwal laboratorium hingga 90%, meminimalisir human error dalam input data manual, serta mempercepat distribusi informasi kepada sivitas akademika melalui sistem terintegrasi. Pengujian User Acceptance Testing dengan metode blackbox menunjukkan keberhasilan 100% dari 32 poin pengujian. Dengan sistem terintegrasi ini, pengelolaan fasilitas laboratorium podcast dan penyampaian informasi digital signage menjadi lebih efektif, efisien, dan mudah diakses.

Kata kunci — manajemen konten, backend, Google Sheets API.

Abstract — The development of a web-based backend application for digital signage content management and podcast laboratory scheduling was carried out to address the need for an integrated information system at the Faculty of Applied Sciences, Telkom University. The application, named "Podsync", offers comprehensive features including digital signage content management with CRUD operations for YouTube links equipped with auto-generated QR codes, Google Sheets API integration for real-time laboratory booking data synchronization, broadcast

management system for live streaming on-air/off-air status control, and dashboard analytics for laboratory usage data visualization through various charts and performance metrics. The development process utilized Agile-Scrum methodology over four sprints from April to June 2025, implementing technologies such as Node.js, Express.js, MySQL, and React.js. Implementation results demonstrate that the application improves laboratory scheduling management efficiency by up to 90%, minimizes human error in manual data entry, and accelerates information distribution to the academic community through an integrated system. User Acceptance Testing using blackbox methodology showed 100% success rate across 32 testing points. With this integrated system, the management of podcast laboratory facilities and digital signage information delivery becomes more effective, efficient, and easily accessible for all stakeholders.

Keywords — content management, backend, Google Sheets API.

I. PENDAHULUAN

Kelompok Keilmuan Applied Digital, Business, Entrepreneurship & Tourism (KK DBEsT) di Fakultas Ilmu Terapan Universitas Telkom fokus pada pengembangan bisnis digital yang sesuai dengan kebutuhan industri [1]. Dengan meningkatnya pengguna media sosial di Indonesia mencapai 170 juta pada Januari 2021 (naik 6,3% dari tahun sebelumnya) [2]. KK DBEsT menyediakan posisi magang sebagai konten kreator di S1 Terapan Teknologi Rekayasa Multimedia dengan penempatan di laboratorium podcast.

Laboratorium podcast sebagai fasilitas penunjang akademik memerlukan sistem penyampaian informasi yang efektif untuk mengoptimalkan penggunaan dan *monitoring*.

Tantangan utama mencakup pengelolaan jadwal, status siaran langsung (on-air), dan distribusi konten pada digital signage. Mengingat sistem multimedia terbukti dapat meningkatkan efisiensi manajemen fasilitas publik hingga 90%, dibutuhkan solusi yang dapat memenuhi kebutuhan secara real-time [3][4].

Proyek ini menghadirkan aplikasi backend "Podsync" yang terintegrasi dengan aplikasi "Digital Signage Laboratorium Podcast" menggunakan *database* terpusat. Aplikasi Podsync memungkinkan operator mengelola data peminjaman, status siaran, dan konten digital signage dengan integrasi Google Sheets API untuk sinkronisasi data *real-time*. Sistem ini bertujuan memperjelas dan mempercepat penyampaian informasi kepada sivitas akademika melalui platform terintegrasi dengan CMS dan API yang konsisten dengan penelitian tentang pentingnya API dalam sinkronisasi aplikasi *frontend* dan *backend* [3][4].

A. Rumusan Masalah dan Solusi

Berdasarkan analisis permasalahan yang sudah diangkat dari latar belakang, terdapat beberapa permasalahan utama yang memerlukan solusi, antara lain:

1. Belum adanya sistem terintegrasi untuk mengelola informasi jadwal peminjaman dan siaran langsung laboratorium podcast secara otomatis
2. Belum tersedianya media *input* data untuk mengelola konten digital signage aktivitas laboratorium.

Untuk mengatasi masalah tersebut, dikembangkan aplikasi web backend "Podsync" dengan empat fitur utama:

1. Sistem Manajemen Konten Digital Signage yang menyediakan operasi CRUD untuk mengelola konten berupa *link* YouTube dengan auto-generate QR code.
2. Integrasi Google Sheets API untuk sinkronisasi real-time data jadwal penggunaan laboratorium.
3. Broadcast Management System: Berfungsi sebagai sistem untuk kontrol status *on-air* dan *off-air* penggunaan laboratorium podcast.
4. Dashboard Analytics yang mengimplementasikan visualisasi data menggunakan *charts* dan *metrics* untuk analisis penggunaan laboratorium podcast.

B. Tujuan

Pengembangan aplikasi web *backend* "Podsync" bertujuan untuk, antara lain:

1. Tujuan Umum:
Mengembangkan sistem terintegrasi yang mengelola konten digital signage dan jadwal laboratorium podcast dengan database untuk aplikasi "Digital Signage Laboratorium Podcast".
2. Tujuan Khusus:

- a. Membangun sistem manajemen konten *digital signage* dengan operasi CRUD menggunakan *link* YouTube yang dilengkapi fitur *auto-generate* QR code
- b. Mengimplementasikan integrasi *real-time* dan manual dengan Google Sheets API untuk sinkronisasi data peminjaman laboratorium
- c. Mengembangkan sistem manajemen broadcast untuk kontrol status on-air/off-air laboratorium podcast.
- d. Membangun dashboard analytics dengan visualisasi data untuk menganalisis penggunaan laboratorium dan performa sistem.

C. Batasan Masalah

Agar dalam pengembangan aplikasi ini lebih terarah, berikut beberapa batasan masalah untuk pengembangan aplikasi web "Podsync":

1. Aplikasi yang dikembangkan hanya berfokus pada sistem *backend* dan manajemen database untuk aplikasi "Digital Signage Laboratorium Podcast".
2. Aplikasi ini tidak mencakup pengembangan perangkat keras, seperti layar *digital signage* atau sistem mikroprosesor untuk menampilkan konten.
3. Pengelolaan data hanya mencakup informasi peminjaman laboratorium podcast, status on-air, dan analytics untuk visualisasi data.
4. Integrasi dengan Google Sheets hanya digunakan untuk pengambilan data peminjaman.

D. Batasan Luaran

Bentuk batasan luaran dari pengembangan aplikasi web "Podsync" antara lain:

1. Aplikasi *backend* "Podsync" berbasis web yang berfungsi sebagai pengelola jadwal peminjaman, status siaran, dan konten *digital signage*.
2. *Database* terpusat yang digunakan oleh dua aplikasi agar informasi yang ditampilkan secara manual dan otomatis.
3. Sistem integrasi Google Sheets API mempermudah pengambilan data peminjaman laboratorium podcast.
4. Dokumentasi teknis yang mencakup perancangan sistem, implementasi, serta hasil pengujian.

II. KAJIAN TEORI

A. Deskripsi Organisasi

Kelompok Keilmuan Applied Digital, Business, Entrepreneurship & Tourism (KK DBEsT) di Fakultas Ilmu Terapan Universitas Telkom berfokus pada pengembangan bisnis digital dan kewirausahaan berbasis teknologi yang relevan dengan kebutuhan industri dan akademik. KK

DBEsT berperan dalam program magang, penelitian, dan pengabdian sebagai penghubung akademisi, praktisi industri, dan mahasiswa, dengan program magang multimedia yang mendukung digitalisasi dan inovasi melalui keterampilan konten kreator dan multimedia interaktif.

Fakultas Ilmu Terapan merupakan fakultas vokasi yang menyelenggarakan pendidikan praktik industri dengan komposisi 60% praktik dan 40% teori, memiliki visi menjadi "National Excellence in Entrepreneurial Vocational Faculty" pada 2028. Program studi dirancang berdasarkan Outcome Based Education (OBE) dengan kurikulum berbasis proyek dan riset terapan, serta program strategis meliputi perkuliahan berbasis industri, sertifikasi kompetensi, mobility program internasional, dan ekosistem wirausaha mahasiswa.

KK DBEsT menjadi tempat pelaksanaan program magang dan riset tugas akhir, khususnya pada Program Studi S1 Terapan Teknologi Rekayasa Multimedia yang fokus mengembangkan keterampilan teknis dan kreatif di bidang desain grafis, produksi video, animasi, *game*, AR/VR, dan konten digital untuk menjawab tantangan industri kreatif multimedia interaktif.

B. Deskripsi Pekerjaan

Selama program magang 2 semester di S1 Terapan Teknologi Rekayasa Multimedia yang bertempat di Laboratorium Podcast Fakultas Ilmu Terapan, diberikan penugasan sebagai operator dalam kegiatan kelas besar dan kelas umum, kameramen dan floor director, pembuat konten *feed* dan *reels* Instagram program studi, penulis konten artikel website program studi, dan operator Laboratorium Podcast Fakultas Ilmu Terapan.

C. Solusi Yang Pernah Ada

Berikut akan dipaparkan beberapa solusi yang pernah ada dan relevan dengan pengembangan aplikasi web "Podsync", antara lain:

1. IsLab (Multimedia-based Information System for Laboratory).
Sistem informasi laboratorium berbasis multimedia, terbukti meningkatkan efisiensi manajemen hingga 90,3% melalui integrasi teks, grafik, dan akses berbasis kebutuhan pengguna, serta layanan terpersonalisasi sesuai peran pengguna. [5]. ISLab menunjukkan bahwa layanan laboratorium dapat ditingkatkan secara signifikan dengan menambahkan konten multimedia.
2. Pembuatan Media Promosi Berbasis Website Menggunakan CMS WordPress untuk Meningkatkan Minat Beli pada CV. Utama Karya Construction Malang.
Penggunaan CMS web untuk media promosi perusahaan terbukti meningkatkan minat beli pelanggan dan menambah fleksibilitas promosi

digital. [6]. Solusi serupa pada aplikasi Podsync menunjukkan relevansi pengelolaan konten berbasis CMS dan integrasi sistem digital signage.

3. Analisis dan Pengembangan Sistem Pengelolaan Digital Signage berbasis Media TV pada Aplikasi Perguruan Tinggi.

Dalam kasus ini bahwa pengembangan sistem informasi pengelolaan *digital signage* berbasis web yang menggunakan *framework* Codeigniter. Tujuan dari sistem ini adalah untuk menggantikan metode manual yang digunakan sebelumnya untuk mengelola konten yang ditampilkan pada media televisi atau monitor di lingkungan perguruan tinggi [7].

D. Teori Penunjang

Pemahaman terhadap teori-teori penunjang menjadi landasan pengembangan aplikasi web "Podsync" yang menggabungkan database terpusat, Google Sheets API, dan metode Agile-Scrum.

1. Integrasi Manajemen Basis Data Terpusat, Google Sheets API, dan Metode Agile-Scrum.
Aplikasi PodSync berbasis pada teknologi *database* terpusat yang terintegrasi dengan API kontemporer seperti Google Sheets. Metode Agile-Scrum menjadi dasar pengembangan aplikasi ini, yang memungkinkan pengelolaan data yang efektif, berskala, dan sesuai dengan kebutuhan klien. Sistem informasi yang responsif terhadap kebutuhan pengguna laboratorium podcast dibuat dengan menerapkan tiga komponen ini. [8] [9].
2. Manajemen Basis Data Terpusat.
Menggunakan repositori tunggal dengan struktur tabel relasional yang terhubung melalui *primary key* dan *foreign key* [10]. Penelitian menunjukkan peningkatan efisiensi operasional hingga 40% pada UMKM di Jawa Timur [8]. Dengan memungkinkan pembaruan data di seluruh unit secara *real-time*, sistem terpusat mengurangi kemungkinan inkonsistensi sistem [11].
3. Google Sheets API sebagai Automasi Pembukuan.
Terbukti mengurangi kesalahan *input* manual sebesar 82% dan mempercepat analitik laporan dari 4,7 menit menjadi 38 detik berdasarkan penelitian 15 kafe di Bandung [9] [12].
4. Metodologi Agile-Scrum dalam Pengembangan Sistem.
Dengan siklus *sprint* dua minggu mencapai tingkat keberhasilan 76% dibandingkan *waterfall* (34%). Metode ini meliputi tahapan: User Story, Product Backlog, Sprint Planning, Daily Scrum, serta Sprint Review dan Retrospective.
5. Pemusatan Teknologi dalam Aplikasi Web "Podsync"

- a. Desain Arsitektur Hibrida: menggunakan pola berlapis dengan Google Sheets sebagai penyimpanan permanen dan *database* terpusat untuk transaksi real-time [9] [12]. Mekanisme *caching* Redis terbukti mengurangi beban API sebesar 28% berdasarkan simulasi 1.000 user. [15].
- b. Alur Kerja Agile-Scrum: Metode ini memungkinkan pengelolaan proyek yang adaptif dan kolaboratif melalui sprint dengan tahapan:
 - a) User Story: Merupakan deskripsi fungsionalitas dari perspektif pengguna sebagai dasar *product backlog* [16].
 - b) Product Backlog: Berisi daftar kebutuhan sistem berprioritas yang dikelola *product owner* [16].
 - c) Sprint Planning dan Sprint Backlog: Perencanaan item *backlog* yang akan dikerjakan dalam sprint [16] [17].
 - d) Daily Scrum: Pertemuan harian untuk evaluasi progres dan hambatan [16].
 - e) Sprint Review dan Retrospective: Presentasi hasil kepada *stakeholder* dan evaluasi proses untuk perbaikan sprint berikutnya [17].
- c. Strategi Pemindahan Data atau Migrasi Data: Menggunakan phased migration dengan *dual-write mechanism* selama 4-6 *sprint* untuk menjaga konsistensi antara Google Spreadsheet dan *database* terpusat. Teknik ini berhasil diterapkan pada 89% proyek UMKM tanpa downtime [12][8].
- d. Penerapan Metode Blackbox untuk User Acceptance Testing (UAT): Metode pengujian ini berfokus pada pengujian fungsionalitas berdasarkan *input-output* tanpa memperhatikan struktur kode internal. [18].

III. METODE

A. Metodologi Pengembangan Aplikasi

Pengembangan aplikasi web "Podsync" menggunakan pendekatan Agile-Scrum yang memberikan keuntungan berupa fleksibilitas tinggi, peningkatan kualitas produk, iterasi berkelanjutan, komunikasi tim efektif, *feedback* responsif, dan frekuensi rilis lebih sering. Metodologi ini diimplementasikan melalui enam tahap utama:

1. Sprint Planning: Mengidentifikasi fitur dan membagi tugas ke dalam *backlog*.

2. Sprint Execution & Development: Menggunakan Node.js (Express.js), MySQL, Google Sheets API, dan Sequelize ORM.
3. Daily Standup Meeting: Melakukan pertemuan singkat untuk membahas kemajuan dan tantangan.
4. Sprint Review & Testing: Menggunakan metode blackbox untuk memastikan fungsi berjalan sesuai rencana.
5. Sprint Retrospective: Mengevaluasi dan melakukan peningkatan efisiensi.
6. Deployment dan Maintenance: Pembaruan secara bertahap dengan pembaruan berkelanjutan.

Keunggulan menggunakan metode ini untuk pengembangan aplikasi web "Podsync", sebagai berikut:

1. Adaptif Terhadap Perubahan,
2. Kolaboratif,
3. Iteratif.

Pendekatan dengan metode Agile-Scrum sangat efektif untuk pengembangan aplikasi "Podsync" karena mampu mengakomodasi perubahan kebutuhan, mempercepat pengembangan, dan dapat meningkatkan kualitas produk melalui kolaborasi dan evaluasi berkelanjutan [16] [17].

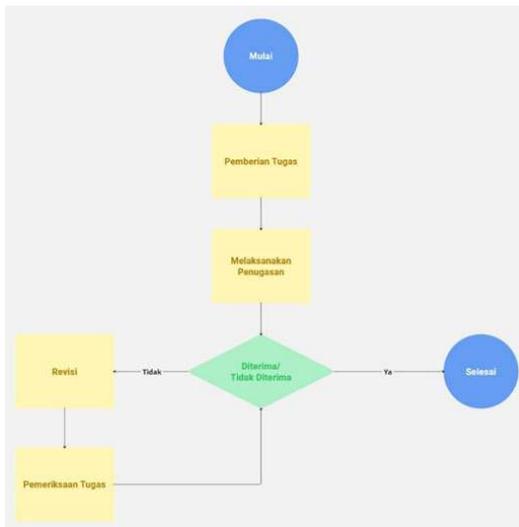


GAMBAR 1
Alur Metode Agile-Scrum

Setelah pengembangan aplikasi web "Podsync" selesai, tahap akhir dilakukan pengujian User Acceptance Testing (UAT) untuk memastikan sistem memenuhi kebutuhan pengguna. UAT menggunakan pendekatan *blackbox* untuk menguji fungsionalitas aplikasi meliputi navigasi menu, pengelolaan data, dan alur transaksi dari sudut pandang pengguna akhir, sehingga memberikan validasi menyeluruh terhadap performa aplikasi dalam kondisi penggunaan nyata [20].

B. Analisis Sistem

Pengembangan aplikasi web "Podsync" diselaraskan dengan sistem alur kerja magang di KK DBEsT sebagai konten kreator S1 Terapan Teknologi Rekayasa Multimedia yang menerapkan metode serupa dengan Agile-Scrum, memberikan fleksibilitas yang baik untuk memenuhi penugasan selama kegiatan magang berlangsung.



GAMBAR 2 Alur Kerja Magang

C. Analisis Kebutuhan Sistem

Berdasarkan hasil analisis, untuk penentuan fitur-fitur yang akan diterapkan pada aplikasi web “Podsync” bahwasanya belum adanya sistem terintegrasi untuk mengelola media penyampaian informasi jadwal peminjaman dan siaran langsung laboratorium podcast secara otomatis dan belum adanya media *input* data untuk mengelola konten *digital signage* untuk keperluan informasi aktivitas laboratorium podcast. Berikut kebutuhan sistem yang dapat diterapkan berdasarkan permasalahan tersebut:

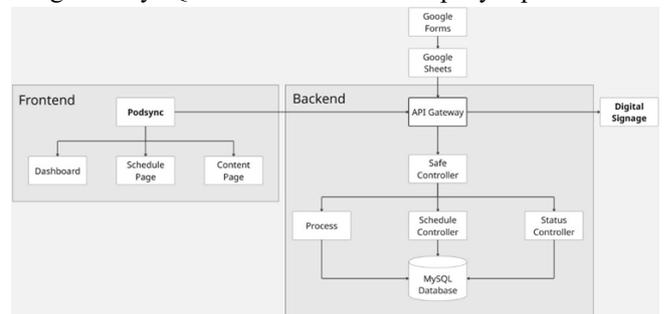
TABEL 1 Analisis Kebutuhan Sistem Aplikasi Web “Podsync”

No.	Kebutuhan Utama Sistem	Penjelasan	Output yang Diharapkan
1	Sinkronisasi Google Sheets Otomatis	Sinkronisasi otomatis data dari Google Sheets (hasil Google Forms) setiap 5 menit, dengan handling data dan konflik	<ul style="list-style-type: none"> • Data tersinkron <i>real-time</i> • Notifikasi status • Log sinkronisasi
2	Broadcast Management System	Sistem kontrol siaran <i>on-air/off-air</i> otomatis berdasarkan jadwal atau manual oleh admin	<ul style="list-style-type: none"> • Kontrol manual & otomatis • Status <i>real-time</i> • Log histori • Integrasi Unity
3	Content Management System	Penngelolaan konten digital signage termasuk pembuatan, penjadwalan, dan QR Code	<ul style="list-style-type: none"> • Upload media • Edit media • Penjadwalan • Preview & QR Code otomatis

		untuk URL media	
4	Unity Digital Signage Integration	Integrasi tampilan <i>real-time</i> dengan Unity engine via HTTP API, termasuk offline mode	<ul style="list-style-type: none"> • Display <i>real-time</i> • Transisi halus • Mode offline • Refresh otomatis 30 detik
5	Database Management System	Sistem database yang handal dengan relasi, indexing, soft delete, dan penanganan zona waktu	<ul style="list-style-type: none"> • Database MySQL • Indexing & relasi tabel • Soft delete • Zona waktu Asia/Jakarta

D. Perancangan Arsitektur Sistem.

Berdasarkan hasil analisis, sistem dirancang menggunakan pendekatan 3-tier architecture yang terdiri dari Presentation Layer dengan React frontend untuk user interface, Application Layer menggunakan Node.js/Express.js untuk logika aplikasi, dan Data Layer dengan MySQL database untuk penyimpanan data.

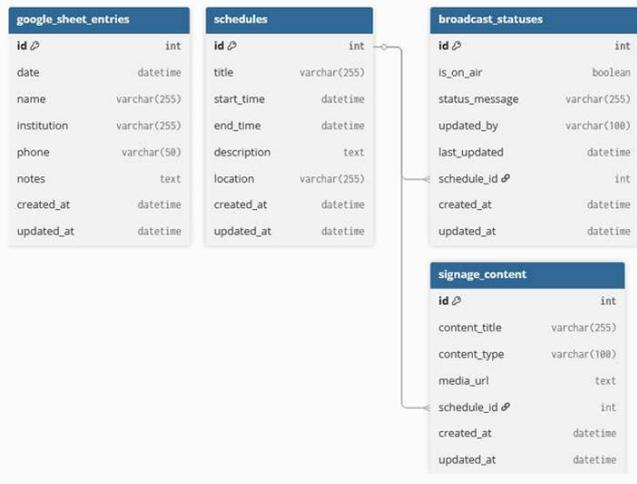


GAMBAR 3 Diagram Arsitektur Sistem "Podsync"

E. Perancangan Database.

Database dirancang menggunakan *Entity Relationship Diagram* (ERD) dengan 4 tabel utama:

1. ``google_sheet_entries``: Data mentah dari Google Forms.
2. ``schedules``: Jadwal lab yang telah diproses.
3. ``broadcast_statuses``: History status *broadcast*.
4. ``signage_content``: Konten digital signage.



GAMBAR 4 ERD Database "Podsync"

Tabel `google_sheet_entries` menjadi sumber data utama yang menyimpan informasi peminjaman meliputi nama peminjam, NIP/NIM, unit/prodi, keperluan, tanggal dan jam peminjaman. Tabel `schedules` memproses data mentah tersebut menjadi jadwal terstruktur untuk aplikasi "Digital Signage Laboratorium Podcast".

Sistem broadcast dikelola melalui tabel `broadcast_statuses` yang mencatat riwayat status *on-air/off-air* secara *real-time*, baik otomatis berdasarkan jadwal maupun manual oleh admin. Tabel `signage_content` menyediakan konten kegiatan laboratorium podcast dengan fitur *auto-generate* QR Code dan sistem penjadwalan tampilan.

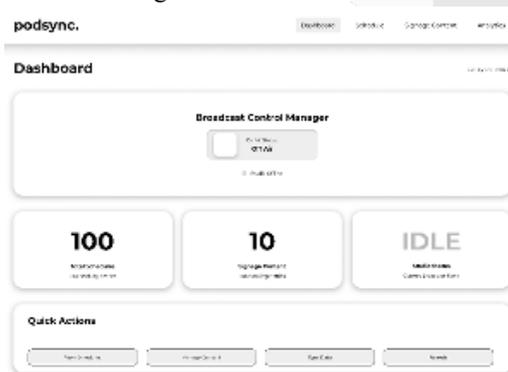
Keunggulan desain *database* ini meliputi pemisahan fungsi yang jelas, fleksibilitas relasi dengan *foreign key*, sistem audit trail lengkap, dan optimasi performa melalui *indexing* yang tepat.

F. User Interface Design.

1. Wireframe Design.

Perancangan *wireframe* dengan menggunakan tools Figma untuk setiap halaman utama dengan fokus pada user experience dan kemudahan navigasi:

a. Dashboard Page Wireframe:



GAMBAR 5 Wireframe Dashboard Page

memiliki Broadcast Control Manager sebagai *toggle status on-air/off-air* manual, *container* informasi Total Schedule (peminjaman), Signage Content (konten digital signage aktif), Studio Status (status database), dan *card* Quick Actions untuk akses cepat jadwal, pengaturan digital signage, sinkronisasi database-Google Sheets, serta *refresh* halaman.

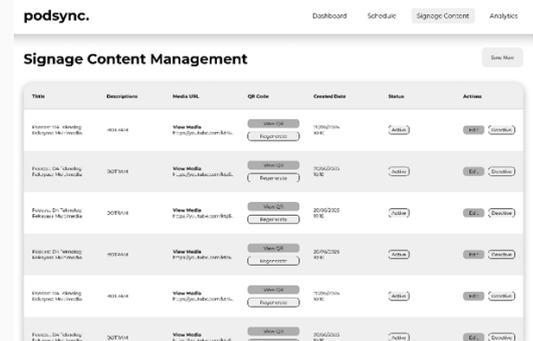
b. Schedule Wireframe:



GAMBAR 6 Wireframe Schedule Page

Dilengkapi tombol "Sync Now" untuk sinkronisasi Google Sheets ke *database*, search box, filter berdasarkan tanggal dan Unit/Prodi, serta dropdown items per-page. Tabel data berisi 8 kolom: timestamp, nama, NIP/NIM, Unit/Prodi, nomor telepon, tujuan penggunaan, fasilitas, dan tanggal.

c. Content Management Wireframe:



GAMBAR 7 Wireframe Content Management Page

Memiliki tombol "Add New Content" untuk menambah konten *digital signage* dan tabel dengan 7 kolom: title, description, media URL, QR Code (view/regenerate), create date, status (aktif/non-aktif), dan actions (edit/aktivasi konten).

d. Analytics Dashboard Wireframe:



GAMBAR 8 Wireframe Analytics Page

Menyediakan *dropdown filtering* perminggu, tombol refresh, *container* informasi (total content, today's podcast, status broadcast, lab bookings), berbagai grafik analisis (peminjaman, penggunaan fasilitas, berdasarkan jam dan Unit/Prodi), *container* System Performance (waktu pakai, memory, versi framework, OS platform), Database Status (kesehatan, kecepatan respon, jumlah data), dan riwayat aktivitas aplikasi.

2. Userflow Design.

Perancangan *userflow* untuk memahami perjalanan pengguna dalam menggunakan aplikasi web "Podsync".



GAMBAR 9 Desain Diagram Alur

G. Desain Integrasi Sistem.

1. Arsitektur Data Flow.

Perancangan alur data input hingga output dengan mempertimbangkan integrasi Google Sheets dan Unity:



GAMBAR 10 Diagram Alur Data

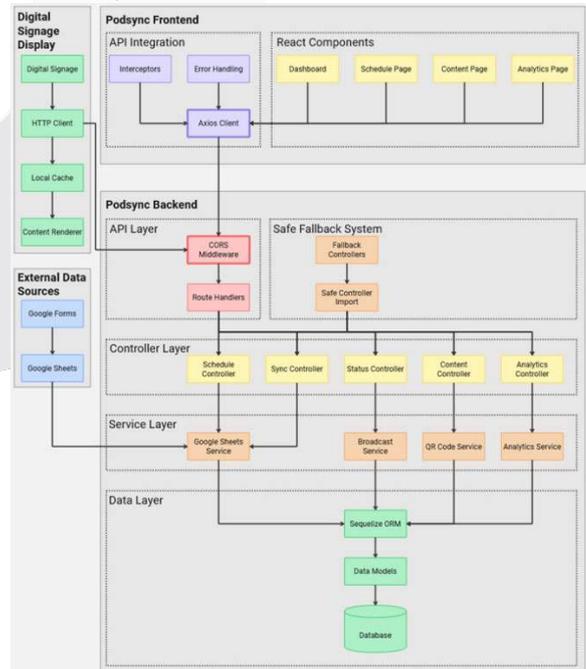
Proses dimulai dari tahap *input*, dimulai dari Google Forms yang diisi peminjam dan *input* manual admin. Data Google Forms otomatis masuk ke Google Sheets, kemudian diambil secara berkala (setiap 5 menit) oleh Sync Service yang meneruskannya ke Data Parser untuk memproses data mentah menjadi format yang sesuai, termasuk deteksi duplikasi, konflik, dan normalisasi nama unit/program studi.

Tahap *storage* menyimpan data hasil proses ke dalam tiga jenis penyimpanan: Raw Data, Processed Data, serta Content Data dan Status Data yang berisi informasi konten digital signage dan status on-air/off-air.

Terakhir, pada bagian *output*, menggunakan REST API sebagai penghubung *backend-frontend* yang menyuplai data *real-time* ke dua saluran utama: Web Dashboard dan Unity Display.

2. Arsitektur API.

Perancangan RESTful API menggunakan safe fallback system untuk memastikan reliabilitas sistem.



GAMBAR 11 Diagram Arsitektur API

Sumber data eksternal berasal dari Google Forms untuk peminjaman yang otomatis tersimpan di Google Sheets sebagai sumber sinkronisasi utama backend.

Disisi *frontend*, dibangun dengan React terdiri dari komponen Dashboard, Schedule, Signage Content, dan Analytics Dashboard. Komunikasi dengan backend menggunakan Axios Client dengan *interceptors* untuk otorisasi dan penanganan *error* terpusat.

Backend tersusun secara berlapis dimulai dari lapisan API yang memiliki CORS *middleware* untuk mengelola permintaan lintas domain serta sistem autentikasi yang menjamin akses aman ke setiap fitur. Semua permintaan pengguna diarahkan oleh *route handlers* ke *controller* yang sesuai. Pada lapisan *controller*, setiap jenis data dikelola oleh *controller* khusus seperti Schedule Controller, Sync Controller, Status Controller, dan Analytics Controller yang berinteraksi dengan service layer untuk menangani logika sinkronisasi Google Sheets, status siaran, pembuatan QR Code, dan analytics. Sistem dilengkapi Safe Fallback System yang mampu mengarahkan permintaan ke fallback controller jika layanan utama gagal, dan semua *service* menggunakan lapisan data yang terhubung ke MySQL Database melalui Sequelize ORM.

Unity Display sebagai *digital signage* melakukan permintaan HTTP ke *backend*, menyimpan data dalam *cache* lokal sebelum ditampilkan *content renderer*. Arsitektur ini mendukung fungsi *offline fallback* agar *signage* tetap aktif *real-time* meski koneksi terganggu.

H. Kebutuhan Perangkat Kerja.

Berikut paparan perangkat lunak dan perangkat keras yang akan digunakan selama pengembangan aplikasi "Podsync".

TABEL 2
Mapping Perangkat Lunak dan Fungsinya

Perangkat Lunak		
No.	Perangkat Lunak	Fungsi Utama
1	Node.js	Backend Runtime
2	Express.js	Routing API server
3	Sequelize ORM	Pengelolaan relasi database
4	MySQL	Penyimpanan data atau database
5	React.js	Frontend web app

6	Tailwind CSS	Styling frontend responsif
7	Google Sheets API	Integrasi data peminjaman
8	Axios	Komunikasi client-server
9	Postman	Pengujian API secara manual
10	VSCode	Text Editor
11	Github	Version control

TABEL 3
Mapping Perangkat Keras yang Digunakan Saat Pengembangan Aplikasi "Podsync"

Perangkat Keras			
No.	Perangkat Keras	Hardware	Spesifikasi
1	Laptop Pribadi (ASUS TUF FA506QM)	CPU	AMD RYZEN 7 5800H
2		GPU	NVIDIA RTX 3060 6GB VRAM
3		RAM	24 GB DDR 4 3200Mhz
4		SSD	NVMe 512 GB PCIe 4.0

I. Implementasi Sistem

Implementasi sistem "PodSync" mengintegrasikan berbagai fitur utama untuk pengelolaan Digital Signage Laboratorium Podcast secara modular untuk kemudahan pengembangan dan maintenance.

1. Fitur Sinkronisasi Google Sheets Otomatis.

Fitur ini memungkinkan sistem melakukan sinkronisasi berkala dengan Google Sheets yang berisi data peminjaman dari Google Forms untuk memastikan data selalu *up-to-date*.

```

1 // Google Sheets Service Implementation
2 class GoogleSheetsService {
3   constructor() {
4     this.auth = new google.auth.GoogleAuth({
5       credentials: {
6         client_email: process.env.GOOGLE_SHEETS_CLIENT_EMAIL,
7         private_key: process.env.GOOGLE_SHEETS_PRIVATE_KEY.replace(/\\n/g, '\n')
8       },
9       scopes: ['https://www.googleapis.com/auth/spreadsheets.readonly']
10    });
11    this.sheets = google.sheets({ version: 'v4', auth: this.auth });
12  }
13
14  async syncData() {
15    try {
16      const response = await this.sheets.spreadsheets.values.get({
17        spreadsheetId: process.env.GOOGLE_SHEETS_SPREADSHEET_ID,
18        range: 'Sheet1!A:H'
19      });
20
21      const rows = response.data.values;
22      return await this.processAndStoreData(rows);
23    } catch (error) {
24      console.error('Google Sheets sync error:', error);
25      throw error;
26    }
27  }
28 }
    
```

GAMBAR 12

Implementasi Teknis Fitur Sinkronisasi google Sheets

Sistem terdiri dari lima komponen: Auto-sync Scheduler menggunakan cron job setiap 5 menit untuk update otomatis, Data Parser mengolah format tanggal/waktu dari Google Forms ke format standar, Unit Mapper menstandarisasi nama unit/program studi, Conflict Resolution mendeteksi dan menangani data duplikat dengan unique constraint pada timestamp dan NIP, serta Error Handling menyediakan retry mechanism dan fallback untuk kegagalan koneksi.

Manfaat Fitur ini mengeliminasi manual data entry untuk mengurangi human error, memberikan real-time update jadwal peminjaman, memastikan konsistensi data antara Google Forms dan PodSync untuk mencegah konflik jadwal, serta menyediakan audit trail untuk transparansi tracking perubahan data.

2. Fitur Broadcast Management System.

Sistem manajemen broadcast dapat mengontrol status *on-air/off-air* secara otomatis berdasarkan jadwal aktif atau manual oleh administrator.

```

1 // Broadcast Scheduler Implementation
2 class BroadcastScheduler {
3   start() {
4     console.log('Starting broadcast scheduler');
5     this.interval = setInterval(() => {
6       this.checkScheduleStatus();
7     }, 60000); // Check setiap menit
8   }
9
10  async checkScheduleStatus() {
11    try {
12      const now = new Date();
13      const activeSchedules = await Schedule.findAll({
14        where: {
15          date: now.toISOString().split('T')[0],
16          start_time: { [Op.lte]: now.toTimeString().split(' ')[0] },
17          end_time: { [Op.gte]: now.toTimeString().split(' ')[0] }
18        }
19      });
20
21      const shouldBeOnAir = activeSchedules.length > 0;
22      await this.updateBroadcastStatus(shouldBeOnAir, 'system-auto');
23    } catch (error) {
24      console.error('Broadcast check error:', error);
25    }
26  }
27
28  async updateBroadcastStatus(isOnAir, updatedBy) {
29    await BroadcastStatus.create({
30      isOnAir,
31      statusMessage: isOnAir ? 'On Air - Auto Control' : 'Off Air - Auto Control',
32      updatedBy,
33      lastUpdated: new Date()
34    });
35  }
36 }
    
```

GAMBAR 13

Implementasi Teknis Fitur Broadcast Management System

Sistem menggunakan arsitektur dengan lima komponen utama: Auto Scheduler memonitor jadwal setiap menit dan mengubah status broadcast otomatis, Manual Override memberikan kontrol admin dengan prioritas tinggi, Status History merekam semua perubahan dengan timestamp untuk audit, Real-time Notification memastikan update langsung ke Unity display, dan API Endpoints menyediakan RESTful API dengan error handling yang robust.

Manfaat Fitur ini memberikan otomatisasi untuk mengurangi intervensi manual, fleksibilitas kontrol admin saat situasi darurat, audit trail untuk troubleshooting dan analisis pola penggunaan lab, serta integrasi real-time dengan Unity digital signage untuk akses informasi terkini.

3. Sistem Content Management.

Sistem pengelolaan konten *digital signage* memungkinkan admin untuk membuat, mengedit, mengatur, dan menjadwalkan konten yang akan ditampilkan.

```

1 // Content Controller Implementation
2 class ContentController {
3   async createContent(req, res) {
4     try {
5       const { title, description, content, type, mediaUrl, displayOrder } = req.body;
6
7       let qrCodeUrl = null;
8       if (mediaUrl) {
9         qrCodeUrl = await this.generateQRCode(mediaUrl);
10      }
11
12      const newContent = await SignageContent.create({
13        title,
14        description,
15        content,
16        type,
17        mediaUrl,
18        qrCodeUrl,
19        displayOrder: displayOrder || 0,
20        isActive: true,
21        createdBy: req.user?.username || 'admin'
22      });
23    }
24  }
25 }
    
```

```

23
24     res.status(201).json({
25         success: true,
26         message: 'Content created successfully',
27         data: newContent
28     });
29 } catch (error) {
30     res.status(500).json({ error: error.message });
31 }
32 }

```

GAMBAR 14

Implementasi Teknis Fitur Content Management System

Content Management System diimplementasikan dengan pendekatan modular melalui beberapa komponen: CRUD Operations menyediakan operasi Create, Read, Update, Delete yang lengkap untuk pengelolaan *content* dengan validasi *input robust* dan error handling, QR Code Generator terintegrasi otomatis dalam proses pembuatan *content* yang secara *automatic generate QR code* berkualitas tinggi dan *scannable* setiap kali user meng-*input* media URL, Media Upload Handler mendukung *upload* berbagai format *file* gambar dan video dengan validasi *size* dan format untuk memastikan kompatibilitas dengan *Unity display*, serta Content Activation menyediakan *toggle mechanism* untuk mengaktifkan dan menonaktifkan *content* secara *real-time* tanpa *restart* sistem.

Manfaat sistem memberikan kemudahan signifikan bagi *administrator* melalui *user-friendly interface* yang memungkinkan mengelola *content digital signage* dengan mudah, serta *automated QR code generator* yang mempermudah *sharing media content* kepada *users* tanpa *tools* eksternal atau proses manual yang kompleks.

4. Fitur Integrasi Unity Digital Signage.

Integrasi dengan *Unity engine* untuk menampilkan informasi *real-time* di *digital signage* dengan *rendering* optimal dan *offline support*.

```

1 // Unity Integration Implementation
2 public class IntegratedBookingDisplayManager : MonoBehaviour
3 {
4     [Header("API Configuration")]
5     public string baseUrl = "http://localhost:8002/api";
6     public float updateInterval = 30f;
7
8     [Header("Display Components")]
9     public Text scheduleDisplay;
10    public Text broadcastStatusDisplay;
11    public RawImage qrCodeDisplay;
12
13    private void Start()
14    {
15        StartCoroutine(DataUpdateRoutine());
16    }
17
18    private IEnumerator DataUpdateRoutine()
19    {
20        while (true)
21        {
22            yield return StartCoroutine(FetchScheduleData());
23            yield return StartCoroutine(FetchBroadcastStatus());
24            yield return StartCoroutine(FetchSignageContent());
25            yield return new WaitForSeconds(updateInterval);
26        }
27    }

```

```

28
29    private IEnumerator FetchScheduleData()
30    {
31        using (UnityWebRequest www = UnityWebRequest.Get($"{baseUrl}/schedule/today"))
32        {
33            yield return www.SendWebRequest();
34
35            if (www.result == UnityWebRequest.Result.Success)
36            {
37                var scheduleData = JsonUtility.FromJson<ScheduleResponse>(www.downloadHandler.text);
38                UpdateScheduleDisplay(scheduleData);
39            }
40            else
41            {
42                Debug.LogWarning("Failed to fetch schedule data, using cached data");
43                LoadCachedScheduleData();
44            }
45        }
46    }
47 }

```

GAMBAR 15

Implementasi Teknis Fitur Integrasi Unity Digital Signage Integration

Unity Digital Signage Integration diimplementasikan dengan arsitektur *robust* dan *performance-oriented* melalui beberapa komponen: HTTP Client menggunakan *Unity UnityWebRequest* untuk komunikasi API dengan *PodSync backend*, dengan *error handling* dan *timeout configuration*, Real-time Updates diimplementasikan melalui *polling mechanism* setiap 30 detik untuk memastikan data yang ditampilkan selalu *up-to-date* tanpa memberatkan *network bandwidth*, Offline Support menyediakan *local caching system* yang menyimpan data terakhir yang berhasil di-*fetch* sehingga *digital signage* tetap dapat menampilkan informasi meskipun koneksi internet terputus sementara, Content Renderer melakukan *dynamic rendering* untuk berbagai jenis *content* termasuk schedule information, broadcast status, dan QR codes dengan *layout* yang *responsive* dan *visually appealing*, serta Performance Optimization memastikan *memory management*

5. Fitur Analytics Dashboard.

Dashboard analytics yang menyediakan *insights* tentang penggunaan laboratorium, statistik *booking*, dan *performance metrics*.

```

1 // Analytics Controller Implementation
2 class AnalyticsController {
3     async getDashboardStats(req, res) {
4         try {
5             const timeRange = req.query.timeRange || '7d';
6             const endDate = new Date();
7             const startDate = this.calculateStartDate(endDate, timeRange);
8
9             const stats = await Promise.all([
10                this.getContentStats(startDate, endDate),
11                this.getScheduleStats(startDate, endDate),
12                this.getBroadcastStats(startDate, endDate),
13                this.getUsageStats(startDate, endDate)
14            ]);
15
16             res.json({
17                 timeRange,
18                 content: stats[0],
19                 schedule: stats[1],
20                 broadcast: stats[2],
21                 usage: stats[3],
22                 timestamp: new Date()
23             });
24         } catch (error) {
25             res.status(500).json({ error: error.message });
26         }
27     }
28 }

```

```

28
29 async getUsageStats(startDate, endDate) {
30   const totalBookings = await Schedule.count({
31     where: {
32       createdAt: { [Op.between]: [startDate, endDate] }
33     }
34   });
35
36   const unitStats = await Schedule.findAll({
37     attributes: [
38       'organizer',
39       [Sequelize.fn('COUNT', Sequelize.col('id')), 'bookingCount']
40     ],
41     where: {
42       createdAt: { [Op.between]: [startDate, endDate] }
43     },
44     group: ['organizer'],
45     order: [[Sequelize.literal('bookingCount'), 'DESC']]
46   });
47
48   return { totalBookings, unitStats };
49 }
50 }

```

GAMBAR 16

Implementasi Teknis Fitur Analytics Dashboard

Analytics Dashboard dirancang untuk memberikan gambaran menyeluruh secara *real-time* tentang penggunaan laboratorium melalui grafik unit/prodi, frekuensi peminjaman, dan pemanfaatan sistem yang memantau kinerja harian secara langsung. Analisis penggunaan per-unit atau program studi menampilkan pola pemakaian yang jelas, sedangkan fitur grafik peminjaman mengungkap waktu paling ramai dipinjam. Visualisasi data ditampilkan melalui berbagai grafik, dan sistem juga memantau performa teknis seperti kecepatan respon dan penggunaan memori untuk memastikan layanan tetap berjalan dengan baik.

J. Pengujian

Pengujian sistem aplikasi “Podsync” dilakukan menggunakan metodologi *blackbox testing* dengan pendekatan “User Acceptance Testing” untuk memvalidasi bahwa sistem memenuhi kebutuhan pengguna dan berfungsi sesuai spesifikasi yang telah ditetapkan.

IV. HASIL DAN PEMBAHASAN

Pengembangan aplikasi web "Podsync" untuk aplikasi "Digital Signage Laboratorium Podcast" telah berhasil diselesaikan dengan baik dan menghasilkan sistem media penyampaian informasi. Kedua sistem yang dikembangkan mampu menampilkan informasi secara *real-time* dengan selaras, *interface user-friendly*, dan performa yang stabil.

A. Hasil Akhir (Luaran).

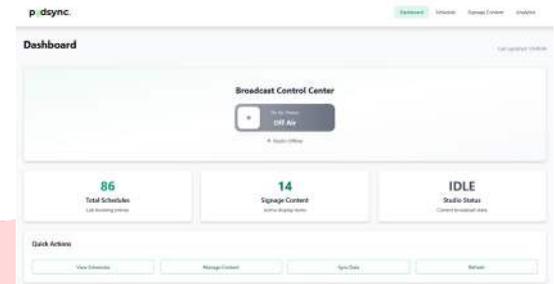
1. Tampilan Antarmuka.

Berikut akan dipaparkan beberapa hasil tampilan antarmuka dari aplikasi web “Podsync”:

a. Dashboard.

Halaman dashboard merupakan pusat kontrol utama sistem PodSync yang menampilkan informasi *real-time* tentang status laboratorium podcast. Dashboard dirancang dengan *layout* yang *clean* dan informatif, menampilkan status

broadcast saat ini dengan indikator visual yang jelas antara kondisi "On Air" dan "Off Air". Bagian atas dashboard menampilkan control panel untuk mengubah status broadcast secara manual, dilengkapi dengan konfirmasi *action* untuk mencegah perubahan yang tidak disengaja.



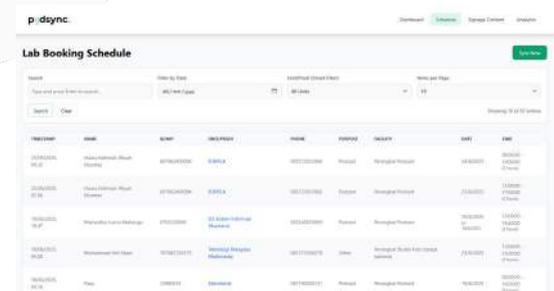
GAMBAR 17

Tampilan Antarmuka Halaman Dashboard

Bagian tengah Dashboard terdapat ringkasan jadwal hari ini yang menampilkan daftar peminjaman dengan informasi lengkap seperti nama peminjam, unit/program studi, waktu mulai dan selesai, serta keperluan peminjaman. Dashboard juga dilengkapi widget statistik yang menampilkan data penting seperti total *booking* hari ini, jam operasional laboratorium, dan status sinkronisasi terakhir dengan Google Sheets. Desain responsif memastikan dashboard dapat diakses dengan optimal dari berbagai *device*.

b. Schedule.

Halaman schedule management menyediakan tampilan untuk pengelolaan jadwal laboratorium dengan berbagai fitur *filtering* dan *searching*. *Interface* utama menampilkan tabel jadwal yang dapat difilter berdasarkan tanggal, unit/program studi, dan status *booking*. Setiap *entry* jadwal menampilkan detail lengkap termasuk informasi peminjam, waktu *booking*, keperluan penggunaan, dan status *approval*.



GAMBAR 18

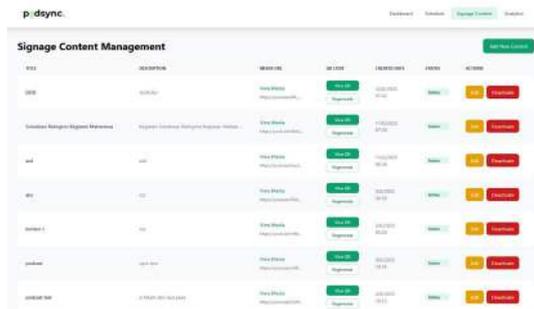
Tampilan Antarmuka Halaman Schedule

Fitur pencarian memungkinkan staff lab untuk mencari peminjaman lebih spesifik dengan *search* menggunakan nama peminjam atau unit.

Bagian atas halaman menampilkan status sinkronisasi dengan Google Sheets beserta *timestamp* terakhir *sync* dan tombol untuk melakukan manual *sync* jika diperlukan.

c. Signage Content.

Halaman content management dirancang untuk memudahkan *administrator* mengelola konten *digital signage* dengan *interface* yang menampilkan *grid view* semua konten tersedia. Setiap *card* konten menampilkan informasi essential seperti *title*, *type* konten, status *active/inactive*, dan tanggal pembuatan.



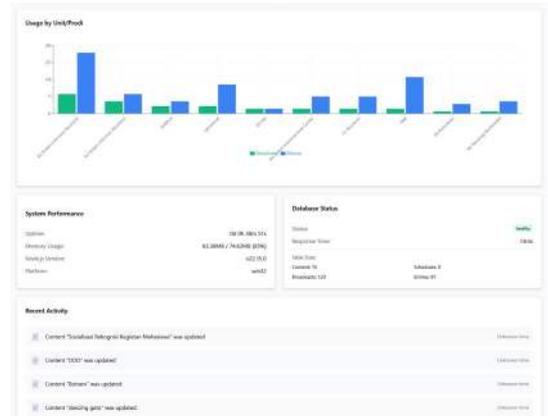
GAMBAR 19

Tampilan Antarmuka Halaman Signage Content

Form konten baru tersedia dengan *fields* termasuk *title*, *description*, *content body*, *media upload*, dan pengaturan *scheduling*. *Auto-generated QR code* ditampilkan *real-time* ketika user menginput media URL dengan *preview* yang dapat di-*scan* langsung. Display order management memungkinkan administrator mengatur urutan tampil konten. Bulk actions tersedia untuk mengaktifkan atau menonaktifkan *multiple content* sekaligus, meningkatkan *efficiency* dalam *content management*.

d. Analytics

Dashboard analytics merupakan fitur yang menyajikan informasi penggunaan laboratorium melalui visualisasi data yang interaktif.



GAMBAR 20

Tampilan Antarmuka Halaman Analytics Aplikasi

Fitur ini menampilkan *chart* utama untuk melihat *trend booking* berdasarkan waktu, *breakdown statistics* yang menunjukkan distribusi penggunaan per-unit atau program studi beserta persentase dan jumlah *booking*, serta *peak hours analysis* dalam format bar *chart* untuk mengidentifikasi jam-jam sibuk laboratorium. Selain itu, *dashboard* juga menyediakan *Key Performance Indicators (KPI)* dalam bentuk *card metrics* yang mencakup total *bookings*, *average session duration*, *utilization rate*, dan *popular time slots* untuk mendukung analisis dan optimasi penggunaan laboratorium.

2. Hasil Pengujian User Acceptance Testing Dengan Metode Blackbox.

Pengujian User Acceptance Testing terhadap aplikasi web "Podsync" telah dilakukan pada 23 Juni 2025 oleh penanggung jawab Laboratorium Podcast Fakultas Ilmu Terapan menggunakan metode *Blackbox testing*. Dari 32 poin pengujian yang dilakukan, seluruh fitur aplikasi berfungsi dengan baik dan memenuhi kriteria "Berhasil" tanpa ada kegagalan fungsi. Pengujian yang dipimpin oleh Staff Laboran Muhammad Harun Arrasyid D. mencakup fungsi-fungsi utama seperti sinkronisasi data dari Google Sheets, pengelolaan jadwal, kontrol status *broadcast*, dan manajemen konten *digital signage*. Hasil evaluasi menunjukkan aplikasi sangat memudahkan operasional harian dengan skor 5/5, kemudahan pemantauan status laboratorium mendapat nilai 4/5, efisiensi kontrol *broadcast* dinilai sangat baik dengan skor 5/5, dan manajemen konten *digital signage* memperoleh skor 4/5. Secara keseluruhan, aplikasi "Podsync" dinyatakan layak untuk digunakan dalam operasional laboratorium dan telah memenuhi ekspektasi fungsional pengguna.

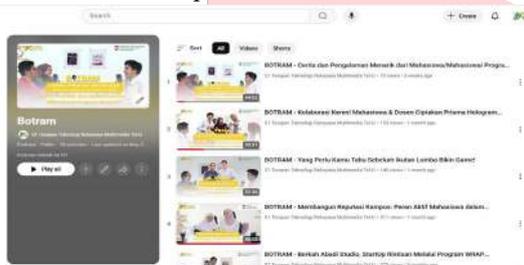
B. Hasil Akhir Magang.

1. Luaran Operasional dan Pembuatan Konten.

- a. Produksi Konten Podcast BOTRAM (Obrolan Teknologi Rekayasa Multimedia).

Selama periode magang, produksi video lebih dari

Produksi 18 episode podcast BOTRAM telah menghasilkan lebih dari 1.125 menit konten audio-visual berkualitas melalui proses kerja yang sistematis meliputi perencanaan pra-produksi, pengaturan perekaman kamera, *editing* pasca-produksi, dan optimasi untuk platform digital. Konten ini mencapai performa yang memuaskan dengan 11.9K penayangan, 1.1K jam waktu tonton, dan penambahan 78 *subscriber*, yang mendemonstrasikan tingginya *engagement* dan kualitas konten di platform YouTube TRM.



GAMBAR 21
Konten Youtube BOTRAM

- b. Manajemen Konten di Multi Platform

Penerapan sistem Content Planner melalui *workspace* Notion telah mendukung koordinasi tim yang efektif dengan menggunakan kategorisasi warna yang terstruktur, yaitu Pink untuk ucapan selamat, Biru untuk *tips and trick*, Hijau untuk informasi multimedia, Kuning untuk informasi prodi/fakultas, Merah untuk konten BOTRAM, dan Orange untuk *meme* TRM. Sistem ini berhasil menciptakan konsistensi konten yang dapat diukur selama periode magang berlangsung.

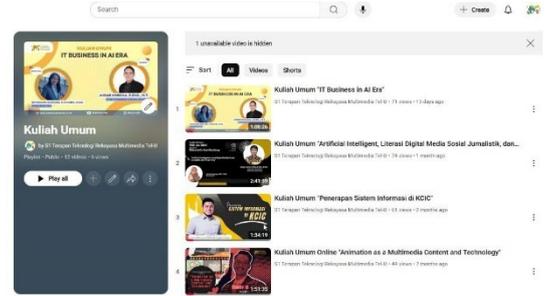


GAMBAR 22
Perencanaan Jadwal Konten di Notion

- c. Live Streaming Operator dan Konten Event

Selama kegiatan magang, telah dilakukan operasi *live streaming* untuk lebih dari 10 kuliah umum dan kuliah besar dengan pengaturan teknis di laboratorium podcast dan aula. Pembuatan konten *event* EXPO dan acara akademik menghasilkan dokumentasi yang mendukung

promosi program studi dan meningkatkan popularitas institusi.



GAMBAR 23
Konten Youtube Event

Hasil produksi selama program magang 2 semester di KK DBES T sebagai Konten Kreator S1 Terapan Teknologi Rekayasa Multimedia mencakup 4 kategori konten utama. Video podcast BOTRAM diproduksi sebanyak 18 episode dengan durasi rata-rata 45 menit yang didistribusikan melalui YouTube @d4trm-telu, konten *feed* Instagram sebanyak 30 post dengan konsistensi visual *branding* di Instagram @s1trm.fit, artikel *website* sebanyak 16 artikel di platform dsm.telkomuniversity.ac.id, serta dokumentasi *event* dan *live streaming* sebanyak 20 acara yang didistribusikan melalui *multi platform*. Semua konten dokumentasi event dan live streaming ini selanjutnya akan dipublikasikan melalui platform YouTube, Instagram, dan artikel untuk mendukung promosi program studi.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

1. Sistem *backend* "Podsync" berhasil memenuhi kebutuhan pengelolaan jadwal laboratorium podcast dan distribusi konten *digital signage* dengan fitur utama yang terintegrasi dan berfungsi sesuai tujuan pengembangan
2. Integrasi *real-time* Google Sheets API mempermudah proses *input* dan sinkronisasi data peminjaman laboratorium, meningkatkan efisiensi dan akurasi pengelolaan data
3. Sistem broadcast management memungkinkan *monitoring* status siaran langsung secara *real-time*, memastikan informasi *digital signage* selalu mutakhir dan dapat diandalkan.
4. Dashboard analytics memberikan kemudahan analisis pola penggunaan laboratorium dan performa sistem untuk pengambilan keputusan berbasis data.

B. Saran

1. Peningkatan Keamanan Data: Menambahkan fitur autentikasi dua faktor, enkripsi data, dan audit log untuk memastikan integritas dan keamanan data

2. Peningkatan Hosting Aplikasi: Menggunakan layanan cloud hosting profesional untuk meningkatkan uptime, keamanan, dan kemudahan pemeliharaan sistem
3. Performance Optimization: Melakukan optimasi performa melalui perbaikan query database, penerapan caching, dan minimalisasi beban server
4. Peningkatan Infrastruktur: Mengimplementasikan *load balancer*, *backup data* berkala, dan *monitoring* sistem *real-time* untuk meningkatkan stabilitas dan kesiapan menghadapi lonjakan trafik.

REFERENSI

- [1] S. Sugiono, "Industri Konten Digital dalam Perspektif Society 5.0," *Jurnal IPTEK-KOM (Jurnal Ilmu Pengetahuan dan Teknologi Komunikasi)*, vol. Vol. 22, no. No. 2, pp. 175 - 191, 2020.
- [2] G. Tiara, "DRAMATURGI PENGELOLAAN KESAN KONTEN KREATOR DI MEDIA," *Jurnal AKRAB JUARA*, vol. 6, no. 3, pp. 14-24, 2021.
- [3] W. Ahmad Jaafar Wan Yahaya, R. R. dan S. S., "Multimedia-based Information System for Technology," *Profesional de la informaci3n*, vol. 33, no. 1, 2024.
- [4] A. Wibisono, Sudargo, I. Menarianti dan Wijayanto, "Development of Integrated Laboratory," *SISFORMA: Journal of Information Systems (e-Journal)*, vol. 9, no. 2, pp. 4359-4370, 2022.
- [5] W. A. J. Wan Yahya, S. Srihadi dan R. Restu, "Multimedia-based Information System for Technology and Vocational Education Laboratory," *Profesional de la informaci3n*, vol. 33, no. 1, pp. 1-10, 2024.
- [6] A. Z. Hilmi dan A. Fauzi, "Pembuatan Media Promosi Berbasis Website Menggunakan CMS WordPress untuk Meningkatkan Minat Beli Pada CV. Hutama Karya Construction Malang," *Jurnal Aplikasi Bisnis (JAB)*, vol. 10, no. 1, pp. 1-7, 2024.
- [7] I. S. Musyaffa dan A. Rahmah, "ANALISIS DAN PENGEMBANGAN SISTEM PENGELOLAAN DIGITAL SIGNAGE," *Jurnal Informatika Terpadu*, vol. 7, no. 2, pp. 75-79, 2021.
- [8] I. N. Z. Aliya, M. A. Priyanto, R. Fiddiyansyah d DATABASE TERSENTRALISASI TERINTEG RECOVERY PADA UMKM," Prosiding Seminar 423-432, 2023.
- [9] R. N. Alghifari, D. S. Kusumo dan N. Selviandr Pengembangan Backend Aplikasi Cafeasy Terintegrasi Google Sheets (Studi Kasus: Cafe D Proceeding of Engineering, vol. 11, no. 4, pp. 5189
- [10] Y. Adisa dan M. I. Padli Nasution, "Konsep Manajemen Basis Data Relasional Pada," *MASIP Administrasi Bisnis dan Publik Terapan*, vol. 1, no.
- [11] A. R. Hakim, "BASIS DATA TERDISTRIBU MANAJEMEN, DAN TANTANGAN IMPLEME vol. 1, no. 2, 2024.
- [12] I. P. G. Widiastawa Adiputra, I. N. Purnama dan "RANCANG BANGUN SISTEM INFORM PELANGGAN POTENSIAL BERBASIS WEB (INDONESIA COMNETS PLUS)," *Jurnal Tekn Komputer*, vol. 9, no. 1, pp. 10-22, 2023.
- [13] G. B. Timur dan E. Utami, "Implementasi Frame Methodology dalam Management Proyek, Studi l Konsultan IT," *Technologia : Jurnal Ilmiah*, vol. 2024.
- [14] S. Pratama, S. Ibrahim dan M. A. Reybaharsyah, " Scrum Dalam Membentuk Sistem Informasi Pe Berbasis Web," *INFORMATIKA DAN TEKNOLO 3*, no. 1, pp. 27-35, 2022.
- [15] Julia, M. I. Mutahari, Renaldi dan Saepullah, "A BASIS DATA TERDISTRIBUSI DALAM LIN COMPUTING," *Karimah Tauhid*, vol. 3, no. 2, pp.
- [16] R. P. Aji, A. D. Pritama, B. D. Putranto dan Darso METODE AGILE SCRUM PADA PERANCA PENGECEKAN PENYAKIT DIABETES ASSESMENT," *Jurnal Rekayasa Informasi*, vol. 2024.
- [17] S. Hadji, M. Taufik dan S. Mulyono, "IMPLEM SCRUM PADA PENGEMBANGAN APLIKASI I BERBASIS WEBSITE (STUDI KASUS PADA LOMBOK IDJO SEMARANG)," *KONFI MAHASISWA UNISSULA (KIMU) 2*, pp. 32-43,
- [18] "UJI FUNGSIONALITAS (BLACKBOX TI INFORMASI LEMBAGA SERTIFIKASI PROFI DENGAN APPERFECT WEB TEST DAN JOINED (Journal of Informatics Education), vol. 1 2018.

- [19] T. Natarajan dan S. Pichai, "Transition From Waterfall to Agile Methodology: An Action Research Study," IEEE Access, vol. 12, pp. 49341-49362, 2024.
- [20] I. Afrianto, A. Heryandi, A. Finadhita dan S. Atin, "User Acceptance Test For Digital Signature Application In Academic Domain To Support The Covid-19 Work From Home Program," International Journal of Information System & Technology, vol. 5, no. 3, pp. 270-280, 2021.
- [21] T. Natarajan dan S. Pichai, "Transition From Waterfall to Agile Methodology: An Action Research Study," IEEE Access, vol. 12, pp. 49341-49362, 2024.
- [22] A. Z. Hilmi dan A. Fauzi, "Pembuatan Media Promosi Berbasis Website Menggunakan CMS WordPress untuk Meningkatkan Minat Beli Pada CV. Utama Karya Construction Malang," Jurnal Aplikasi Bisnis (JAB), vol. 10, no. 1, pp. 1-7, 2024.

