PENGEMBANGAN BASIS DATA WEBSITE RADYAKARTIYASA MENGGUNAKAN *HEADLESS CMS DIRECTUS*

Disusun Oleh:

Nama : Mutiara Irdina

NIM : 21523034

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2025

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PENGEMBANGAN BASIS DATA WEBSITE RADYAKARTIYASA MENGGUNAKAN *HEADLESS CMS DIRECTUS*

TUGAS AKHIR JALUR MAGANG

Disusun Oleh:

Nama	: Mutiara Irdina
NIM	: 21523034

Yogyakarta, 19 Oktober 2025

Pembimbing,



Dr. Ahmad Luthfi S.Kom., M.Kom.

HALAMAN PENGESAHAN DOSEN PENGUJI

PENGEMBANGAN BASIS DATA WEBSITE RADYAKARTIYASA MENGGUNAKAN *HEADLESS CMS DIRECTUS*

TUGAS AKHIR JALUR MAGANG

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta,

Tim Penguji

Ketua Penguji

Ahmad Luthfi, Dr., S.Kom., M.Kom.

Anggota 1

Aridhanyati Arifin, S.T., M.Cs.

Anggota 2

Moh. Idris, S.Kom., M.Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama	:	Mutiara Irdina
NIM	:	21523034

Tugas akhir dengan judul:

PENGEMBANGAN BASIS DATA WEBSITE RADYAKARTIYASA MENGGUNAKAN *HEADLESS CMS DIRECTUS*

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 6 September 2025



(Mutiara Irdina)

HALAMAN PERSEMBAHAN

Segala puji dan syukur saya panjatkan kepada Allah SWT. atas limpahan nikmat dan karunia-Nya sehingga saya menyelesaikan tugas akhir ini. Karena dengan izin Allah saya mampu melewati setiap proses selama penyusunan laporan ini. Semoga segala usaha ini dapat menjadi amal yang diterima dan bisa bermanfaat bagi saya dan sekitar.

Tugas akhir ini saya persembahkan kepada keluarga tersayang Bapak, Ibu, serta Bani dan Adek, yang senantiasa memberikan dukungan tanpa henti dalam proses perkuliahan hingga selesai. Terima kasih atas setiap doa, nasihat, bimbingan, arahan, dan kepercayaan yang telah diberikan kepada saya. Tanpa adanya keluarga, saya tidak akan menjadi seperti saya yang sekarang.

HALAMAN MOTO

"Sesungguhnya bersama kesulitan ada kemudahan."
(QS. Al-Insyirah: 6)

KATA PENGANTAR

Assalamu'alaikum Wr.Wb.

Alhamdulillahirabbil 'alamin puji syukur penulis panjatkan kepada Allah SWT, karena telah memberikan rahmat, karunia, dan hidayah-Nya, sehingga penulis dapat menyelesaikan laporan tugas akhir yang berjudul "Pengembangan Basis Data *Website* Radyakartiyasa Menggunakan *Headless CMS Directus*" dengan baik dan lancar. Laporan tugas akhir ini disusun sebagai syarat untuk menyelesaikan pendidikan sarjana penulis pada Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Laporan tugas akhir ini tidak dapat terlaksana tanpa adanya dukungan, arahan, serta bantuan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa terimakasih yang sebesar-besarnya kepada:

1. Allah SWT yang telah memberikan kesehatan, kekuatan, dan kelancaran untuk menyelesaikan tugas akhir ini mulai dari pelaksanaan magang sampai dengan selesainya penulisan laporan akhir ini.
2. Kedua orang tua penulis, Bapak Tri Laksitohadi dan Ibu Titie Wahyuni, yang senantiasa memberikan dukungan baik itu doa, moral, maupun materi yang tidak ternilai agar penulis dapat melakukan perkuliahan dengan lancar sampai akhir.
3. Saudari kandung penulis, Hany Esti Palupi, dan Najla Sekar Amirah yang telah memberikan dukungan serta arahan kepada penulis agar dapat menjalankan perkuliahan dengan baik.
4. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Jurusan Informatika Universitas Islam Indonesia.
5. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D., selaku Ketua Program Studi Informatika Program Sarjana.
6. Bapak Andhika Giri Persada, S.Kom., M.Eng., selaku dosen pembimbing akademik yang telah memberikan bimbingan serta arahan dari awal perkuliahan. Ibu Erika Ramadhani, S.T., M.Eng., selaku dosen pembimbing akademik yang telah memberikan arahan hingga akhir perkuliahan di Universitas Islam Indonesia.
7. Bapak Dr. Ahmad Luthfi S.Kom., M.Kom., selaku Dosen Pembimbing tugas akhir yang telah membantu memberikan bimbingan, arahan, serta dukungan dari awal memulai pengerjaan tugas akhir sampai dengan selesainya pengerjaan tugas akhir.
8. Kepada seluruh petinggi dan karyawan PT Lumintu Logic yang telah memberikan kesempatan dan ilmu baru selama melakukan kegiatan magang.

9. Adristi Tazkia Putri, selaku teman dekat penulis yang telah menemani dan memberikan dukungan dari masa awal sekolah hingga sekarang.
10. Sasa, Emil, Cikuy, Tegar, dan Alwi selaku teman selama masa perkuliahan penulis yang telah membantu dan memberikan dukungan tanpa henti untuk penulis dalam menyelesaikan perkuliahan dan penulisan tugas akhir.
11. Teman-teman Informatika angkatan 2021 selaku teman seperjuangan penulis selama perkuliahan di Universitas Islam Indonesia.

Yogyakarta, 6 September 2025



(Mutiara Irdina)

SARI

Perkembangan teknologi digital memberikan peluang besar dalam pelestarian dan promosi budaya melalui media *website*. Kawedanan Radyakartiyasa bekerja sama dengan PT Lumintu Logic mengembangkan *website* sebagai sarana promosi destinasi dan *event* budaya Keraton Yogyakarta. Penulis berperan sebagai *backend developer* dalam program magang dengan fokus pada perancangan basis data menggunakan *Headless Content Management System* (CMS) Directus. Metodologi yang digunakan adalah Agile dengan kerangka kerja Scrum yang menekankan iterasi, kolaborasi tim, serta kemampuan adaptasi terhadap perubahan kebutuhan. Tahapan pengembangan mencakup konfigurasi Directus, pembuatan koleksi basis data sesuai kebutuhan konten *website (navigation, destination index, event index, about us, contact info*, hero, dan FAQ), serta pengujian API menggunakan Postman. Hasil pelaksanaan magang menunjukkan bahwa penggunaan Directus mampu mempercepat proses pengembangan basis data sekaligus menghasilkan sistem yang fleksibel dan mudah dikelola. Pengalaman ini juga memberikan pembelajaran praktis terkait penerapan teknologi industri yang belum diperoleh dalam perkuliahan, sehingga menjadi jembatan antara teori akademik dan praktik kerja.

Kata kunci: basis data, website, *Headless CMS*, Directus, Agile Scrum.

GLOSARIUM

Agile	Metode pengembangan perangkat lunak.
Application Programming Interface (API)	Sebuah antarmuka untuk menghubungkan dan mengintegrasikan aplikasi.
Cloudflare	Layanan untuk meningkatkan keamanan dan kinerja *website*.
Directus	*Tools* untuk mengelola basis data.
Docker	*Platform* untuk menjalankan aplikasi dalam container.
Headless CMS	Sistem pengelolaan konten yang memisahkan *backend* dengan tampilan *website*.
Lakon Lumintu	Aplikasi internal yang digunakan PT Lumintu Logic untuk berkomunikasi dan manajemen tugas.
Postman	Aplikasi untuk menguji API.
REST API	Cara untuk menghubungkan dan bertukar data antar sistem melalui HTTP.
Scrum	Suatu kerangka kerja dalam Agile untuk mengatur pengembangan perangkat lunak.
URL	Alamat unik untuk mengakses halaman atau sumber daya di internet.

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI ix
GLOSARIUM x
DAFTAR ISI xi
DAFTAR TABEL xiii
DAFTAR GAMBAR xiv
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Ruang Lingkup Magang 2
1.3 Tujuan 4
1.4 Manfaat 4
1.5 Sistematika Penulisan 5
BAB II DASAR TEORI 6
2.1 Website 6
2.2 Headless CMS 6
2.3 Directus 8
2.4 *Agile* 9
2.5 *Scrum* 10
2.6 Tinjauan Pustaka 13
BAB III PELAKSANAAN MAGANG 19
3.1 Welcoming 21
3.2 Training 21
3.3 Exit Interview 22
3.4 Pengerjaan Proyek Magang 23
3.4.1 Manajemen Proyek 23
3.4.2 Hasil Pengembangan Proyek 31

3.5 Evaluasi dan Penutupan Proyek 54
BAB IV REFLEKSI PELAKSANAAN MAGANG 56
4.1 Relevansi Akademik 56
4.2 Pembelajaran Magang 57
4.2.1 Manfaat 57
4.2.2 Kendala, Hambatan, dan Tantangan 57
BAB V PENUTUP 59
5.1 Kesimpulan 59
5.2 Saran 60
DAFTAR PUSTAKA 61
LAMPIRAN 64

DAFTAR TABEL

Tabel 2.1 Penelitian Sejenis 13
Tabel 3. 1 Daftar *Sprint 1*..... 36
Tabel 3. 2 Daftar *Sprint 2*..... 42
Tabel 3. 3 Daftar *Sprint 3*..... 47

DAFTAR GAMBAR

Gambar 1. 1 Struktur Organisasi Perusahaan 2
Gambar 2. 1 Cara Kerja *Headless CMS* 7
Gambar 2. 2 Arsitektur Directus 9
Gambar 3.1 Flow Tahapan Magang 19
Gambar 3. 2 *Timeline Project* 20
Gambar 3. 3 *Code* dari *File* Docker-Compose.yaml 26
Gambar 3. 4 Tampilan Lakon 27
Gambar 3.5 Tampilan Directus 28
Gambar 3.6 Tampilan Postman 29
Gambar 3.7 Tampilan Pengujian Metode GET 29
Gambar 3.8 *Raw Body Navigations* 30
Gambar 3.9 Tampilan Pengujian Metode Post 30
Gambar 3.10 *Raw Body* Untuk Metode POST 31
Gambar 3.11 Kode Konfigurasi Directus dalam Proyek 33
Gambar 3. 12 Struktur Koleksi Directus 35
Gambar 3.13 Tampilan Koleksi Navigation 37
Gambar 3. 14 Pengujian API Koleksi *Navigation (*GET*)* 37
Gambar 3. 15 Pengujian API Koleksi *Navigation* (POST) 38
Gambar 3.16 Tampilan Koleksi *Destination Index* 39
Gambar 3. 17 Pengujian API Koleksi *Destination Index* 40
Gambar 3.18 Tampilan Koleksi *Event Index* 41
Gambar 3. 19 Pengujian API Koleksi *Event Index* 41
Gambar 3.20 Tampilan Koleksi *About Us* 42
Gambar 3. 21 Pengujian API Koleksi *About Us* (GET) 43
Gambar 3. 22 Pengujian API Koleksi *About Us* (PATCH) 43
Gambar 3.23 Tampilan Koleksi *Contact Info* 44
Gambar 3. 24 Pengujian API Koleksi *Contact Info (*GET*)* 45
Gambar 3. 25 Pengujian API Koleksi *Contact Info* (PATCH) 45
Gambar 3.26 Tampilan Koleksi *Home Gallery* 46
Gambar 3. 27 Pengujian API Koleksi *Home Gallery* 47

Gambar 3.28 Tampilan Koleksi *Hero Banner* 48
Gambar 3. 29 Pengujian API Koleksi *Hero Banner* 49
Gambar 3.30 Tampilan Koleksi *Hero Backdrop* 49
Gambar 3.31 Tampilan Koleksi *Hero Card* 50
Gambar 3.32 Tampilan Koleksi *FAQ* 51
Gambar 3. 33 Pengujian API Koleksi *FAQ* 51
Gambar 3.34 Tampilan Koleksi *QnA* 52
Gambar 3. 35 Pengujian API Koleksi *QnA* (POST) 53
Gambar 3.36 Tampilan *Access Policies* 54

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi digital telah membuka peluang baru dalam upaya pelestarian dan pengenalan budaya kepada masyarakat luas. Salah satu media yang efektif untuk mendukung tujuan tersebut adalah *website*, yang dapat diakses dengan mudah oleh berbagai kalangan. *Website* dapat dimanfaatkan sebagai sarana edukasi dan promosi budaya (Surentu et al., 2020). Dimana perkembangan ini juga dimanfaatkan oleh keraton Yogyakarta sebagai salah satu upaya untuk mempromosikan berbagai destinasi di wilayah keraton dan juga *event* kebudayaan.

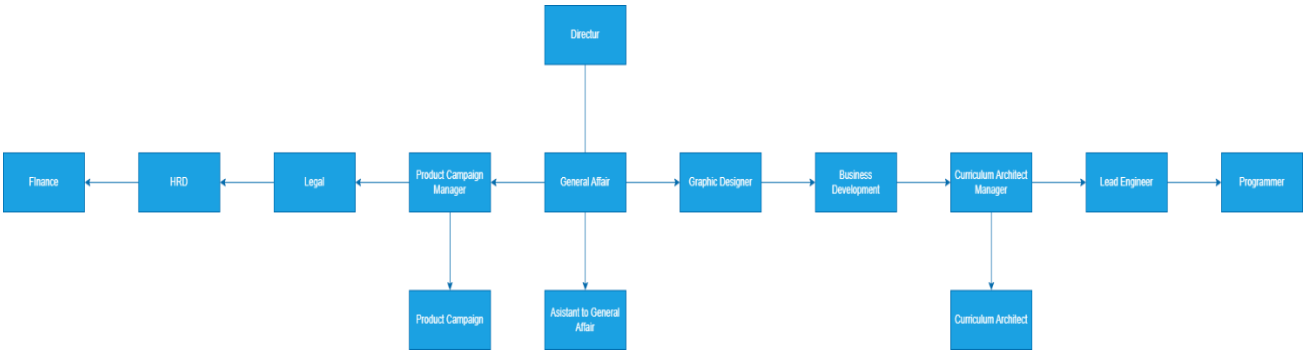
Proyek ini merupakan salah satu proyek di bawah naungan Kawadenan Radyakartiyasa. Kawadenan Radyakartiyasa berencana untuk melakukan pembaharuan aset keraton dan memperluas promosi destinasi wisata Kagungan Dalem Keraton Yogyakarta. Oleh karena itu, sebagai salah satu upaya untuk memperkenalkan berbagai destinasi wisata dan kegiatan budaya kepada masyarakat, Kawadenan Radyakartiyasa berniat untuk merancang satu *website* khusus. Destinasi wisata yang dikelola oleh pihak keraton meliputi empat unit utama yaitu Kedhaton, Wahanarata, Pagelaran, dan Tamansari.

Pihak keraton bekerja sama dengan PT Lumintu Logic dalam proses pengembangan *website* Radyakrtiyasa. PT Lumintu Logic adalah perusahaan teknologi informasi yang berasal dari Yogyakarta, Indonesia. PT Lumintu Logic menawarkan beberapa layanan berupa *information and technology consulting, software development*, pelatihan teknologi informasi sebagai institusi tersertifikasi, *repository* atau *e-office*. Selama PT Lumintu logic berdiri, mereka sudah menangani sekitar 65 proyek dari kurang lebih 14 klien.

PT Lumintu logic memiliki visi menjadi anggota aktif ekosistem global dalam rekayasa perangkat lunak sambil berkontribusi pada kemajuan teknologi dan penyebaran pengetahuan kepada generasi insinyur perangkat lunak di masa depan, serta misi menjadi perusahaan terkemuka yang menetapkan standar implementasi teknologi informasi secara nasional, sekaligus menjaga pengembangan sumber daya manusia. PT Lumintu logic juga memiliki nilai-nilai yang diterapkan yaitu ketenagakerjaan, kecerdasan, ketegasan, kreativitas, keefektifan, dan disiplin.

PT Lumintu Logic terdiri dari beberapa divisi kerja seperti Finance, HRD, Legal, Product Campaign Manager, General Affair, Graphic Designer, Business Development, Curriculum Architect Manager, Lead Engineer, dan juga Programmer yang dimana pemagang sendiri berada di divisi tersebut. Untuk divisi Programmer sendiri diketuai langsung oleh direktur. Divisi

Programmer di PT Lumintu Logic bertugas mengembangkan dan memelihara sistem serta aplikasi Perusahaan maupun klien agar berjalan dengan lancar. Divisi ini juga memastikan setiap perangkat lunak aman, dan mendukung kebutuhan operasional serta inovasi digital. Berikut ini struktur organisasi PT Lumintu Logic yang lebih terperinci.



Gambar 1. 1 Struktur Organisasi Perusahaan
(Company Profile PT Lumintu Logic, 2024)

Dalam pelaksanaan program magang ini, diharuskan untuk mengikuti fase *training* terlebih dahulu. Dimana fase *training* yang dilakukan bertujuan untuk mempelajari teknologi yang nantinya akan digunakan perusahaan selama magang. Setelah menyelesaikan masa *training*, dilakukan magang dengan posisi sebagai *backend developer* yang berada pada divisi Programmer. Tanggung jawab sebagai *Backend Developer* dalam proyek ini sendiri bertugas untuk merancang basis data *website* Radyakartiyasa yang dimiliki oleh keraton Yogyakarta.

Kontribusi yang dilakukan dalam perancangan basis data untuk proyek *website* Radyakartiyasa. Teknologi yang digunakan meliputi *headless CMS* Directus untuk merancang basis data dan Postman untuk menguji API. Seluruh proses bertujuan memastikan data dapat diakses dan dikelola dengan optimal melalui antarmuka API. Selain mengerjakan *project website*, pemagang juga berperan sebagai operator pada acara devs dialog dimana peran operator sendiri bertanggung jawab dalam penayangan *slide* presentasi pembicara.

1.2 Ruang Lingkup Magang

Pelaksanaan magang di PT Lumintu Logic Indonesia berlangsung dari tanggal 16 September 2024 sampai dengan 17 Maret 2025. Ruang lingkup proyek dan batasan aktivitas yang dilakukan selama pengembangan basis data untuk *website* Radyakartiyasa ini dijabarkan sebagai berikut:

1.2.1 Konfigurasi Directus

Tahap awal pengembangan basis data pada *website* Radyakartiyasa dimulai dengan melakukan konfigurasi pada *Headless CMS* Directus. Proses ini diawali dengan menyesuaikan *file* docker-compose.yaml, yaitu *file* yang digunakan untuk mengatur kerja Directus di dalam *container*. Di dalam *file* tersebut terdapat beberapa pengaturan seperti jaringan, *port*, dan lainnya. Penyesuaian dilakukan sesuai dengan kebutuhan proyek, contohnyanya menentukan nama layanan, lokasi penyimpanan data, dan juga akses pengguna. Tahap ini bertujuan agar Directus dapat berfungsi secara stabil dan siap untuk digunakan untuk pengelolaan data pada tahap selanjutnya.

1.2.2 Pembuatan Koleksi

Setelah tahap konfigurasi sistem, langkah berikutnya dalam pengembangan situs web Radyakartiyasa adalah pembuatan berbagai koleksi (*collections*) pada Directus yang disesuaikan dengan struktur dan kebutuhan konten *website*. Koleksi dalam Directus adalah tempat untuk menyimpan data, yang cara kerja atau fungsinya sama dengan tabel pada *database*. Setiap koleksi berisi kolom dan baris data, namun Directus menambahkannya dengan pengaturan antarmuka dan metadata sehingga pengelolaan struktur dan isinya dapat dilakukan tanpa menulis *query* SQL. Ketika sebuah koleksi dibuat, Directus juga membuat tabel yang sesuai di dalam *database.* Koleksi-koleksi ini dirancang untuk mengelola jenis data tertentu yang akan ditampilkan kepada pengguna.

a. *Navigation*

Digunakan untuk mengatur menu dan submenu pada halaman *website*.

b. *Destination Index* dan *Event Index*

Berfungsi untuk menampilkan daftar destinasi dan acara unggulan di halaman utama *website*.

c. *About Us*

Berisi informasi singkat mengenai sejarah dan budaya Keraton Yogyakarta.

d. *Contact Info*

Memuat informasi kontak resmi untuk memudahkan komunikasi dengan pihak keraton.

e. *Hero Banner*

Berguna sebagai penggabung dari relasi antara koleksi *backdrop* yang berisi media visual seperti gambar atau video dan koleksi *card* yang berisi judul dan arahan ke halaman konten destinasi.

f. *Frequently Asked Questions* (FAQ)

Menyediakan jawaban atas pertanyaan umum dari pengunjung mengenai Keraton Yogyakarta.

Pembuatan koleksi ini menjadi landasan penting dalam memastikan bahwa konten *website* dapat dikelola dengan baik dan ditampilkan secara efektif kepada pengguna.

1.3 Tujuan

Penulisan tugas akhir ini bertujuan untuk mendokumentasikan proses perancangan dan pengembangan basis data menggunakan *Headless CMS* Directus dalam pengembangan situs web Radyakartiyasa. Tujuan khusus dari penulisan ini mencakup:

a. Merancang struktur basis data yang mendukung pengelolaan konten.
b. Mendefinisikan cara kerja antar muka grafis Directus dalam merancang dan mengembangkan basis data tanpa perlu menulis *query* SQL secara manual.
c. Memastikan bahwa struktur basis data yang dibuat mampu memenuhi kebutuhan antarmuka *frontend*.
d. Menghasilkan basis data yang memudahkan pengelolaan konten dan mendukung efisiensi promosi destinasi serta *event* budaya.

1.4 Manfaat

Manfaat yang diperoleh dari penulisan tugas akhir ini antara lain:

a. Memberikan pemahaman mengenai perancangan struktur basis data, sehingga penulis dapat menyusun model data yang teratur dan sesuai dengan kebutuhan pengelolaan konten.
b. Mempermudah proses pengembangan basis data melalui antarmuka grafis Directus, sehingga dapat mengatur dan memodifikasi data tanpa harus menulis query SQL secara manual.
c. Mendukung proses integrasi dengan tampilan *website*, karena struktur basis data yang dibuat telah disesuaikan dengan kebutuhan *frontend* sehingga konten dapat ditampilkan dengan baik dan sesuai.
d. Bagi Kawedanan Radyakartiyasa, hasil pengembangan ini memberikan kemudahan dalam pengelolaan dan pembaruan konten *website* secara mandiri, sehingga kegiatan promosi destinasi dan *event* budaya dapat dilakukan dengan lebih efisien.

1.5 Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, ruang lingkup magang, tujuan, manfaat, dan sistematika penulisan.

BAB II LANDASAN TEORI DAN KAJIAN PUSTAKA

Bab ini memuat teori-teori yang mencakup definisi serta penjelasan yang diperoleh dari studi literatur dan berbagai sumber yang relevan dengan penyusunan laporan pengembangan basis data *website* Radyakartiyasa.

BAB III PELAKSANAAN MAGANG

Bab ini berisi dokumentasi aktivitas pengerjaan tugas yang dilakukan ketika pelaksanaan magang.

BAB IV REFLEKSI PELAKSANAAN MAGANG

Bab ini mencakup manfaat-manfaat yang diperoleh penulis selama melaksanakan kegiatan magang di PT Lumintu Logic Indonesia.

BAB V PENUTUP

Bab ini berisi kesimpulan dari pembahasan yang telah disampaikan pada bab sebelumnya, dan saran untuk pengembangan selanjutnya agar dapat menjadi lebih baik.

DAFTAR PUSAKA

LAMPIRAN

BAB II
DASAR TEORI

2.1 Website

Di era transformasi digital, pengembangan *website* dan sistem informasi berbasis web menjadi langkah strategis bagi berbagai organisasi untuk meningkatkan efisiensi operasional dan jangkauan publik (Ayurira & Fajri, 2024). Bagi institusi di sektor pariwisata dan kebudayaan, *website* memegang peran vital sebagai media utama untuk promosi, diseminasi informasi, dan interaksi dengan masyarakat (Surentu et al., 2020). Pengelolaan institusi budaya yang kompleks, seperti yang dijalankan oleh Kawedanan Radyakartiyasa, juga menuntut adanya *platform* informasi modern untuk mendukung fungsi manajemen dan pelayanannya (Pritandari et al., 2024). Secara umum, keberadaan *platform* digital yang responsif merupakan solusi untuk meningkatkan kualitas layanan (Chasanah et al., 2025) dan mengelola reservasi fasilitas secara efisien (Maulana et al., 2024).

Website merupakan sebuah kumpulan halaman digital yang umumnya dibuat menggunakan bahasa HTML dan diakses melalui jaringan internet dengan bantuan *web browser.* Koneksi dengan jaringan internet sendiri berfungsi agar *website* dapat diakses oleh siapa saja di seluruh dunia (Sari & Suhendi, 2020). *Website* biasanya berada dalam satu domain atau subdomain, dan dimiliki oleh individu, kelompok, atau institusi (Miftahuljannah & Suharso, 2023). Kumpulan halaman ini saling terhubung melalui *hyperlink* dan menyajikan berbagai jenis informasi, seperti teks, gambar, animasi, audio, maupun video, baik dalam bentuk statis maupun dinamis, sehingga membentuk satu kesatuan yang terstruktur (Nana et al., 2022).

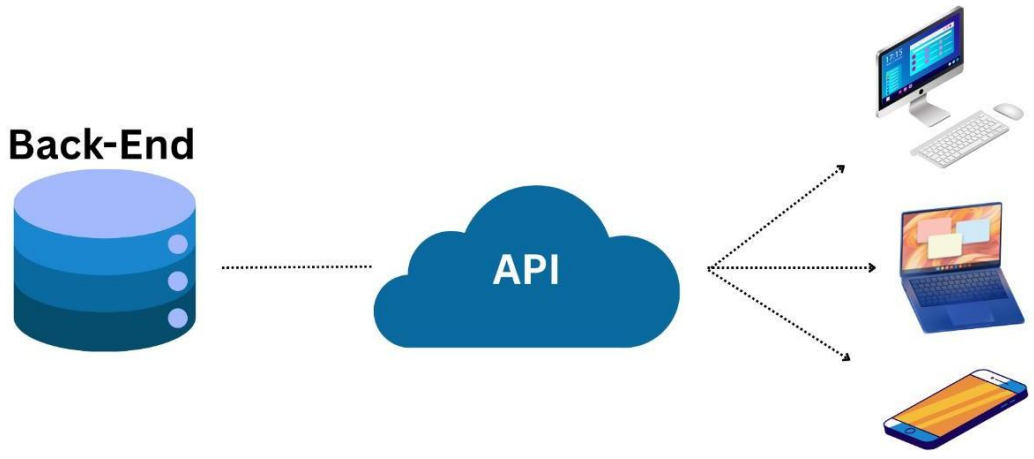
Dalam perkembangannya, *website* tidak hanya digunakan sebagai media informasi, tetapi juga sebagai sarana promosi, komunikasi, hingga layanan interaktif. Berbagai sektor, seperti pendidikan, bisnis, pemerintahan, dan hiburan, memanfaatkan *website* untuk mendukung aktivitas mereka. Hal ini menunjukkan bahwa *website* memiliki peran penting dalam kehidupan digital masyarakat modern. Oleh karena itu, pemahaman tentang jenis dan fungsi website menjadi semakin relevan dalam era transformasi digital saat ini.

2.2 Headless CMS

Headless CMS (Content Management System) adalah sistem manajemen konten yang memisahkan antara bagian pengelolaan konten (*backend*) dan bagian tampilan atau desain

(*frontend*) (Arifiany & Kusuma, 2025). Sistem ini hanya fokus pada bagaimana konten dibuat dan disimpan di bagian belakang, tanpa mengatur bagaimana konten itu akan ditampilkan. Konten yang sudah dibuat kemudian dapat diakses melalui *Application Programming Interface* (API) dan ditampilkan di berbagai perangkat seperti *website*, aplikasi *mobile*, atau *platform* digital lainnya. Karena konten di dalamnya tidak terikat dengan tampilan tertentu, editor dapat mengelola dan menerbitkan konten tanpa harus terlibat dengan aspek visual atau tata letak antarmuka pengguna. Selain itu, sistem ini juga memudahkan pengembangan *website* modern dan dapat berfungsi sebagai pusat distribusi konten ke banyak saluran sekaligus.

Konsep *Headless CMS* menekankan pemisahan antara sistem pengelolaan konten dan penyajiannya, di mana konten hanya disimpan serta dikelola tanpa terikat pada tampilan bawaan, sehingga pengembang bebas menentukan teknologi *front-end* yang ingin digunakan seperti React, Vue, atau AstroJS untuk menampilkan data sesuai kebutuhan. Pendekatan ini memberikan keunggulan berupa fleksibilitas, skalabilitas, dan efisiensi dalam distribusi konten lintas *platform*, baik pada aplikasi web, *mobile*, maupun perangkat IoT (Berndtsson & Fagerberg, 2025). Selain itu, arsitektur *Headless* dinilai sangat mendukung pengembangan sistem multilingual, karena struktur basis data dapat dirancang secara adaptif untuk mengakomodasi konten dalam berbagai bahasa.



Gambar 2. 1 Cara Kerja *Headless CMS*
(Emadamerho-Atori, 2023)

Dalam arsitektur *Headless CMS*, konten disimpan di bagian *back-end* sebagai repositori utama dan disalurkan ke berbagai *platform* melalui API. Pendekatan ini memisahkan antara pengelolaan konten dan tampilan, sehingga memberikan fleksibilitas dalam penyajian di

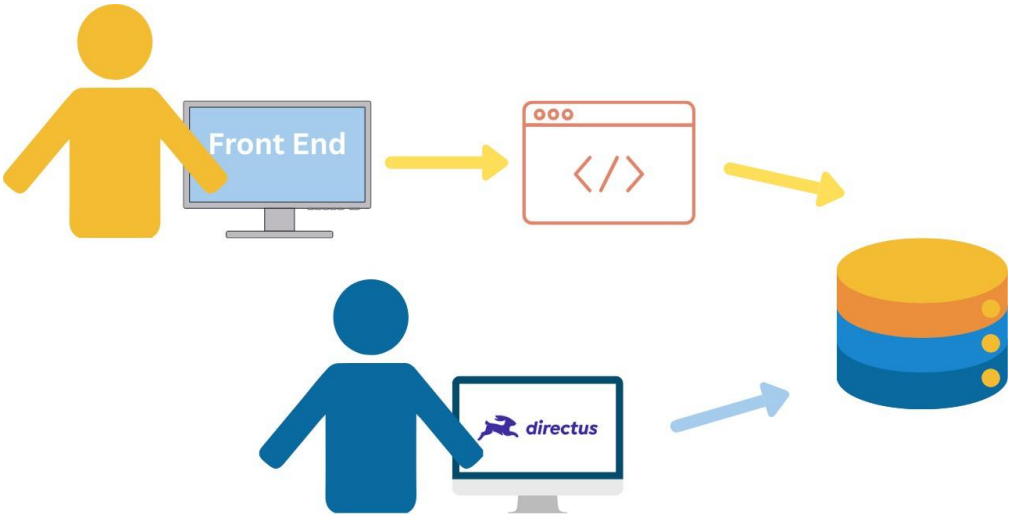
berbagai perangkat seperti *website*, IoT, atau asisten digital. Istilah *Headless* menunjukkan tidak adanya *front-end* bawaan seperti pada CMS tradisional. Dengan kebebasan memilih *framework* seperti React, Angular, atau Vue, pengembang dapat merancang antarmuka yang lebih dinamis dan sesuai dengan kebutuhan pengguna (Eassa, 2025).

2.3 Directus

Directus adalah sistem manajemen konten berbasis *Headless CMS* yang memungkinkan pengguna untuk mengelola data secara efisien tanpa terikat pada antarmuka tampilan tertentu. Sistem ini dirancang untuk bekerja langsung dengan basis data relasional yang telah ada, sehingga tidak memerlukan migrasi atau perubahan skema yang kompleks. Salah satu karakteristik utama Directus adalah kemampuannya dalam menyajikan data melalui antarmuka API, baik menggunakan REST maupun GraphQL, yang menjadikannya sangat fleksibel untuk digunakan di berbagai jenis aplikasi modern.

Dalam implementasinya, Directus menawarkan antarmuka grafis berbasis web yang memudahkan pengguna non-teknis dalam mengelola data. Meskipun demikian, sistem ini tetap memberikan kendali penuh kepada pengembang terhadap struktur data, arsitektur basis data, serta integrasi API yang dihasilkan. Selain itu, fitur-fitur seperti manajemen peran pengguna, kontrol akses yang terperinci, autentikasi bawaan, serta sistem ekstensi dan *webhook* menjadikan Directus sebagai *platform* yang adaptif terhadap berbagai kebutuhan proyek.

Penelitian Sobri et al. (2022) menyatakan bahwa Directus mampu memenuhi kebutuhan dasar sistem dan menawarkan fitur tambahan yang tidak dimiliki oleh *platform* lain seperti Strapi dan Supabase. Penelitian tersebut juga menekankan keunggulan Directus dalam hal fleksibilitas struktur data dan kemudahan dalam mengintegrasikan API yang bisa digunakan untuk berbagai jenis proyek berbasis web. Selain itu, Directus dinilai unggul dalam efisiensi pengembangan karena pengelolaan data dan pembuatan API dapat dilakukan tanpa penulisan kode manual, sehingga mempercepat implementasi sistem. Hasil penelitian menunjukkan bahwa Directus bukan hanya sekadar CMS, tetapi juga *platform* pengelolaan data modern yang adaptif terhadap berbagai kebutuhan pengembangan teknologi informasi.



Gambar 2. 2 Arsitektur Directus

(Mohammed, 2024)

Directus bekerja dengan arsitektur data yang memungkinkan pengguna menghubungkannya langsung dengan struktur basis data yang ada tanpa harus mengubah skema aslinya. Sistem ini secara otomatis membentuk tipe data serta menghasilkan *endpoint* GraphQL dan RESTful API sesuai dengan struktur *database* tersebut. Directus juga mendukung migrasi skema basis data dengan mendeteksi dan menyesuaikan perbedaan antar sistem secara otomatis (Sobri et al., 2022).

2.4 *Agile*

Dalam pembangunan basis data untuk *website* Radyakartiyasa, digunakan pendekatan *Agile* sebagai metode pengembangannya. Metode *Agile* adalah salah satu pendekatan dalam *Software Development Life Cycle* (SDLC)(Arnomo & Kurniawan, 2024). Dimana *Agile* merupakan metode yang berfokus pada proses iteratif dan bertahap dengan penekanan pada kolaborasi tim serta kemampuan beradaptasi terhadap perubahan kebutuhan(Ariesta et al., 2021). Setiap tahapan pengembangan dilakukan secara berulang, memungkinkan evaluasi dan penyempurnaan sistem secara berkelanjutan sepanjang proyek berlangsung. Setiap bagian akan dievaluasi dan disempurnakan secara terus-menerus. Dengan cara ini, *Agile* memungkinkan tim pengembang untuk lebih cepat merespons perubahan kebutuhan atau permintaan dari pengguna selama proses pengembangan berlangsung (Amarta & Anugrah, 2021).

Metode *Agile* sendiri terdiri dari berbagai kerangka kerja yang medukung pengembangan perangkat lunak agar lebih cepat, efisien, dan responsif terhadap perubahan. Beberapa jenis *Agile* yang paling umum digunakan antara lain *Scrum*, *Kanban*, *Extreme Programming* (XP), dan *Lean Software Development* yang memiliki cara kerja dan karakteristik yang berbeda. Meskipun berbeda dalam penerapannya, seluruh metode tersebut tetap berpegang pada prinsip utama *Agile* yaitu iterasi singkat, kolaborasi tim, serta kemampuan untuk menyesuaikan diri dengan perubahan sesuai kebutuhan selama proses pengembangan. Studi oleh (Maghfirah & Suranto, 2023) menunjukkan bahwa metode-metode *Agile* ini banyak diterapkan di Indonesia, terutama dalam proyek-proyek yang membutuhkan kecepatan dan fleksibilitas tinggi.

Dalam konteks proyek pengembangan *website* Radyakartiyasa, penerapan metode *Agile* membantu tim untuk bekerja secara kolaboratif, dan dapat beradaptasi dengan perubahan sesuai dengan kebutuhan selama masa pengerjaan proyek. Setiap tahap pengembangan memungkinkan pengembang dalam mengevaluasi fungsi sistem, memperbaiki kesalahan, serta menambahkan fitur baru berdasarkan masukan atau kebutuhan. Proses ini sesuai dengan tujuan utama pengembangan berbasis *Agile*, yaitu menghasilkan perangkat lunak yang sesuai dengan kebutuhan pengguna. Dengan demikian, penerapan metode *Agile* tidak hanya meningkatkan efektivitas pengembangan, tetapi juga membangun kerja sama yang produktif antara tim.

2.5 *Scrum*

Scrum adalah salah satu metode yang menerapkan pendekatan *Agile* dalam proses pengembangan perangkat lunak. *Agile* berfungsi sebagai seperangkat prinsip yang mendorong penyelesaian masalah secara adaptif dan fleksibel, sementara *Scrum* menjadi salah satu bentuk implementasi nyata dari prinsip-prinsip tersebut melalui mekanisme kerja yang sistematis dan berulang (Gutama & Dirgahayu, 2021). Keunggulan *Scrum* terletak pada kemampuannya untuk merespons perubahan kebutuhan dengan cepat dan menghasilkan produk yang berkualitas melalui pertemuan rutin seperti perencanaan sprint dan evaluasi harian. Penelitian oleh (Dzaky & Kurniawan, 2023) menunjukkan bahwa *Scrum* dapat meningkatkan produktivitas tim dan memberikan solusi yang cepat terhadap perubahan kebutuhan dalam pengembangan sistem informasi.

Selain itu, (Ariesta et al., 2021) juga menekankan pentingnya *Scrum* dalam mempercepat penyelesaian tugas dan memastikan bahwa proyek dapat disesuaikan dengan kebutuhan pengguna. Dalam rekayasa perangkat lunak, metodologi *Agile Scrum* telah diakui secara luas sebagai pendekatan yang lebih efisien dibandingkan metode tradisional seperti *Waterfall*,

terutama untuk proyek-proyek yang membutuhkan fleksibilitas (Renata et al., 2022). *Scrum* bekerja dengan cara membagi proses pengembangan menjadi beberapa tahap pendek yang disebut *sprint*, biasanya berlangsung antara dua hingga empat minggu. Di awal setiap sprint, tim akan merencanakan apa saja yang ingin diselesaikan. Setiap *sprint* memungkinkan tim untuk fokus pada bagian tertentu dari proyek, dengan tujuan untuk meningkatkan kolaborasi, fleksibilitas, dan pemantauan kemajuan.

Selama *sprint* berjalan, tim melakukan pertemuan singkat setiap hari untuk saling berbagi progres dan kendala yang dihadapi. Setelah *sprint* selesai, tim akan mengevaluasi hasil kerja dan mendiskusikan apa yang bisa diperbaiki untuk *sprint* berikutnya. Dengan proses yang berulang dan teratur ini, *Scrum* membantu tim bekerja lebih fokus, terbuka satu sama lain, dan cepat beradaptasi jika ada perubahan.

Proses kerja dalam *Scrum* diatur melalui serangkaian tahapan formal yang dirancang untuk inspeksi dan adaptasi. Berikut merupakan tahapan-tahapan dalam metode *Scrum* yang dilakukan dalam pengembangan *website* Radyakartiyasa:

a. Product Backlog

Product Backlog merupakan daftar prioritas kebutuhan produk yang digunakan sebagai dasar pekerjaan tim. Seluruh pekerjaan yang akan dilakukan dimuat dalam *Product Backlog*, yang menjadi satu-satunya sumber rujukan bagi tim (Arnomo & Kurniawan, 2024). Item di dalamnya disempurnakan secara bertahap melalui diskusi untuk memperjelas detail, prioritas, dan estimasi pengerjaan (Schwaber & Sutherland, 2020). *Backlog* berfungsi sebagai acuan utama dalam merancang rencana kerja melalui metode sprint, yang mencakup kebutuhan pengguna baik fungsional maupun non-fungsional, serta disusun berdasarkan tingkat prioritas (Ayurira & Fajri, 2024). Umumnya, tim akan memprioritaskan *product backlog* saat pengerjaan tugas di *sprint planning* kedepannya.

b. Sprint Planning

Setiap *sprint* kemudian dimulai dengan *sprint planning*, yaitu tahapan dimana tim merencanakan pekerjaan yang akan diselesaikan dalam periode *sprint* tersebut. Jika terdapat *backlog*, tim pengembang melakukan peninjauan terhadap daftar *product backlog* untuk membahas dan memahami tujuan dari setiap fitur yang akan dikerjakan (Usman et al., 2021). *Sprint planning* juga menjadi kesempatan bagi anggota tim untuk mendiskusikan serta memahami *item-item backlog* yang akan diselesaikan selama masa *sprint*.

Dalam implementasinya, kegiatan *Sprint Planning* dilakukan tim setiap hari selasa untuk menentukan tugas yang harus dikerjakan dalam seminggu kedepan. Kegiatan ini menjadi

forum rutin bagi tim pengembang untuk menentukan tugas-tugas yang akan diselesaikan dalam satu minggu ke depan, serta membagi tanggung jawab pekerjaan sesuai keahlian anggota tim. Setiap *task* yang telah ditentukan kemudian dimasukkan ke dalam sistem manajemen proyek sebagai acuan selama masa *sprint.*

c. Sprint

Sprint merupakan tahap dimana seluruh aktivitas pengembangan dijalankan berdasarkan item-item yang telah ditentukan pada tahap sebelumnya (Arsa et al., 2024). Pada tahap ini, tim pengembang focus menyelesaikan *task* yang telah diprioritaskan dalam *product backlog* agar dapat diselesaikan sesuai dengan target waktu yang telah ditentukan. Setiap *sprint* umumnya memiliki durasi waktu yang singkat, dalam proyek ini tahap *sprint* berlangsung selama lima hari. Dalam pelaksanaannya, setiap anggota tim bertanggung jawab untuk menyelesaikan bagian tugas yang menjadi bagiannya dan melaporkan progresnya secara rutin.

Selama pelaksanaan *sprint*, tim juga selalu berdiskusi satu sama lain terkait dengan tugas yang sedang dikerjakan. Diskusi ini bertujuan agar tidak terjadi kesalahpahaman dalam implementasi fitur dan untuk memastikan setiap anggota memiliki pemahaman yang sama terhadap prioritas pekerjaan. Dengan demikian, tahapan *sprint* bukan hanya sekadar fase pengerjaan teknis, tetapi juga proses kolaboratif yang memastikan sistem dikembangkan secara konsisten, efisien, dan selaras dengan tujuan proyek.

d. Daily Scrum

Daily Scrum adalah kegiatan harian yang dilakukan oleh seluruh anggota tim untuk meninjau progres pekerjaan dan membahas kendala yang sedang dihadapi. Selama *sprint* berlangsung, kegiatan *daily scrum* berfungsi sebagai forum komunikasi singkat yang memastikan status perngerjaan tugas seluruh anggota tim. Kegiatan ini bertujuan untuk memberikan gambaran terbaru mengenai kondisi pengembangan sistem ke seluruh anggota tim (Maulana et al., 2024). Dengan adanya *daily scrum*, tim dapat dengan cepat mengidentifikasi permasalahan yang muncul dan mencari solusi bersama sebelum masalah tersebut berdampak pada keberlangsungan *sprint.*

Dalam konteks pengembangan *website* Radyakartiyasa, *daily scrum* dilaksanakan secara rutin setiap pagi sebelum memulai pekerjaan. Setiap anggota tim menyampaikan laporan kemajuan tugas serta menginformasikan jika terdapat kendala teknis yang dihadapi selama pengerjaan tugas. Hasil diskusi dari *daily scrum* kemudian digunakan untuk menyesuaikan prioritas harian.

e. Sprint Review

Di akhir tahapan *sprint*, dilaksanakan tahap *sprint review* untuk mendemonstrasikan hasil kerja kepada pemangku kepentingan dan mendapatkan umpan balik (Afriyanto & Noviandi, 2023). Pada tahap ini, tim pengembang proyek secara rutin melakukan evaluasi terhadap penyelesaian item-item dalam *product backlog*. Kegiatan ini dilakukan setiap hari Kamis dan berfungsi sebagai forum untuk mengidentifikasi hambatan yang dihadapi selama pelaksanaan *sprint* serta mendiskusikan solusi yang telah diimplementasikan. Pelaksanaan evaluasi berkala semacam ini dapat meningkatkan efektivitas kolaborasi tim dan mendukung pencapaian tujuan proyek.

f. Sprint Retrospective

Sprint review diikuti oleh *sprint retrospective*, di mana tim berefleksi untuk meningkatkan proses kerja pada *sprint* berikutnya. Tujuan dari dilakukannya tahap ini adalah untuk mendorong perbaikan proses, meningkatkan efektivitas kerja, dan menghasilkan kualitas yang lebih baik pada sprint selanjutnya (Chasanah et al., 2025). Tahap ini merupakan bagian penting dalam proses pengembangan sistem, di mana tim melakukan evaluasi terhadap hasil kerja untuk mengidentifikasi adanya kekurangan atau hambatan. Jika ditemukan permasalahan, maka akan segera dirumuskan solusi yang tepat guna menyempurnakan sistem yang telah dikembangkan (Renata et al., 2022).

2.6 Tinjauan Pustaka

Berikut ini beberapa penelitian yang sebelumnya telah dilakukan dan membahas topik mengenai pengembangan basis data.

Tabel 2.1 Penelitian Sejenis

No	Judul	Penulis	Metode	Hasil Penelitian
1.	*Leveraging Headless Content Management System As A Service In A Service-Based Architecture: Enhancing User Experience And Overcoming Resource Limitations For*	Ilma Arifiany, Gede Putra Kusuma	Wawancara	Penelitian ini menyimpulkan bahwa *Headless CMS* dapat digunakan sebagai fondasi dalam membangun arsitektur berbasis layanan yang sederhana namun tetap *scalable* untuk startup. Dengan memanfaatkan fitur plugin dan *database* pada *Headless CMS*, pengembang dapat membangun layanan modular tanpa kompleksitas tinggi seperti pada *microservices*. Pendekatan ini mempercepat proses pengembangan, mengurangi beban infrastruktur, dan memfokuskan tim pada peningkatan

No	Judul	Penulis	Metode	Hasil Penelitian
	Start-Ups (2025)			pengalaman pengguna. Implementasi arsitektur ini juga menghasilkan performa sistem yang cepat, throughput tinggi, dan waktu respons yang baik. Secara keseluruhan, *Headless CMS* memberikan solusi praktis bagi *startup* yang membutuhkan arsitektur fleksibel dengan sumber daya terbatas (Arifiany & Kusuma, 2025).
2.	*Comparison between Headless CMS and Backend-as-a-Service Products for E-Suripreneur Backend (2022)*	Nur Ayuni Nor Sobri, Mohamad Aqib Haqmi Abas, Ahmad Ihsan Mohd Yassin, Megat Syahirul Amin Megat Ali, Nooritawati Md Tahir,, Azlee Zabidi, Zairi Ismael Rizman	Studi Komparatif	Studi ini membandingkan Strapi, Directus, dan Supabase untuk *backend startup*. Hasilnya, Directus paling sesuai untuk sistem *e-Suripreneur* karena memenuhi kebutuhan CRUD, otentikasi, migrasi database, kemampuan self-hosting, fleksibilitas custom code, dan penyediaan REST API. Directus unggul sebagai solusi *Headless* yang cepat dikembangkan dan mudah diintegrasikan (Sobri et al., 2022).
3.	*Integrating Social Media Functionality into a Headless CMS: A Unified Approach to Content Management and Publishing Workflows (2025)*	Arvid Berndtsson, Ellie Fagerberg	*design and creation research*	Studi ini mengembangkan plugin modular pada *Payload CMS* untuk integrasi sosial media secara langsung. Plugin memungkinkan pembuatan dan distribusi konten *multichannel* dari satu CMS. Hasil evaluasi menunjukkan sistem ini potensial untuk strategi marketing digital meski terdapat tantangan seperti kompleksitas arsitektur dan perbedaan perspektif antara *marketer* dan *developer* (Berndtsson & Fagerberg, 2025).
4.	*Enhancing E-commerce Footwear Business: A Headless Implementation*	Devender Yadav	Studi Literatur	*Headless* architecture memungkinkan pemisahan frontend–backend sehingga UI dapat dikembangkan lebih bebas dan responsif. Hasil penelitian menunjukkan keuntungan utama

No	Judul	Penulis	Metode	Hasil Penelitian
	Approach for Website and App Development (2021)			berupa fleksibilitas, personalisasi pengalaman pengguna, skalabilitas tinggi, serta dukungan omnichannel melalui API-first. Pendekatan ini meningkatkan efisiensi pengembangan dan kepuasan pelanggan pada bisnis e-commerce footwear (Yadav, 2021).
5.	*Using Headless Solution For E-Commerce Website Implementation (2025)*	Valtteri Hietala	*Research and Development*	Penelitian ini mengimplementasikan *e-commerce* berbasis *Headless* menggunakan BigCommerce, React, dan Node/Express. Hasilnya, *Headless* commerce memberikan fleksibilitas tinggi, pemisahan frontend-backend yang memudahkan pengembangan, performa lebih baik, serta skalabilitas untuk kebutuhan toko *online*. Dokumentasi BigCommerce mendukung proses integrasi dan mempermudah pengembangan fitur checkout, katalog, dan pembayaran (Hietala, 2025).
6.	Pengaruh Implementasi *Headless Content Management System* Terhadap Performa Dan Kinerja *E-Learning* (2025)	Muhammad Ramzi Adhitya, I Made Suartana	*Research and Development*	Penelitian ini menunjukkan bahwa implementasi *Headless CMS* pada *platform e-learning* mampu meningkatkan performa sistem secara signifikan. Pemisahan antara backend dan frontend memungkinkan proses pemuatan konten menjadi lebih cepat dan stabil, bahkan pada ukuran data besar. Pengujian menggunakan Lighthouse menghasilkan metrik FCP, LCP, TBT, dan Speed Index yang tetap konsisten di bawah 1,5 detik dengan skor performa di atas 90. Hasil ini membuktikan bahwa arsitektur *Headless* memberikan respons yang lebih efisien dibandingkan pendekatan *CMS* tradisional. Secara keseluruhan, *Headless CMS* efektif dalam meningkatkan kecepatan, stabilitas, dan pengalaman pengguna pada sistem *e-learning* (Adhitya & Suartana, 2025).

No	Judul	Penulis	Metode	Hasil Penelitian
7.	Pengembangan Basis Data Multilingual Menggunakan *Headless CMS* Directus Pada *Website* Radya Kartiyasa (2025)	Tegar Bagus Hartadi, Nur W. Rahayu	*Agile Scrum*	Penelitian ini membuktikan bahwa Directus mampu menjadi solusi pengelolaan basis data multilingual untuk kebutuhan *website* budaya. Directus memberikan fleksibilitas tinggi, kemudahan konfigurasi, integrasi API, serta mempercepat pengembangan tanpa membangun backend dari awal. Scrum terbukti membantu proses kerja tim menjadi lebih terstruktur dan adaptif. Sistem multilingual memungkinkan penyampaian informasi budaya Keraton Yogyakarta dapat menjangkau audiens global (Hartadi & Rahayu, 2025).
8.	*Optimizing Web Application Development: A Proposed Architecture Integrating Headless CMS and Serverless Computing (2025)*	Ahmed M. Eassa	Studi literatur	Penelitian ini menjelaskan konsep, arsitektur, dan keunggulan *Headless CMS* dibandingkan *CMS* tradisional. *Headless CMS* dinilai mampu menyediakan fleksibilitas tinggi melalui pendekatan API-first. Peneliti menunjukkan bahwa arsitektur ini mendukung integrasi multi-*platform* dan pengembangan antarmuka yang independen. Secara keseluruhan, *Headless CMS* dianggap sebagai solusi modern untuk kebutuhan pengelolaan konten yang lebih dinamis (Eassa, 2025).
9.	*Exploring WebAssembly, Progressive Web Apps, and Headless CMS for Cutting-Edge Web Development (2023)*	Tushar Singh	Studi literatur	Penelitian ini mengkaji potensi integrasi WebAssembly, PWA, dan *Headless CMS* dalam pengembangan web modern. WebAssembly dapat meningkatkan performa aplikasi, sedangkan PWA menyediakan pengalaman pengguna yang responsif. *Headless CMS* berperan sebagai penyedia konten fleksibel melalui API. Integrasi ketiganya dinilai mampu menghasilkan aplikasi web yang lebih cepat dan efisien (Singh, 2023).
10.	Interoperabilitas Sistem Informasi Profil	Rudy Tandra, Debbie	*Agile Scrum*	Penelitian ini membangun sistem informasi profil jurusan yang terhubung dengan *Headless CMS*

No	Judul	Penulis	Metode	Hasil Penelitian
	Jurusan Berbasiskan *Headless Content Management System Website* Jurusan	Yuari Siallagan, Era Prestoroika		agar konten dapat ditampilkan secara otomatis ke TV display dan kiosk. Proses pengembangan dilakukan dengan metode *Agile–Scrum* sehingga sistem dapat dikembangkan secara iteratif dan terstruktur. Pengujian menggunakan SUS menghasilkan nilai 77,29 yang masuk kategori good dan layak digunakan. Penelitian ini membuktikan bahwa *Headless CMS* efektif untuk mendukung pengelolaan konten *multi-channel* (Tandra et al., 2025).

Berdasarkan kajian terhadap sepuluh penelitian terdahulu, terdapat kesamaan dalam penggunaan arsitektur *Headless CMS* sebagai pendekatan alternatif untuk membangun sistem pengelolaan konten yang lebih fleksibel, terstruktur, dan mudah diintegrasikan. Penelitian yang dilakukan oleh Tandra et al., (2025) maupun Hartadi & Rahayu, (2025) memiliki relevansi yang kuat karena sama-sama menerapkan metode *Agile Scrum* serta memanfaatkan *Headless CMS* sebagai fondasi pengembangan sistem. Perbedaannya terletak pada tujuan implementasi yaitu penelitian tersebut fokus pada kompatibilitas tampilan dan pengelolaan konten organisasi, sedangkan penelitian ini berfokus pada perancangan dan pembangunan basis data *website* Radyakartiyasa agar lebih terstruktur dan mudah dikembangkan.

Penelitian lain seperti yang ditulis oleh Eassa (2025), Singh (2023), Sobri et al. (2022), Berndtsson & Fagerberg (2025), Yadav (2021), dan Hietala (2025) memberikan landasan teoritis mengenai konsep *Headless CMS*, keunggulan arsitektur API-*first*, serta penerapannya pada berbagai domain seperti *e-commerce*, integrasi media sosial, maupun optimalisasi performa aplikasi. Walaupun tidak seluruhnya melakukan implementasi langsung, pembahasan tersebut memperkuat dasar pemilihan *Headless CMS* dalam penelitian ini, terutama terkait fleksibilitas *backend* dan kemudahan integrasi dengan *frontend* yang beragam.

Sementara itu, penelitian Arifiany & Kusuma (2025) dan Adhitya & Suartana (2025), menunjukkan bahwa *Headless CMS* dapat digunakan untuk membangun sistem berbasis API dengan struktur data yang dapat dikembangkan secara modular. Walaupun konteks aplikasinya berbeda seperti pembuatan *e-learning* atau manajemen konten berbasis React.js, konsep yang diusung sejalan dengan tujuan penelitian ini, yaitu menciptakan basis data yang rapi, konsisten, dan mudah dikembangkan melalui *platform Headless CMS*.

Secara keseluruhan, penelitian-penelitian terdahulu memberikan gambaran bahwa Headless CMS merupakan pendekatan yang relevan untuk pengembangan sistem informasi modern. Namun, penelitian ini memiliki perbedaan yang jelas, yaitu berfokus secara khusus pada pembangunan basis data untuk *website* Radyakartiyasa menggunakan Directus, dengan struktur konten yang menyesuaikan kebutuhan penyajian informasi budaya dan kegiatan organisasi. Dengan demikian, penelitian ini memberikan kontribusi praktis berupa perancangan dan implementasi basis data yang lebih terarah untuk kebutuhan *website* Radyakartiyasa.

BAB III
PELAKSANAAN MAGANG

PT Lumintu Logic adalah Perusahaan Teknologi Informasi yang berasal dari Yogyakarta, Indonesia. Lumintu menawarkan beberapa layanan berupa *Information and Technology consulting*, *software development*, pelatihan teknologi informasi sebagai institusi tersertifikasi, *repository* atau *e-office*. Lumintu Logic pertama kali didirikan pada tahun 2009 dengan bentuk badan usaha CV (*Commanditaire Venootschap*). Pada tahun 2022, Lumintu Logic meningkatkan status hukumnya menjadi PT (Perseroan Terbatas) sebagai respons terhadap perluasan area bisnisnya. Hingga akhir tahun 2023, PT Lumintu Logic berhasil mendaftarkan diri sebagai Lembaga Pelatihan Profesi di bidang Teknologi Informasi. Sebagai lembaga yang berkomitmen memajukan industri teknologi informasi, PT Lumintu Logic menyediakan pelatihan inovatif untuk mempersiapkan individu menghadapi perkembangan dunia teknologi. Dengan dukungan instruktur ahli dan kurikulum terkini, PT Lumintu Logic bertujuan untuk meningkatkan potensi sumber daya manusia, khususnya di bidang teknologi informasi, agar mampu bersaing di pasar global.

Pada kegiatan magang, terdapat beberapa fase atau tahapan yang dilakukan. Dimulai dari *welcoming, training, exit interview*, pengerjaan proyek magang, hingga konsultasi.



Gambar 3.1 Flow Tahapan Magang

Berikut ini merupakan *timeline* kegiatan magang, kegiatan magang sendiri berlangsung selama 6 bulan. Jadwal yang ada pada *timeline* dibagi menjadi perminggu pada setiap bulan, sebagai mana yang ditampilkan pada gambar di bawah:

[Web Radyakartiyasa] Project Schedule

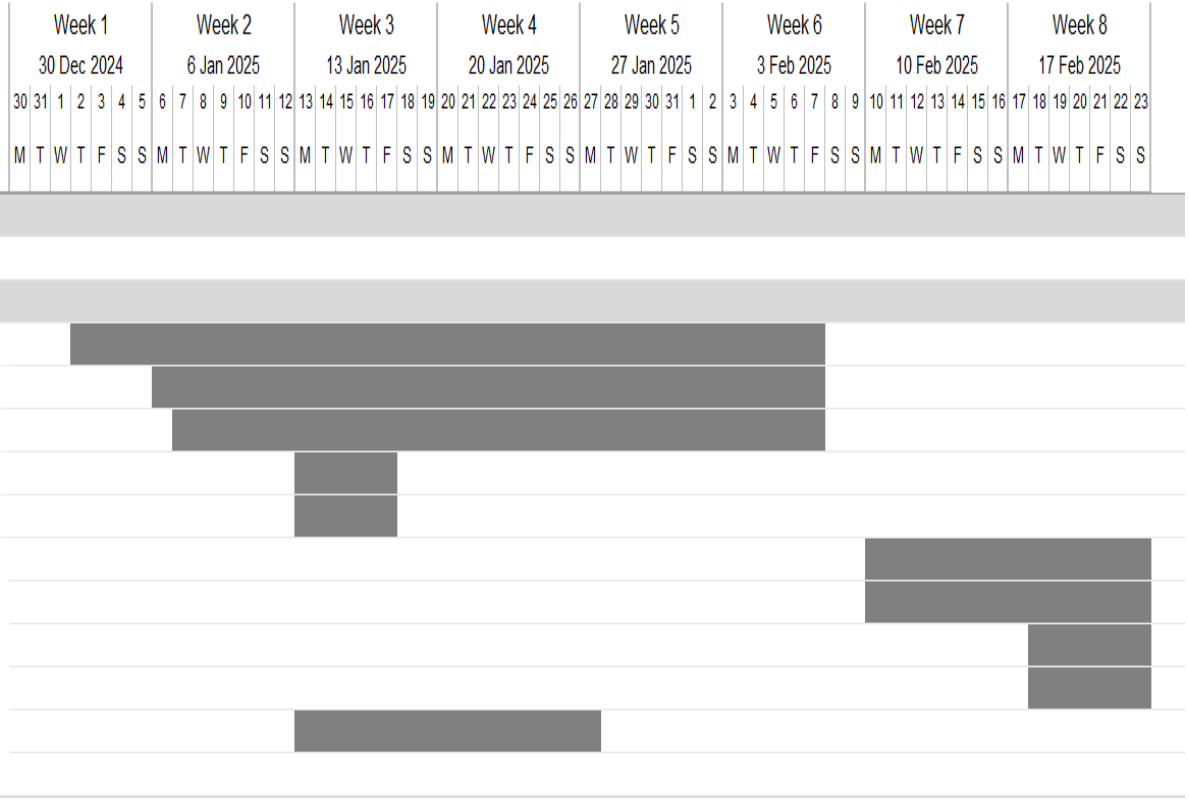
Gantt Chart Template © 2006-2025 by Vertex42.com.

PT Lumintu Logic

Project Start Date 1/2/2025 (Thursday) Display Week 1

Project Lead Happy Hartana

WBS	TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
1	**Training**			-			-
1.1	Masa Training	Happy	Thu 9/19/24	Thu 12/19/24	92	100%	66
2	**Pelaksanaan Project**			-			-
2.1	Create Navigation Collection		Thu 1/02/25	Fri 2/07/25	37	100%	27
2.2	Create Destination Index Collection		Mon 1/06/25	Fri 2/07/25	33	100%	25
2.3	Create Event Index Collection		Tue 1/07/25	Fri 2/07/25	32	100%	24
2.4	Create About Us Collection		Mon 1/13/25	Fri 1/17/25	5	100%	5
2.5	Create Contact Info Collection		Mon 1/13/25	Fri 1/17/25	5	100%	5
2.6	Create Hero (backdrop) Collection		Mon 2/10/25	Tue 2/25/25	16	100%	12
2.7	Create Hero (card) Collection		Mon 2/10/25	Tue 2/25/25	16	100%	12
2.8	Create FAQ Collection		Tue 2/18/25	Tue 2/25/25	8	100%	6
2.9	Create QnA Collection		Tue 2/18/25	Tue 2/25/25	8	100%	6
2.10	Create Home Gallery Collection		Mon 1/13/25	Mon 1/27/25	15	100%	11
2.11	Access Policies		Wed 2/26/25	Fri 2/28/25	3	100%	3



Gambar 3. 2 *Timeline Project*

Berdasarkan gambar 3.1 dan 3.2, berikut ini merupakan penjelasan lebih rinci dari tiap fase yang dilakukan:

3.1 Welcoming

Pada fase pertama, diadakan kegiatan *welcoming* yang menjadi tahap awal dalam proses magang di perusahaan. Kegiatan ini memiliki tujuan utama untuk memperkenalkan lingkungan kerja secara menyeluruh, baik dari segi budaya perusahaan, struktur organisasi, hingga pembagian tanggung jawab di setiap divisi. Selama kegiatan ini berlangsung, pihak perusahaan juga memberikan penjelasan secara detail mengenai aturan dan tata tertib yang berlaku di lingkungan kerja, termasuk etika profesional, jam kerja, serta prosedur komunikasi antar pegawai agar para peserta magang dapat beradaptasi dengan baik. Selain itu, dijelaskan pula mengenai tugas-tugas serta proyek yang akan menjadi fokus selama masa magang berlangsung, sehingga peserta memiliki gambaran awal mengenai target dan ekspektasi yang harus dicapai.

Tidak hanya itu, dalam fase ini perusahaan juga memaparkan rencana pelatihan (*training*) yang akan diberikan kepada para pemagang, beserta mekanisme evaluasi yang nantinya menjadi acuan dalam menentukan peran dan penempatan peserta selama program magang berlangsung. *Training* ini biasanya mencakup pengenalan terhadap alat kerja, sistem perusahaan, hingga metode kerja yang diterapkan agar pemagang siap secara teknis maupun mental dalam menjalankan tugasnya. Sebagai bagian dari proses orientasi, para pemagang juga diberi kesempatan untuk mempresentasikan hasil kerja atau proyek sebelumnya yang pernah mereka kerjakan. Kegiatan ini bertujuan agar pihak perusahaan dapat mengetahui kemampuan, keahlian, serta bidang yang paling dikuasai oleh masing-masing peserta magang. Dengan demikian, perusahaan dapat menyesuaikan pembagian tugas dan memberikan arahan yang lebih tepat agar hasil magang menjadi maksimal serta memberikan manfaat baik bagi peserta maupun perusahaan.

3.2 Training

Fase *training* merupakan tahap yang wajib diikuti oleh seluruh peserta magang sebelum resmi menjalankan kegiatan magang di PT Lumintu Logic. Tahap ini menjadi fondasi penting dalam mempersiapkan peserta agar memiliki kemampuan teknis dan pemahaman yang sesuai dengan kebutuhan proyek di perusahaan. *Training* ini tidak hanya bertujuan untuk memperkenalkan teknologi yang digunakan oleh perusahaan, tetapi juga untuk menilai sejauh mana kompetensi dan kesiapan peserta dalam beradaptasi dengan lingkungan kerja profesional.

Hasil dari fase *training* ini nantinya akan menjadi dasar dalam pengambilan keputusan perusahaan terkait penentuan peran dan penempatan posisi peserta magang selama program berlangsung, disesuaikan dengan bidang yang paling dikuasai dan potensi yang dimiliki.

Selama mengikuti *training*, para peserta magang mendapatkan kesempatan untuk mempelajari dua bidang utama, yaitu *Frontend* dan *Backend*, yang masing-masing memiliki peran penting dalam pengembangan sistem berbasis web. Berikut ini merupakan penjelasan dari masing-masing peran yang ada:

a. *Front End*, tahap awal training untuk peran frontend dilakukan dengan mempelajari dan melakukan latihan yang berfokus pada *framework* Astro JS, sesuai dengan teknologi yang biasa digunakan oleh perusahaan.
b. *Back End*, dilakukan dengan mempelajari *tool Headless Content Management System* (CMS) yaitu Directus. Dimana Directus merupakan teknologi yang nantinya akan digunakan pada saat mengembangkan basis data *website* Radyakartiyasa.

Berdasarkan hasil *training* yang dilaksanakan selama kurang lebih 3 bulan, penulis diberikan *role* tugas sebagai *backend*. Hal ini sesuai dengan pertimbangan perusahaan dan kemampuan penulis dalam menguasai *tools* yang diajarkan selama fase *training*. Beberapa materi *training* yang dipelajari untuk posisi *backend* adalah *Installing Directus, Accessing API, Creating Collections and Relationships, Directus Flows*, dan sebagainya. Kegiatan *training* yang dilakukan meliputi pembelajaran teori dasar dan juga praktik secara langsung Sedangkan beberapa tugas yang nantinya akan dikerjakan oleh *backend* meliputi membuat CMS dan *Application Programming Interface* (API) menggunakan Directus, meluncurkan *website* agar dapat diakses oleh publik dengan menyiapkan Cloudflare untuk akses ke CMS dan *frontend*, melindungi *host* yang dibatasi aksesnya, menyiapkan tahap *preview*, dan meluncurkan ke tahap produksi.

3.3 Exit Interview

Fase ini dilaksanakan pada 19 Desember 2024 dan merupakan tahap lanjutan yang berperan penting dalam menentukan penempatan posisi kerja bagi peserta magang. Pada tahap ini, penulis diminta untuk menampilkan serta menjelaskan hasil dari materi *training* yang telah dipelajari selama masa pelatihan. Kegiatan ini bukan hanya sekadar formalitas, melainkan dimaksudkan untuk menilai sejauh mana pemahaman dan penguasaan peserta magang terhadap materi yang diberikan serta kemampuannya dalam mengimplementasikan ilmu tersebut ke dalam praktik kerja yang nyata.

Pelaksanaan *exit interview* dimulai dengan presentasi individu oleh setiap peserta magang. Dalam sesi tersebut, penulis memaparkan secara rinci materi yang dipahami penulis seperti konsep, tahapan pengerjaan, serta hasil dari latihan yang telah diselesaikan selama *training*. Setelah presentasi, supervisor dan karyawan senior memberikan kesempatan untuk sesi tanya jawab yang berfungsi sebagai evaluasi langsung terhadap tingkat pemahaman, kemampuan pemecahan masalah, dan kejelasan materi.

Sejak awal masa *training*, perusahaan telah menjelaskan adanya dua posisi kerja utama yang menjadi opsi penempatan, yakni *frontend developer* dan *backend developer*. Dan berdasarkan hasil evaluasi selama presentasi dan sesi tanya jawab pada *exit interview*, penulis dinilai memiliki kemampuan yang lebih dominan di bidang *backend*. Dengan pertimbangan tersebut, perusahaan menugaskan penulis untuk menempati posisi *backend developer* agar potensi yang dimiliki dapat dikembangkan secara optimal dan memberikan kontribusi maksimal pada proyek perusahaan di masa mendatang.

3.4 Pengerjaan Proyek Magang

3.4.1 Manajemen Proyek

Berikut merupakan tahapan manajemen proyek yang dilakukan dalam pengembangan basis data *website* Radyakartiyasa:

a. Pendefinisian Proyek

Proyek yang akan dikerjakan adalah mengembangkan basis data *website* Radyakartiyasa milik Keraton Yogyakarta Proyek ini sendiri dikerjakan dengan tim yang berisi 4 orang, yang terdiri dari 2 orang mengerjakan sisi *front end* dan 2 orang lagi pada aspek *backend*. Dikarenakan adanya 2 orang di posisi *backend,* perbedaan tugas yang dikerjakan berada pada beberapa tugas dan koleksi Directus yang dibuat. Masing-masing *backend* memiliki tugas untuk membuat koleksi yang berbeda. Koleksi yang dibuat nantinya akan ditentukan pada saat awal pembagian tugas per *milestone.* Selain koleksi Directus, terdapat pula perbedaan tugas seperti *prepare* Directus *container*, *deploy* Directus, *protect* Directus, dan sebagainya.

Website Radyakartiyasa dikembangkan sebagai bagian dari inisiatif digitalisasi Kraton Yogyakarta untuk menjangkau masyarakat luas secara daring. Sebagai *backend*, penulis bertanggung jawab untuk merancang dan mengembangkan basis data menggunakan Directus sebagai *Headless CMS*. Dalam proyek ini, penulis membuat beberapa *collection* yang nantinya akan digunakan untuk menyimpan berbagai jenis konten yang ditampilkan pada *website* Radyakartiyasa, seperti informasi destinasi, acara, galeri media, dan komponen halaman

lainnya. Setiap *collection* dirancang agar mudah dikelola oleh tim konten Kraton serta mendukung kebutuhan penyesuaian tampilan di sisi *frontend*. Selain itu, penulis juga menetapkan relasi antar koleksi dan mengatur *access policy* agar data dapat diakses dan dimodifikasi sesuai dengan peran masing-masing pengguna.

b. Pelaksanaan Proyek

Pelaksanaan proyek pengembangan basis data *website* Radyakartiyasa dilakukan dengan menerapkan metode *Agile Scrum*. Proses ini diawali dengan konfigurasi pada sistem *Headless CMS* Directus yang digunakan, dilanjutkan dengan pembuatan berbagai koleksi basis data untuk memenuhi kebutuhan konten, dan diakhiri dengan tahap pengujian API untuk memastikan fungsinya sebelum diintegrasikan ke sisi *frontend*. Berikut ini merupakan isi dari file docker-compose.yaml yang disediakan oleh Directus:

```
services:
  database:
    image: postgis/postgis:13-master
    volumes:
      - ./data/database:/var/lib/postgresql/data
    environment:
      POSTGRES_USER: "directus"
      POSTGRES_PASSWORD: "directus"
      POSTGRES_DB: "directus"
    healthcheck:
      test: ["CMD", "pg_isready", "--host=localhost", "--
username=directus"]
      interval: 10s
      timeout: 5s
      retries: 5
      start_interval: 5s
      start_period: 30s
```

```
cache:
  image: redis:6
  healthcheck:
    test: ["CMD-SHELL", "[ $$(redis-cli ping) = 'PONG' ]"]
    interval: 10s
    timeout: 5s
    retries: 5
    start_interval: 5s
    start_period: 30s


directus:
  image: directus/directus:REPLACE_WITH_VERSION
  ports:
    - 8055:8055
  volumes:
    - ./uploads:/directus/uploads
    - ./extensions:/directus/extensions
  depends_on:
    database:
      condition: service_healthy
    cache:
      condition: service_healthy
  environment:
    SECRET: "REPLACE_WITH_YOUR_SECRET"
```

```
    DB_CLIENT: "pg"
    DB_HOST: "database"
    DB_PORT: "5432"
    DB_DATABASE: "directus"
    DB_USER: "directus"
    DB_PASSWORD: "directus"

    CACHE_ENABLED: "true"
    CACHE_AUTO_PURGE: "true"
    CACHE_STORE: "redis"
    REDIS: "redis://cache:6379"

    ADMIN_EMAIL: "REPLACE_WITH_YOUR_EMAIL"
    ADMIN_PASSWORD: "REPLACE_WITH_YOUR_PASSWORD"

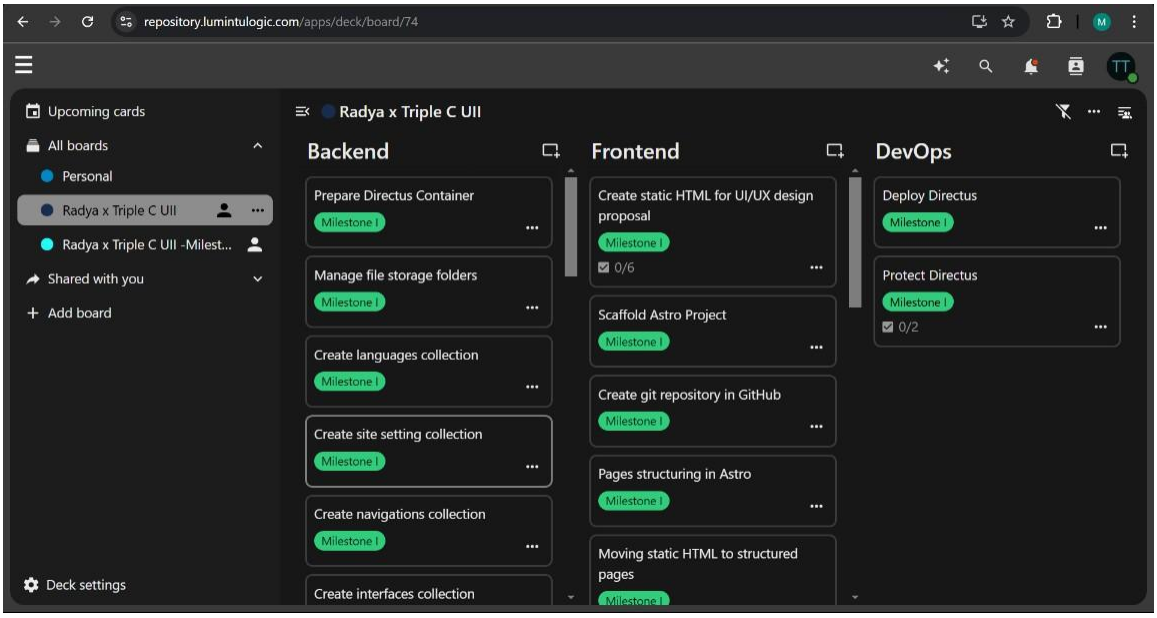
    PUBLIC_URL: "REPLACE_WITH_YOUR_URL"
```

Gambar 3. 3 *Code* dari *File* Docker-Compose.yaml

Seluruh rangkaian kegiatan ini didukung oleh berbagai *tools* pendukung, mulai dari *platform* utama untuk membangun basis data, aplikasi untuk pengujian, hingga alat manajemen proyek internal untuk komunikasi dan koordinasi tugas dengan pihak supervisor. *Tools-tools* yang digunakan adalah sebagai berikut:

1. **Lakon Lumintu**

   Lakon adalah sebuah *tool* milik PT Lumintu Logic Indonesia yang digunakan sebagai media komunikasi untuk para karyawan perusahaan. *Tool* ini juga digunakan selama kegiatan magang untuk melihat daftar tugas yang harus dikerjakan, dan membuat *appointment* dengan supervisor. Pihak perusahaan juga dapat melihat dan memberikan *update* terkait tugas yang dikerjakan.



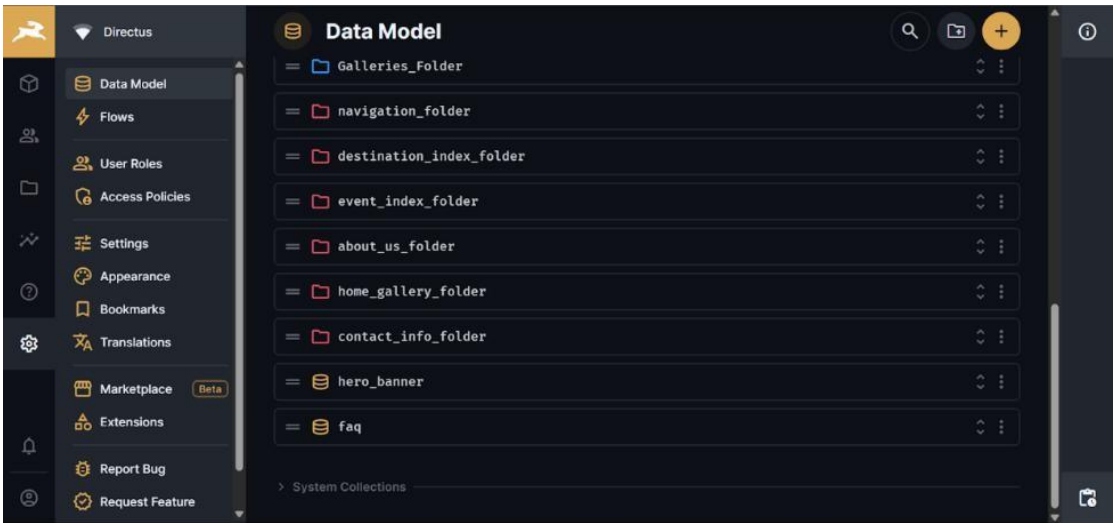
Gambar 3. 4 Tampilan Lakon

Terdapat beberapa fitur yang ada pada Lakon, seperti *deck, talk, appointment, calendar, notes, tasks, tables,* dan sebagainya. Dalam kegiatan magang, Lakon memiliki peran penting sebagai pusat koordinasi dan dokumentasi aktivitas kerja. Dengan menggunakan *tool* ini, peserta magang dapat mengetahui tugas yang harus dikerjakan, memantau progres pekerjaan, serta berkomunikasi langsung dengan supervisor untuk mendapatkan arahan atau umpan balik. Pihak perusahaan juga dapat memantau perkembangan tugas secara *real-time* dan memberikan pembaruan apabila diperlukan. Sebagai contoh, ketika pembagian tugas, supervisor dapat menambahkan detail tugas tersebut pada fitur *deck* beserta dengan tenggat waktu pengerjaannya. Pada fitur *deck* juga dapat memberikan pembaruan progres, melakukan diskusi melalui fitur *talk*, dan menjadwalkan pertemuan untuk evaluasi hasil pekerjaan menggunakan fitur *appointment* yang otomatis tercatat di calendar. Dengan demikian, seluruh proses kerja selama magang dapat berlangsung secara lebih efisien, terarah, dan terdokumentasi dengan baik.

2. Directus

Directus merupakan *Headless content management system* (CMS) yang digunakan untuk pembuatan basis data *website* Radyakartiyasa. Directus digunakan untuk membuat koleksi-koleksi yang nantinya akan digunakan untuk menyimpan kumpulan data konten *website*. Penulis memiliki tanggung jawab untuk membuat dan mengelola berbagai koleksi konten pada *website*, antara lain koleksi *navigation, hero, destination index,* dan

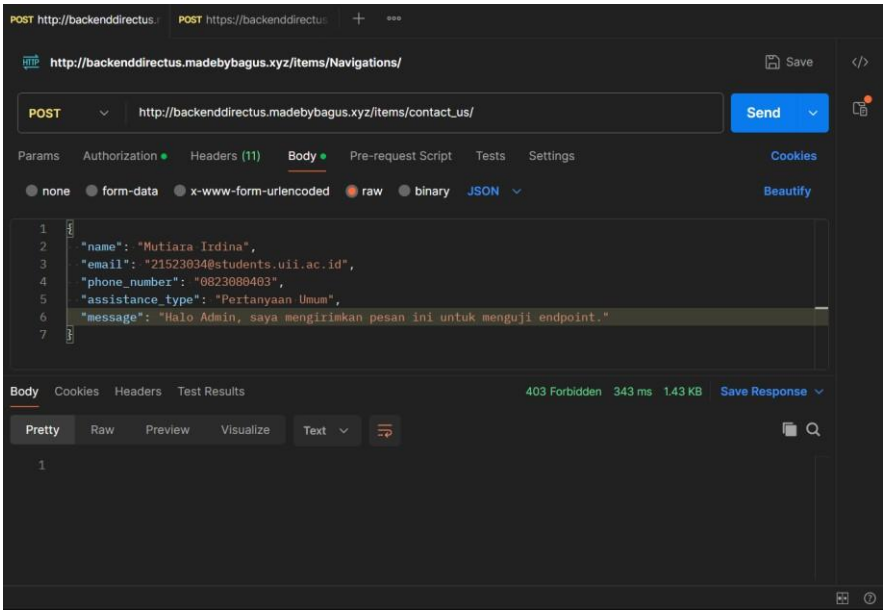
event index. Selain itu, penulis juga bertugas menyiapkan koleksi untuk bagian informasi seperti *about us, Frequently Asked Questions (FAQ), Questions and Answer (QnA)* serta *contact info*, yang berfungsi sebagai media komunikasi dan informasi bagi pengunjung situs. Tidak hanya itu, penulis juga ditugaskan untuk membuat koleksi *home gallery*, yang menampilkan media atau galeri utama di halaman depan *website*, sehingga keseluruhan pengalaman pengguna menjadi lebih menarik dan informatif.



Gambar 3.5 Tampilan Directus

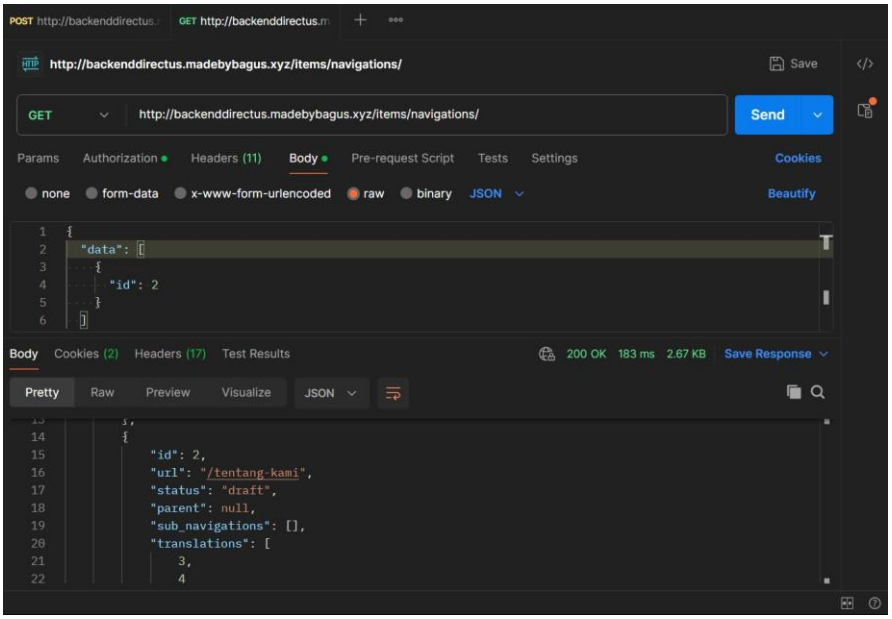
3. Postman

Postman digunakan untuk menguji API yang dihasilkan dari koleksi yang telah dibuat di Directus. Melalui Postman, penulis dapat mengirim permintaan seperti GET, POST, dan DELETE untuk memastikan data dapat diakses, ditambahkan, diubah, atau dihapus dengan benar. Pengujian ini dilakukan guna membantu memastikan API bekerja sesuai kebutuhan sebelum nantinya akan digunakan pada bagian *frontend.*



Gambar 3.6 Tampilan Postman

Untuk melakukan pengujian API, diperlukan *static token* karena Directus memiliki sistem keamanan yang dapat membatasi akses. *Static token* berfungsi sebagai kunci untuk masuk ke API tanpa perlu melakukan *log in* berulang kali, sehingga proses pengujian menjadi lebih praktis. Token tersebut dibuat dan diberikan oleh Directus pada bagian admin, sehingga akses tetap aman dan hanya bisa digunakan sesuai dengan izin yang diberikan. Permintaan atau metode yang sering digunakan adalah GET dan POST, berikut ini merupakan contoh dari penggunaan kedua metode tersebut:



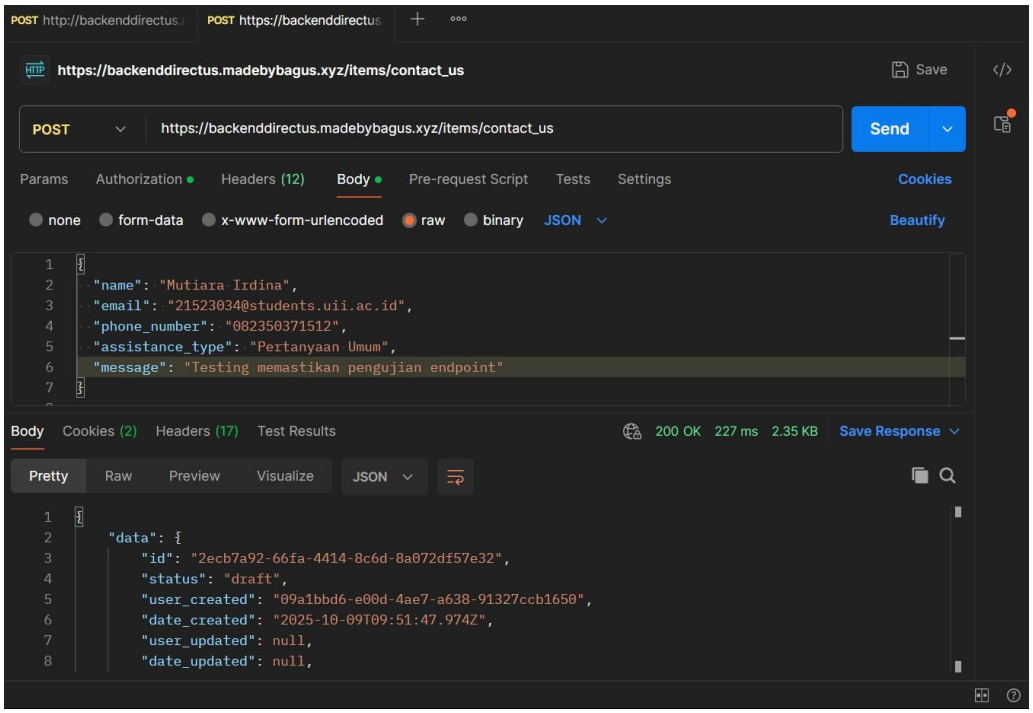
Gambar 3.7 Tampilan Pengujian Metode GET

Pada Gambar 3.6 dilakukan pengujian API pada koleksi *"navigations"* dengan metode GET menggunakan *endpoint* "https://backenddirectus.madebybagus.xyz/items/event_index". Pada pengujian ini, *raw body* JSON yang dikirim adalah sebagai berikut:


```
{
  "data": [
    {
      "id": 2
    }
  ]
}
```


Gambar 3.8 *Raw Body Navigations*

Hasil pengujian pada Gambar 3.6 menunjukkan bahwa sistem mengembalikan status *code* 200 OK dengan *response* berisi detail data *navigations* seperti id, URL, status, parent, sub_navigations, dan translations. Hal ini membuktikan bahwa data berhasil diambil sesuai ID yang diminta pada *request body*, serta struktur *response* yang dihasilkan sudah sesuai dengan rancangan basis data yang dibuat di Directus.



Gambar 3.9 Tampilan Pengujian Metode Post

Pada Gambar 3.8, pengujian API dilakukan pada koleksi “contact_us” dengan metode POST menggunakan endpoint “https://backenddirectus.madebybagus.xyz/items/contact_us”. Data dikirim melalui format raw body berjenis JSON yang berisi informasi nama, email, nomor telepon, *assistance type*, dan pesan. Contoh data yang dikirim adalah sebagai berikut:

```
{
  "name": "Mutiara Irdina",
  "email": "21523034@students.uii.ac.id",
  "phone_number": "0823080403",
  "assistance_type": "Pertanyaan Umum",
  "message": "Halo Admin, saya mengirimkan pesan ini untuk
 menguji endpoint."
}
```

Gambar 3.10 *Raw Body* Untuk Metode POST

Hasil pengujian pada Gambar 3.8 menunjukkan bahwa sistem mengembalikan status *code* 200 OK dan menampilkan data yang sama seperti data yang dikirimkan. Hal ini menandakan bahwa proses penyimpanan data pada basis data berjalan dengan benar dan *endpoint* dapat menerima permintaan POST dengan struktur data yang sesuai.

3.4.2 Hasil Pengembangan Proyek

Berikut merupakan hasil dari pengembangan proyek yang dilakukan pada PT. Lumintu Logic Indonesia:

a. Konfigurasi Directus

Pengembangan basis data pada situs web Radyakartiyasa diawali dengan tahap konfigurasi *Headless Content Management System* (CMS) Directus. Langkah awal dalam proses ini melibatkan penyesuaian *file docker-compose.yaml* yang disediakan oleh Directus. Beberapa bagian dalam *file* tersebut perlu disesuaikan agar memenuhi kebutuhan spesifik proyek, seperti pengaturan *port* dan konfigurasi akun administrator yang dilakukan melalui variabel “ADMIN_EMAIL” dan “ADMIN_PASSWORD”. Untuk menjaga kerahasiaan dan privasi proyek, informasi yang berkaitan dengan kedua variabel tersebut disamarkan dalam dokumentasi ini.

```
version: "3"

services:
  database:
    image: postgis/postgis:13-master
    volumes:
```

```
      - ./data/database:/var/lib/postgresql/data
    environment:
      POSTGRES_USER: "directus"
      POSTGRES_PASSWORD: "directus"
      POSTGRES_DB: "directus"
    healthcheck:
      test: ["CMD", "pg_isready", "--host=localhost", "--
username=directus"]
      interval: 10s
      timeout: 5s
      retries: 5
      start_interval: 5s
      start_period: 30s

  cache:
    image: redis:6
    healthcheck:
      test: ["CMD-SHELL", "[ $$(redis-cli ping) = 'PONG' ]"]
      interval: 10s
      timeout: 5s
      retries: 5
      start_interval: 5s
      start_period: 30s

  directus:
    image: directus/directus:11.3.5
    ports:
      - 8123:8055
    volumes:
      - ./uploads:/directus/uploads
      - ./extensions:/directus/extensions
    depends_on:
      database:
        condition: service_healthy
      cache:
        condition: service_healthy
    environment:
      SECRET: "replace-with-secure-random-value"

      DB_CLIENT: "pg"
      DB_HOST: "database"
      DB_PORT: "5432"
      DB_DATABASE: "directus"
      DB_USER: "directus"
      DB_PASSWORD: "directus"

      CACHE_ENABLED: "true"
      CACHE_AUTO_PURGE: "true"
      CACHE_STORE: "redis"
      REDIS: "redis://cache:6379"

      ADMIN_EMAIL: "xxxxxxxxxxxxx@example.com"
      ADMIN_PASSWORD: "xxxxxxxxxxxxxxxxxxxx"
```

```
# PUBLIC_URL: "https://directus.example.com"
```

Gambar 3.11 Kode Konfigurasi Directus dalam Proyek

Berikut penjelasan lebih lengkap terkait konfigurasi yang terdapat pada gambar di atas:

1.) File ini menggunakan version: "3" yang merupakan versi dari Docker Compose.

2.) *Service database* memakai image postgis/postgis:13-master, yaitu PostgreSQL yang telah disertai fitur PostGIS. Volume ./data/database:/var/lib/postgresql/data dipakai agar data *database* tidak hilang meskipun *container* dimatikan. Variabel POSTGRES_USER, POSTGRES_PASSWORD, dan POSTGRES_DB berfungsi untuk membuat akun dan nama *database* secara otomatis. *Healthcheck* memastikan *database* benar-benar siap sebelum digunakan.

3.) *Service cache* menggunakan Redis versi 6 sebagai sistem penyimpanan *cache*. *Healthcheck* memeriksa apakah Redis sudah berjalan dengan benar melalui perintah redis-cli ping.

4.) *Service* directus menggunakan *image* directus/directus:11.3.5 dan membuka aplikasi Directus melalui port 8123. Volume uploads dan extensions digunakan agar file dan ekstensi tetap tersimpan di luar *container*. Bagian depends_on memastikan Directus hanya dijalankan setelah *database* dan Redis dinyatakan sehat.

5.) Directus membutuhkan berbagai pengaturan penting, seperti SECRET yang menjadi kunci keamanan aplikasi. Pengaturan database seperti *host*, *port*, nama *database*, *username*, dan *password* dilakukan melalui variabel DB_CLIENT, DB_HOST, DB_PORT, DB_DATABASE, DB_USER, dan DB_PASSWORD.

6.) Penggunaan Redis sebagai *cache* diaktifkan melalui pengaturan CACHE_ENABLED, CACHE_AUTO_PURGE, CACHE_STORE, dan REDIS, sehingga Directus dapat berjalan lebih cepat.

7.) Aplikasi secara otomatis membuat akun administrator pertama menggunakan ADMIN_EMAIL dan ADMIN_PASSWORD.

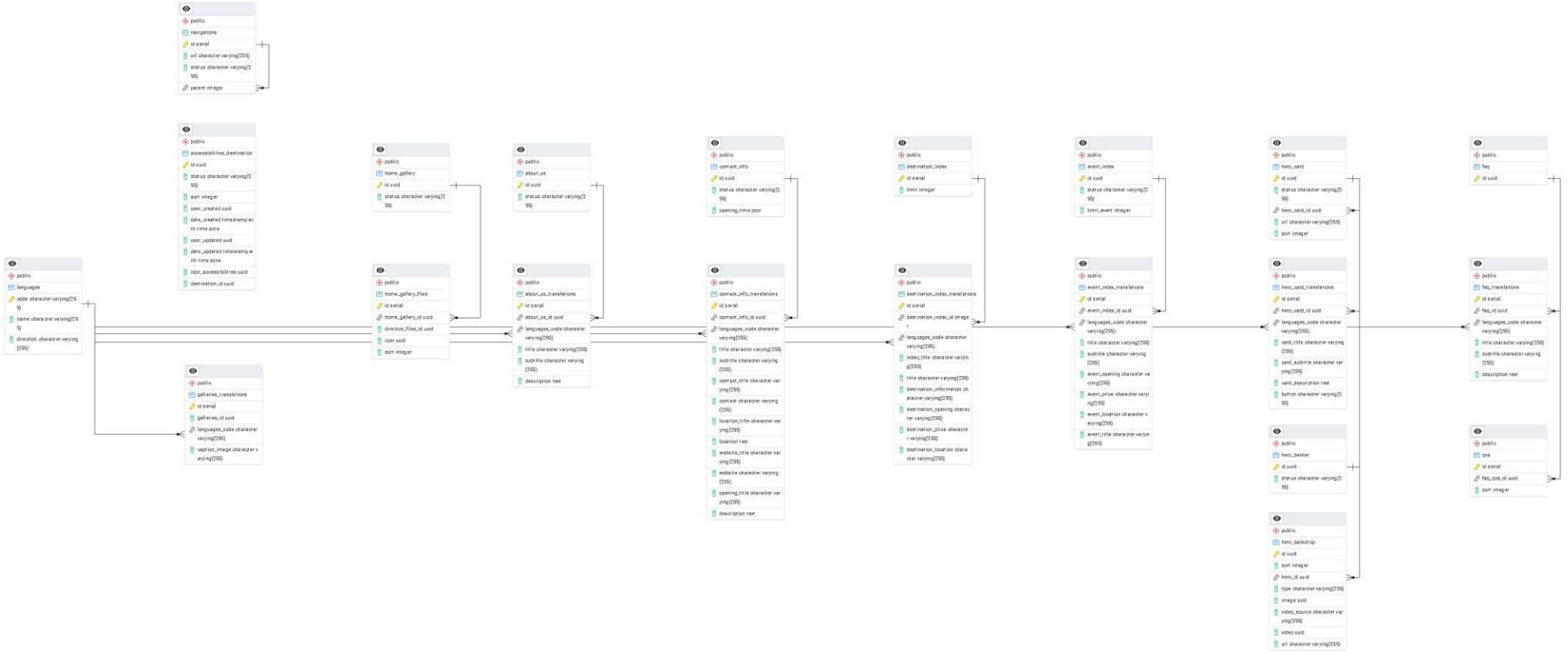
8.) Bagian PUBLIC_URL bersifat opsional dan digunakan jika aplikasi dijalankan di server publik dengan domain tertentu.

Pada fungsi “ports: 8123:8055”, *port* 8055 merupakan *port* bawaan yang digunakan oleh aplikasi Directus di dalam *container*. Konfigurasi "ports: 8123:8055" berarti bahwa ketika pengguna mengakses Directus melalui *port* 8123 pada *host* (contohnya http://localhost:8123), permintaan tersebut akan diteruskan ke *port* 8055 yang berjalan di dalam *container*. Fungsi ADMIN_EMAIL: "**********@example.com" digunakan untuk menetapkan alamat email yang akan digunakan oleh akun administrator (Super Admin). Sementara itu, fungsi “ADMIN_PASSWORD: "**********"” berperan dalam menentukan kata sandi yang akan digunakan untuk mengakses Directus melalui alamat email yang telah ditetapkan sebelumnya pada variabel ADMIN_EMAIL.

b. Pembuatan Koleksi

Tahap selanjutnya adalah pembuatan koleksi (*collections*) pada Directus, yang disesuaikan dengan struktur dan kebutuhan konten dari *website* Radyakartiyasa. Koleksi-koleksi ini berfungsi sebagai wadah untuk mengelola berbagai jenis data yang akan ditampilkan pada situs web. Beberapa koleksi yang telah dirancang dan implementasikan antara lain *Navigation, Destination Index, Event Index, About Us, Home Gallery, Contact Info, Hero Banner, dan Frequently Asked Questions (FAQ).* Sebagian besar koleksi yang dibuat juga memiliki relasi dengan koleksi *translations* yang bertujuan untuk membuat *website* menjadi multilingual.

Directus menyediakan API otomatis untuk setiap koleksi yang dibuat, sehingga aplikasi dapat mengambil dan mengelola data tanpa konfigurasi tambahan API tersebut mengikuti struktur yang konsisten sehingga memudahkan integrasi dengan sistem atau aplikasi lain. Pada koleksi bertipe biasa, *endpoint* yang tersedia memungkinkan pengelolaan banyak data dalam satu koleksi. Sementara itu, koleksi bertipe *singleton* memiliki *endpoint* khusus yang hanya digunakan untuk mengambil dan memperbaharui satu data tetap. Dengan demikian, seluruh koleksi yang dibuat pada Directus telah disediakan API bawaan yang siap digunakan dalam proses pengembangan aplikasi.



Gambar 3. 12 Struktur Koleksi Directus

Gambar ini memperlihatkan keseluruhan struktur koleksi pada Directus yang telah dirancang untuk mengelola berbagai jenis konten dalam sistem. Diagram tersebut menunjukkan hubungan antar koleksi serta alur pengorganisasian data yang mendukung fitur-fitur utama *website*, termasuk pengelolaan halaman, media, dan konten multilingual. Berikut ini merupakan penjelasan mengenai koleksi-koleksi yang dibuat per sprint:

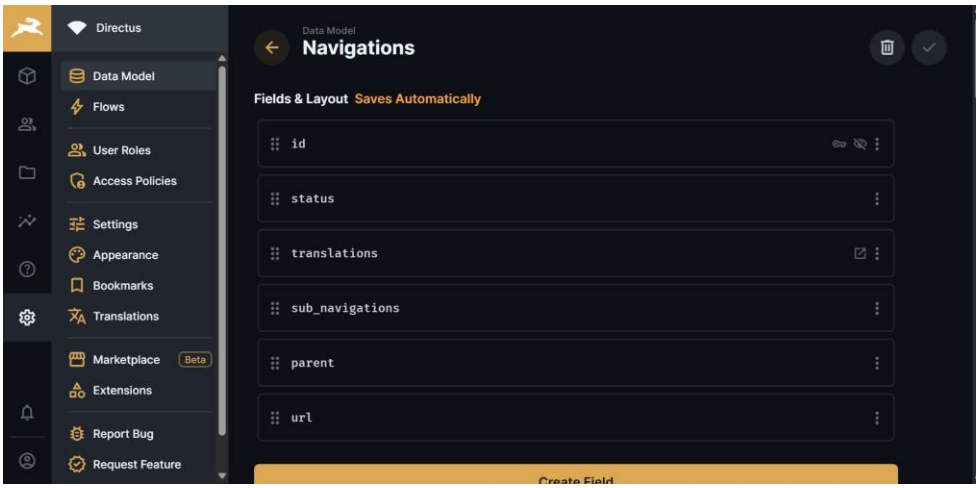
Tabel 3. 1 Daftar *Sprint 1*

No.	Tugas	Tanggal	Feedback Perusahaan	Perbaikan
1.	*Create Navigation Collection*	02/01/25 – 07/02/25	- Penggunaan *interface repeater* pada bagian submenu masih kurang tepat. - Tambahkan *field parent* untuk *foreign key* menggunakan relasi *many to one*.	- *Interface repeater* telah diubah menjadi *treeview*. - Telah ditambahkan *field parent*, dan relasi *many to one*.
2.	*Create Destination Index Collection*	06/01/25 – 07/02/25	- Bersifat *singleton* namun tetap memiliki relasi dengan koleksi *translation*. - Yang dapat ditampilkan hanya yang ada di index, dan dapat di *sort.* - Hapus *fields maximum value*.	- Koleksi telah diubah menjadi bersifat *singleton* dan berelasi dengan *translation*. - Menambahkan *Sort*. - *Fields maxium value* telah dihapus.
3.	*Create Event Index Collection*	07/01/25 – 07/02/25	- Bersifat *singleton* namun tetap memiliki relasi dengan koleksi *translation*. - Yang dapat ditampilkan hanya yang ada di *index*, dan dapat di *sort*. - Hapus *fields maximum value.*	- Koleksi telah diubah menjadi bersifat *singleton* dan berelasi dengan *translation*. - Menambahkan *sort*. - *Fields maxium value* telah dihapus.

Berikut ini penjelasan lebih rinci mengenai daftar pekerjaan yang ada pada *sprint* pertama:

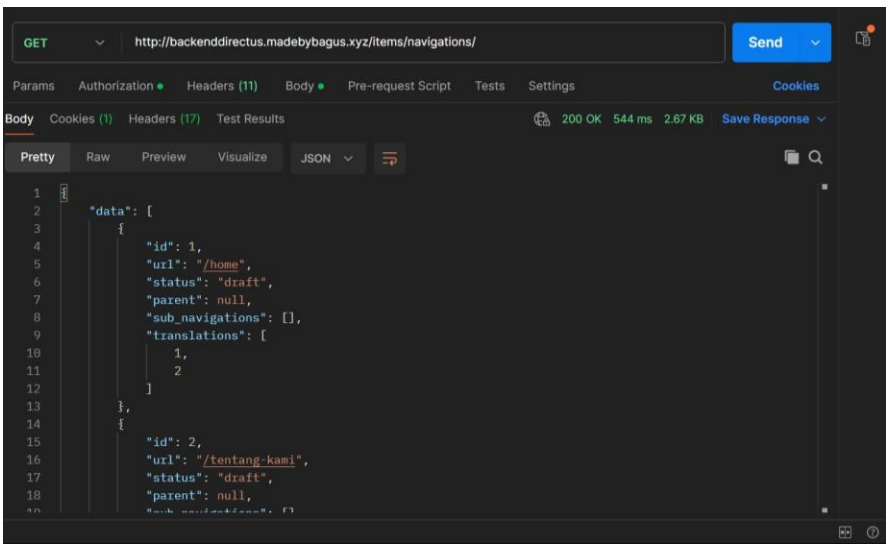
1. Navigation

Koleksi *Navigation* berfungsi sebagai koleksi yang digunakan untuk menampilkan menu dan submenu pada *homepage website* Radyakartiyasa.



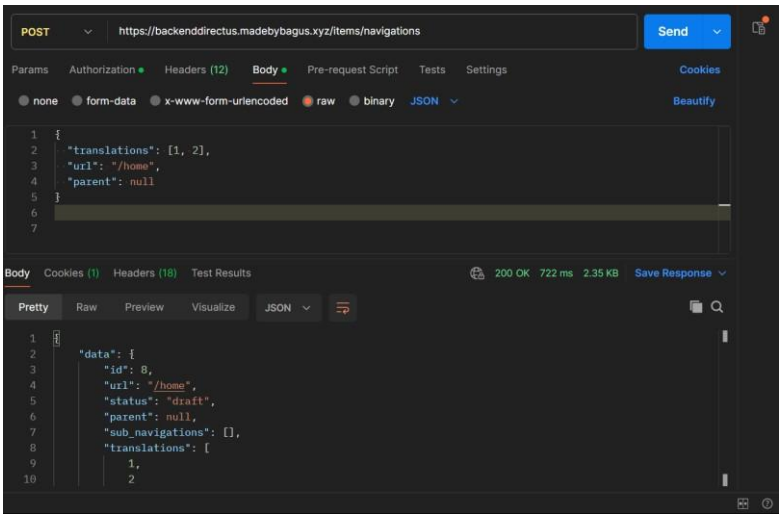
Gambar 3.13 Tampilan Koleksi Navigation

Isi dari koleksi *Navigation* meliputi beberapa *field* seperti *translations, sub navigation, parent,* dan *Uniform Resource Locator* (*URL*). Field *translations* memiliki relasi dengan koleksi *navigations translations*. Sedangkan koleksi *navigations translations* berisi *field name* untuk memberi nama pada *sub navigations*. *Fields sub_navigation* untuk sub menu dibuat menggunakan *interface treeview* yang memiliki relasi dengan koleksi navigation itu sendiri. *Field parent* berperan sebagai *foreign key* dari *field sub_navigation.*



Gambar 3. 14 Pengujian API Koleksi *Navigation (*GET*)*

Gambar di atas menunjukkan hasil pengujian API untuk koleksi *Navigation* menggunakan metode GET. Pengujian ini dilakukan untuk memastikan bahwa data pada koleksi dapat diambil dengan benar melalui *endpoint* yang tersedia. Respons yang ditampilkan memperlihatkan struktur data yang sesuai dengan konfigurasi *fields* pada koleksi *Navigation*.

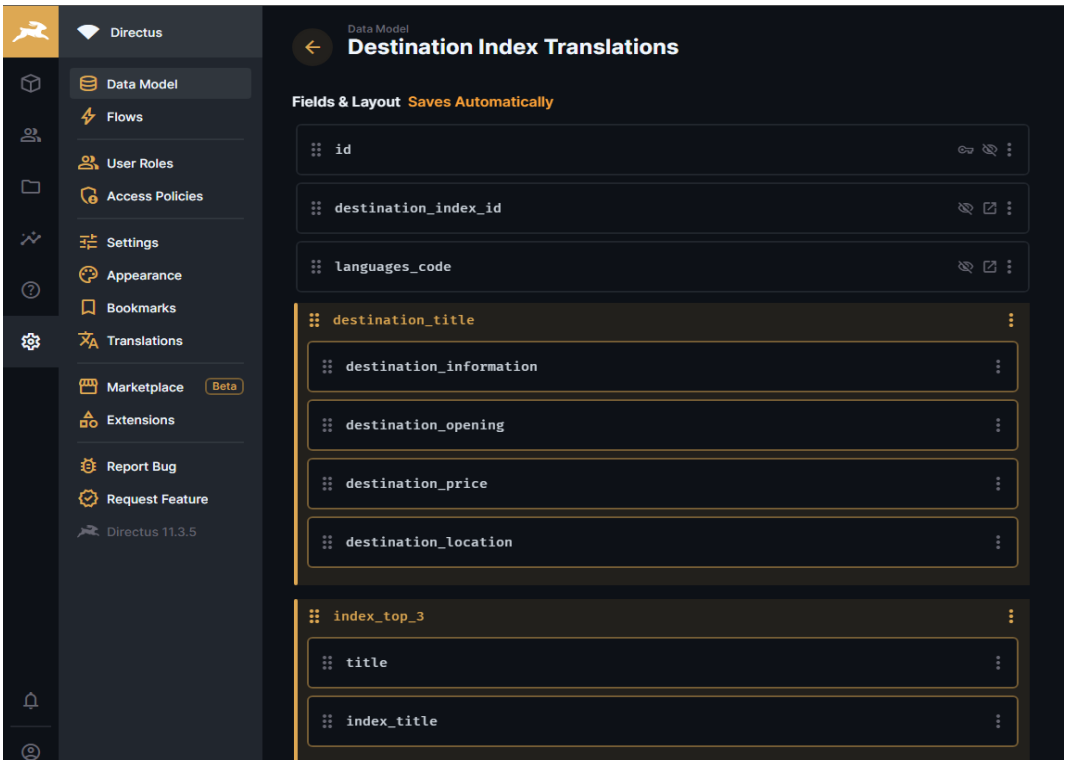


Gambar 3. 15 Pengujian API Koleksi *Navigation* (POST)

Pengujian *endpoint* POST pada koleksi *navigation* berhasil dilakukan dan menghasilkan respons 200 OK, yang menunjukkan bahwa data baru dapat tersimpan dengan baik. *Field* yang dikirim melalui *body*, seperti *translations*, url, dan *parent*, diterima sesuai permintaan. Directus juga otomatis mengisi *field* status dengan nilai default "*draft*".

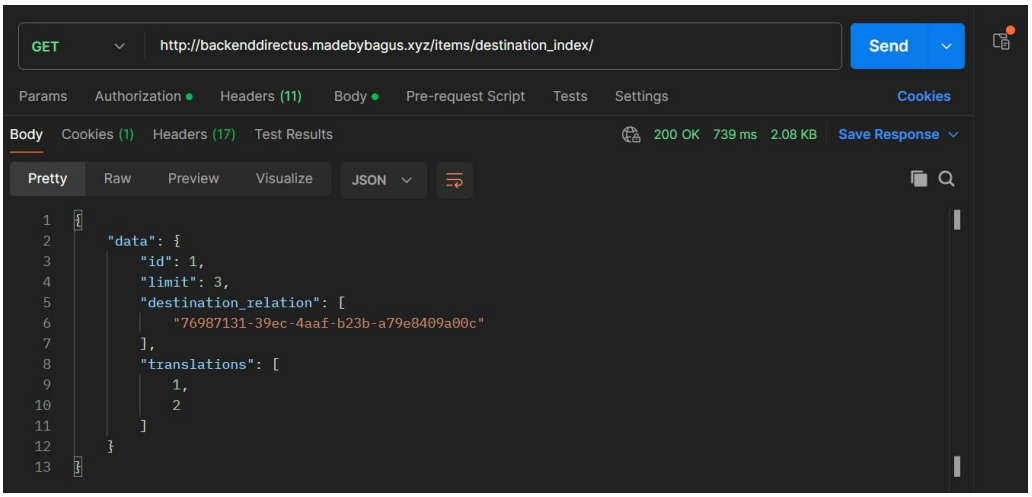
2. Destination Index

Koleksi ini berfungsi untuk membuat daftar beberapa destinasi yang ingin ditampilkan pada halaman utama *website*. Destinasi yang ditampilkan merupakan data yang diambil dari koleksi *destination*. Pada koleksi ini, admin dapat mengatur beberapa destinasi prioritas untuk ditampilkan kepada *user* saat mengakses *website* Radyakartiyasa.



Gambar 3.16 Tampilan Koleksi *Destination Index*

Dalam koleksi ini, terdapat berbagai *fields* yang memiliki fungsinya masing-masing. *Field group destination title* berisi *fields* bertujuan untuk menyimpan informasi dari destinasi-destinasi yang ada. Sedangkan *fields group index* top 3 berfungsi untuk mengatur tiga destinasi yang nantinya akan ditampilkan pada halaman utama situs web. Destinasi yang ingin ditampilan sendiri ditentukan oleh pihak administrator dengan menggunakan *sort* pada koleksi *index*.

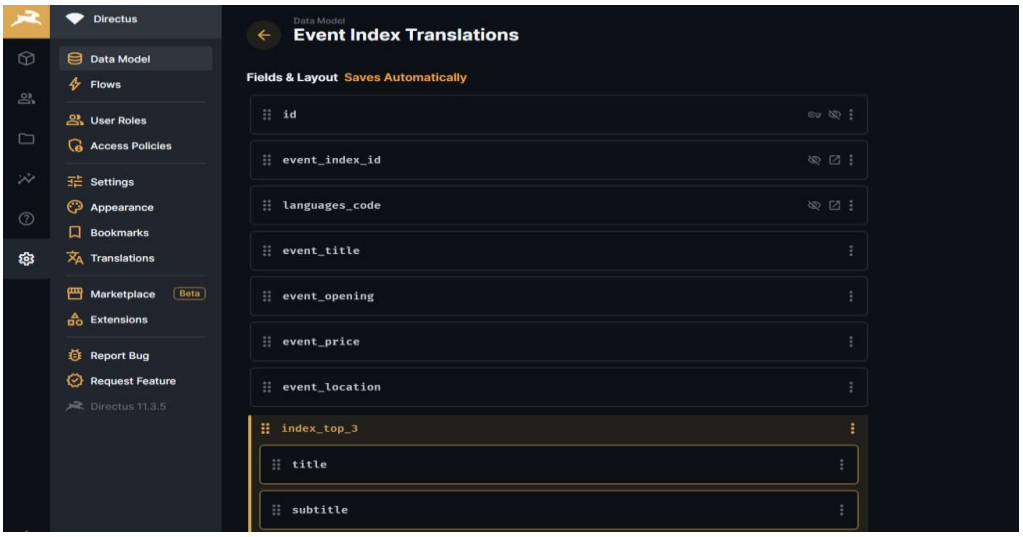


Gambar 3. 17 Pengujian API Koleksi *Destination Index*

Gambar ini menunjukkan hasil respons dari pengujian API GET pada koleksi *destination index*. Informasi destinasi yang muncul mengindikasikan bahwa struktur data telah tersimpan dengan benar dan dapat dipanggil melalui *endpoint* tanpa kendala. Pengujian ini memastikan bahwa data *destination index* dapat diintegrasikan secara lancar ke dalam halaman terkait pada *website*.

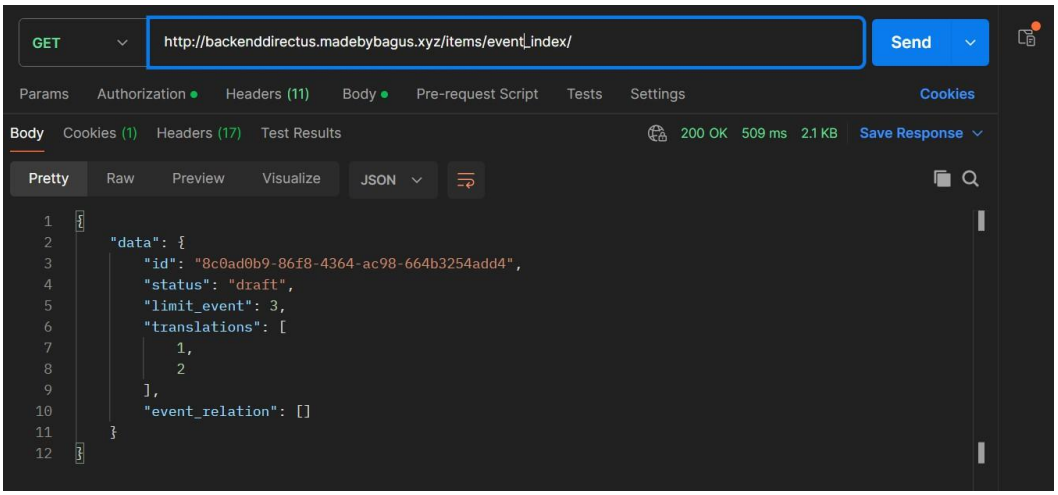
3. Event Index

Sama seperti koleksi *Destination Index*, koleksi ini berfungsi untuk membuat daftar beberapa *event* yang ingin ditampilkan pada halaman utama *website*. Event yang ditampilkan merupakan data yang diambil dari koleksi event. Pada koleksi ini, admin dapat mengatur beberapa *event* prioritas untuk ditampilkan kepada *user* saat mengakses *website* Radyakartiyasa.



Gambar 3.18 Tampilan Koleksi *Event Index*

Pada koleksi ini juga terdapat *group index* top 3 untuk mengatur tiga *event* yang dapat dipilih admin untuk ditampilkan di halaman utama web. Selain itu, juga terdapat *fields* lain yang berfungsi untuk menyimpan data informasi *event* seperti *title*, *opening*, *price*, dan *location*.



Gambar 3. 19 Pengujian API Koleksi *Event Index*

Gambar ini menampilkan hasil pengujian API dengan metode GET pada koleksi *Event Index*. Respon yang dihasilkan menunjukkan bahwa data acara dapat diakses dan ditampilkan dengan benar oleh sistem. Pengujian ini membuktikan bahwa koleksi tersebut dapat digunakan sebagai sumber data pada proses pengembangan aplikasi.

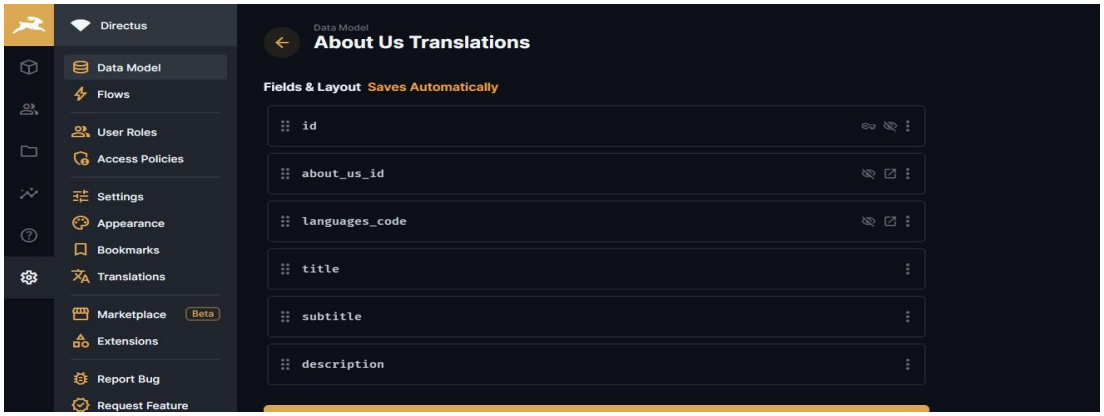
Tabel 3. 2 Daftar *Sprint 2*

No.	Tugas	Tanggal	Feedback Perusahaan	Perbaikan
1.	*Create About Us Collection*	13/01/25 – 17/01/25	Bersifat *singleton*.	Koleksi telah diubah menjadi bersifat *singleton.*
2.	*Create Contact Info Collection*	13/01/25 – 17/01/25	Bersifat *singleton*.	Koleksi telah diganti menjadi bersifat *singleton*.
3.	*Create Home Gallery Collection*	13/01/25 – 27/01/25	- Tag kecil dimasukkan dan multilingual. - Memiliki relasi *many to many.* - *Junction* antar *gallery* (*home_gallery_files*)	- Tak kecil multilingual. - Telah dibuat *relasi many to many.* - Telah dibuat koleksi home_gallery_files sebagai *junction*.

Berikut ini penjelasan lebih rinci mengenai daftar pekerjaan yang ada pada *sprint* kedua:

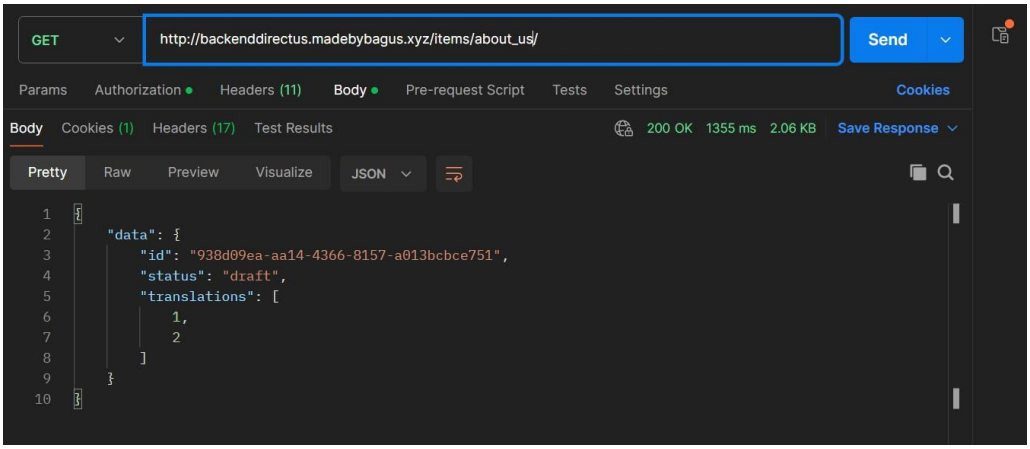
1. About Us

Koleksi *About Us* berisi deskripsi singkat mengenai keraton Yogyakarta, yang mencakup informasi dasar seperti sejarah dan budaya, serta kata sambutan untuk pengunjung *website*. Konten ini disusun secara ringkas namun informatif untuk memberikan gambaran umum kepada pengunjung mengenai identitas kraton.



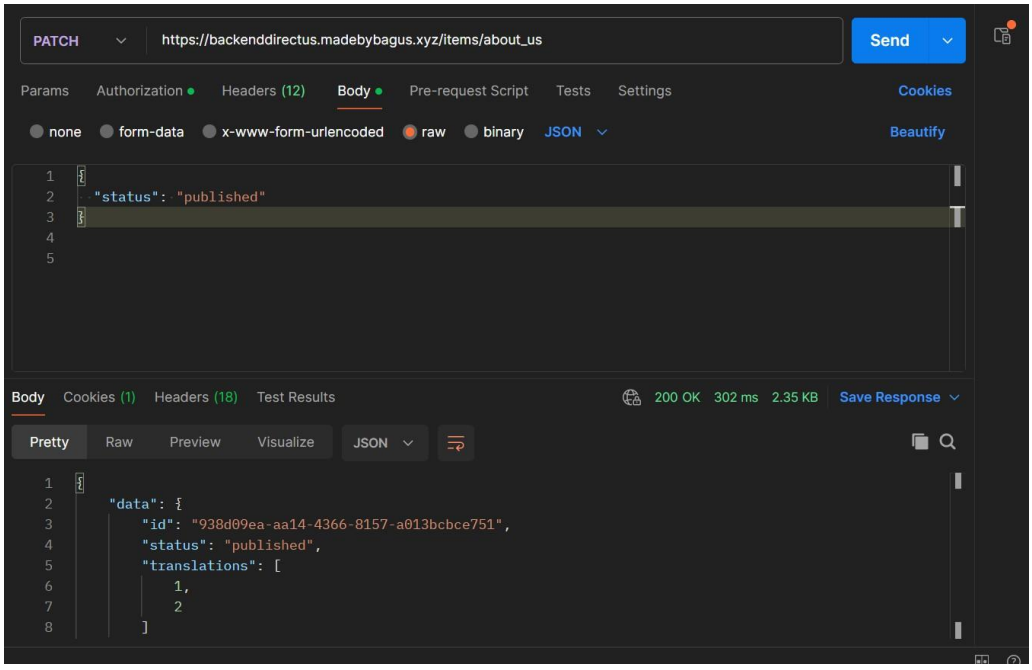
Gambar 3.20 Tampilan Koleksi *About Us*

Fields yang ada di dalam koleksi ini meliputi *title* untuk menyimpan judul halaman, *subtitle* untuk sub judul, dan *description* untuk deskripsi singkat untuk setiap judul. Koleksi ini bersifat *singleton* dimana koleksi hanya berisi satu data.



Gambar 3. 21 Pengujian API Koleksi *About Us* (GET)

Gambar ini menampilkan hasil pengujian API metode GET pada koleksi *about us*. Konten yang ditampilkan dalam respons menunjukkan bahwa data deskripsi dan informasi profil telah tersimpan dengan baik serta dapat diambil oleh sistem. Pengujian tersebut memastikan bahwa bagian *about us* dapat menampilkan informasi secara otomatis sesuai dengan data yang ada di *backend*.



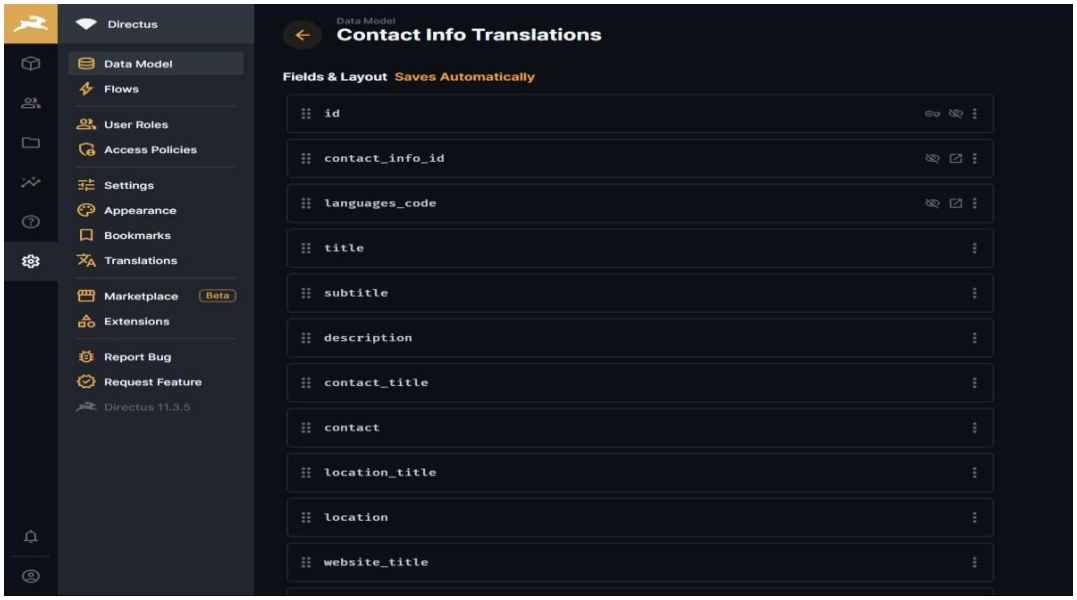
Gambar 3. 22 Pengujian API Koleksi *About Us* (PATCH)

Pengujian metode PATCH pada koleksi *about us* dilakukan untuk memastikan bahwa konten *about us* dapat diperbarui melalui API. Pengujian dilakukan dengan

mengirimkan body JSON berisi perubahan pada *field* status, dan server merespons dengan 200 OK, yang menunjukkan bahwa pembaruan berhasil diterapkan.

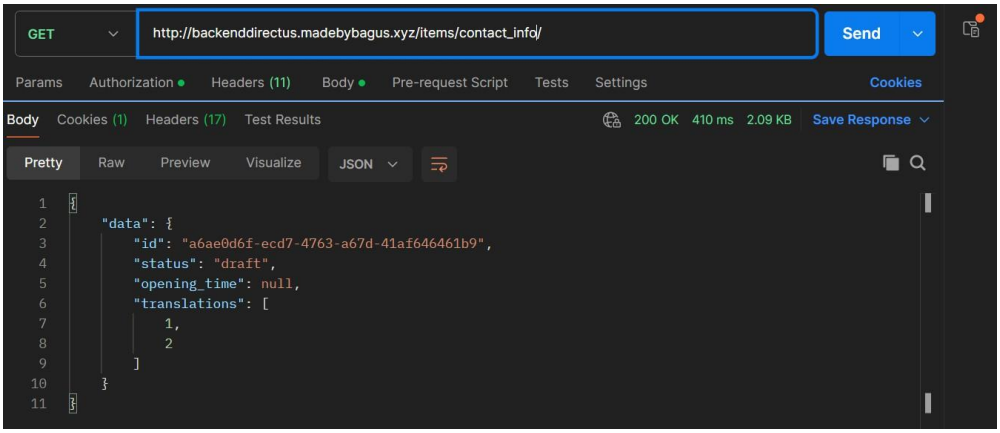
2. Contact Info

Koleksi ini berisi informasi kontak resmi Kraton Yogyakarta, seperti alamat, nomor telepon, dan alamat *website*. Tujuannya adalah untuk memudahkan pengunjung *website* dalam mengakses informasi komunikasi yang relevan dan terpercaya. Dengan adanya koleksi ini, pengguna dapat menghubungi pihak Kraton Yogyakarta secara langsung apabila membutuhkan informasi lebih lanjut atau ingin menjalin kerja sama.



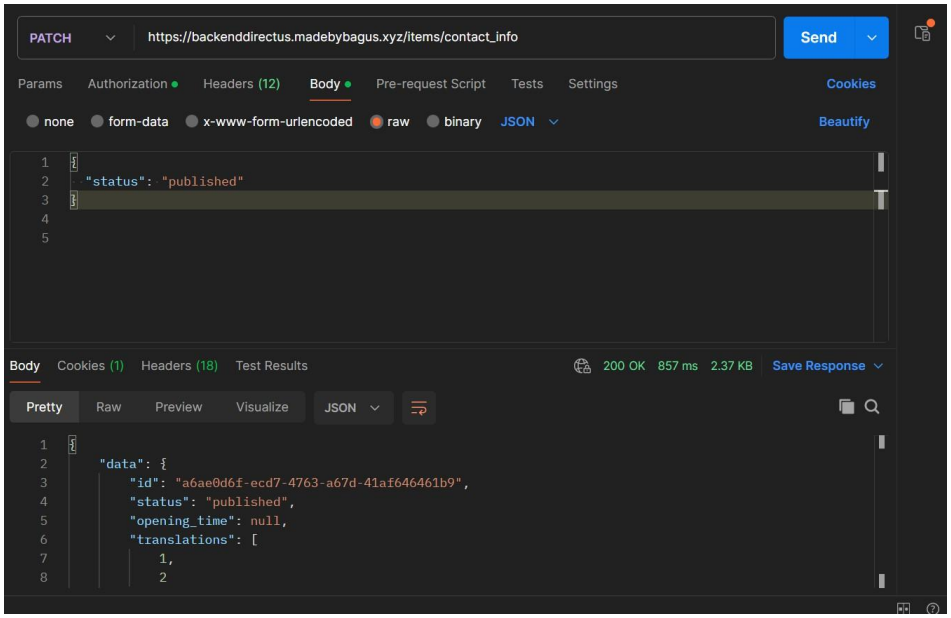
Gambar 3.23 Tampilan Koleksi *Contact Info*

Koleksi *Contact Info* berisi berbagai *fields* yang berfungsi untuk menyimpan data informasi kontak seperti nomor telepon yang dapat dihubungi pengunjung *website*, lokasi, *link website* terkait, dan jam operasional admin.



Gambar 3. 24 Pengujian API Koleksi *Contact Info (*GET*)*

Gambar tersebut menampilkan hasil pemanggilan API GET untuk koleksi *Contact Info*. Data informasi kontak berhasil dipanggil oleh sistem, menunjukkan bahwa seluruh *field* dapat diakses tanpa kendala. Pengujian ini membuktikan bahwa koleksi ini siap mendukung kebutuhan penyediaan informasi bagi pengguna.

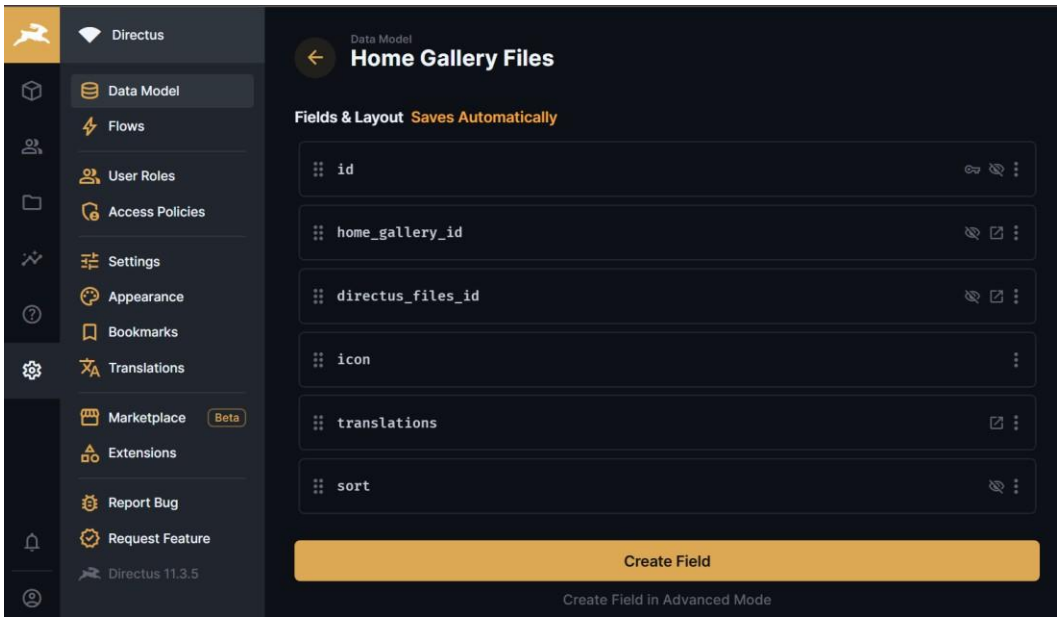


Gambar 3. 25 Pengujian API Koleksi *Contact Info* (PATCH)

Pengujian pembaruan data (*update*) pada koleksi *contact info* dilakukan menggunakan metode PATCH untuk mengganti nilai pada *field* status. Pengujian ini menghasilkan respons 200 OK, yang menunjukkan bahwa perubahan data berhasil diproses oleh server.

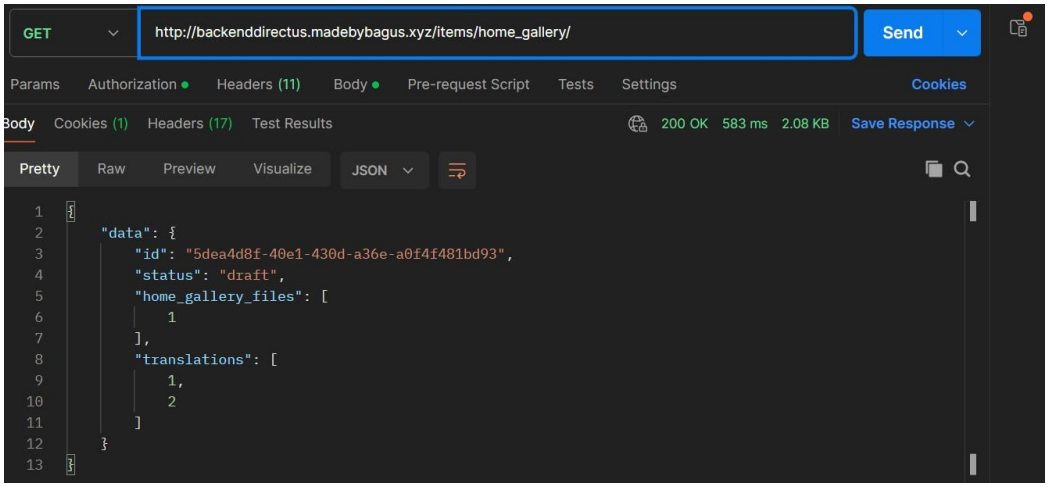
3. Home Gallery

Koleksi *home gallery* berfungsi untuk menampilkan gambar-gambar yang sebelumnya sudah pernah diunggah di koleksi lain. Sehingga *home gallery* dapat mengambil semua data yang bersifat media di koleksi lain.



Gambar 3.26 Tampilan Koleksi *Home Gallery*

Home gallery memiliki koneksi dengan koleksi *home gallery files*. Koleksi ini memungkinkan administrator atau pengelola konten untuk dengan mudah menambahkan, menghapus, maupun mengatur urutan gambar tanpa harus mengubah struktur kode secara manual. Selain itu, dengan adanya koneksi antara *Home Gallery* dan *Home Gallery Files*, setiap perubahan yang dilakukan pada data sumber akan otomatis terintegrasi dan diperbarui pada tampilan galeri di *website*.



Gambar 3. 27 Pengujian API Koleksi *Home Gallery*

Gambar ini memperlihatkan respons API GET dari koleksi *Home Gallery*. Data yang ditampilkan mengonfirmasi bahwa elemen-elemen galeri dapat diakses dan ditampilkan dengan benar oleh sistem. Hasil tersebut menunjukkan bahwa koleksi ini berfungsi sebagaimana mestinya untuk kebutuhan tampilan media.

Tabel 3. 3 Daftar *Sprint 3*

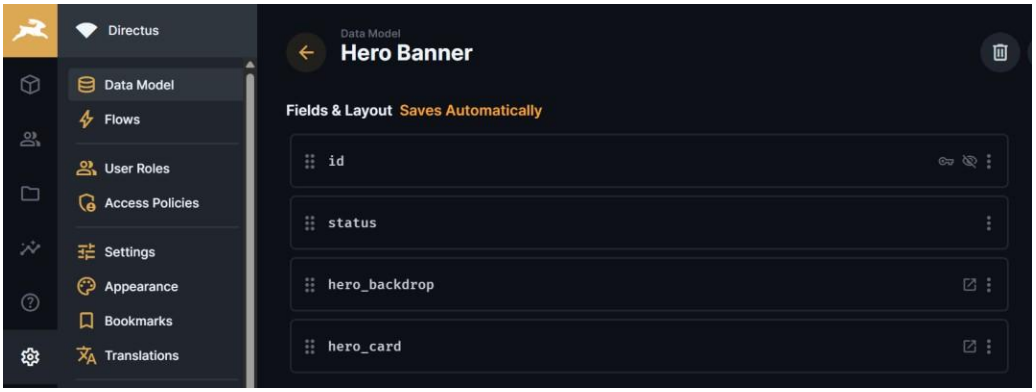
No.	Tugas	Tanggal	Feedback Perusahaan	Perbaikan
1.	*Create Hero Backdrop Collection*	10/02/25 – 23/02/25	- Relasi antar *card* dan *backdrop* dihapus. - Buat koleksi utama yaitu *Hero Banner* yang bersifat *singleton.* - Untuk memilih *type backdropnya* digunakan *interface radio button.* - Terdapat dua *fields* (untuk menyimpan gambar dan video). - Kondisi pada *field image* jika *type* tidak *equal* dengan *image* maka akan *hidden*.	- Relasi *card* dan *backdrop* dihapus. - Telah dibuat koleksi utama yaitu *hero banner.* - *Fields image* dan video telah dibuat sesuai dengan permintaan supervisor.
2.	*Create Hero Card Collection*	10/02/25 – 23/02/25	Menggunakan *sort*.	Telah ditambahkan *sort.*

3.	*Create FAQ Collection*	17/02/25 – 23/02/25	- Bersifat *singleton*. - Memiliki relasi *one to many* dengan koleksi QnA.	- Sifat koleksi telah diubah menjadi *singleton*. - Relasi *one to many* dan koleksi QnA telah ditambahkan.
4.	*Create QnA Collection*	17/02/25 – 23/02/25	- Berisi *fields* relasi dengan koleksi translation. - Pada koleksi QnA *translation* berisi pertanyaan beserta dengan jawabannya.	Koleksi sudah diisi dengan data pertanyaan dan jawaban yang multilingual.

Berikut ini penjelasan lebih rinci mengenai daftar pekerjaan yang ada pada *sprint* ketiga:

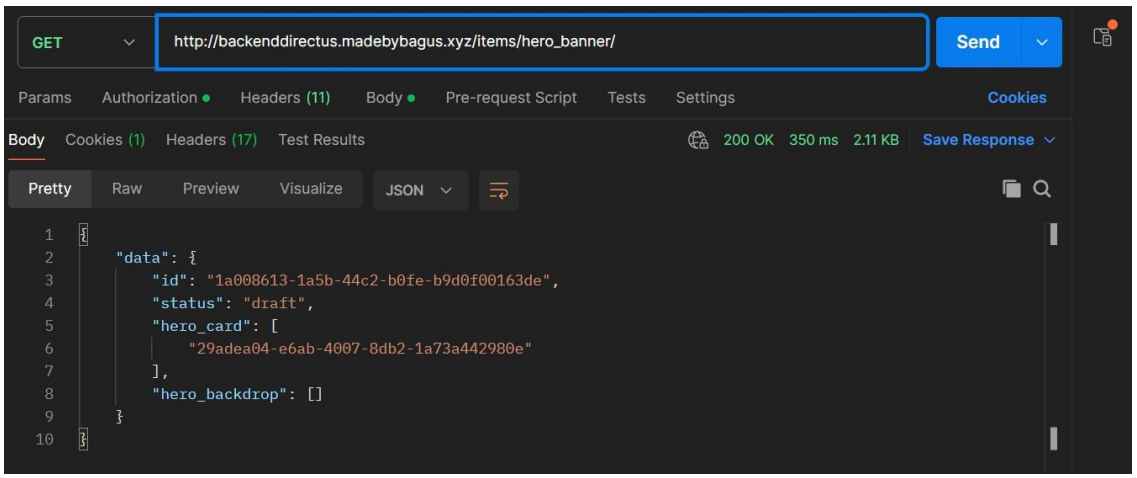
1. Hero Banner

Hero pada *website* Radyakartiyasa terbagi menjadi dua bagian yaitu *backdrop* dan *card*. Sehingga koleksi *hero banner* ini berperan sebagai penggabung dan akan memiliki relasi *one to many* dengan kedua koleksi lainnya. Koleksi *backdrop* dan *card* sendiri tidak memiliki ketergantungan satu sama lain.



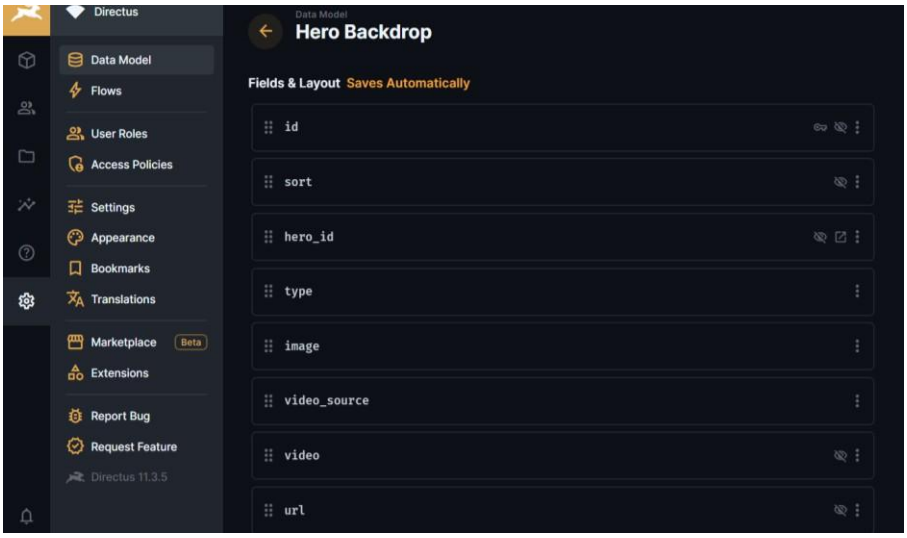
Gambar 3.28 Tampilan Koleksi *Hero Banner*

Gambar di atas menunjukkan struktur data koleksi *Hero Banner* yang telah dibuat di dalam Directus. *Platform* ini kemudian secara otomatis menyediakan *endpoint* REST API untuk koleksi tersebut. Gambar 3.26 menampilkan hasil pengujian *endpoint* API itu, yang mengembalikan data untuk digunakan oleh aplikasi *frontend*.



Gambar 3. 29 Pengujian API Koleksi *Hero Banner*

Koleksi *backdrop* digunakan sebagai tempat untuk menyimpan data background dari *card*. Koleksi ini berisikan media, yang berupa gambar atau video, dan bisa berasal dari internal maupun eksternal. Jika video yang diunggah berasal dari eksternal, maka pengunggah dapat memasukan *link URL (Uniform Resource Locatorsurl)*-nya di *field* yang sudah disediakan. Namun jika gambar atau video yang diunggah berasal dari internal, maka pengunggah bisa langsung mengunggah media yang ingin dimasukkan melalui *field* yang sudah disediakan. Media yang diunggah juga dapat berjumlah lebih dari satu.



Gambar 3.30 Tampilan Koleksi *Hero Backdrop*

Koleksi *card* ini berisi judul dan arahan jika ingin mengetahui lebih lanjut mengenai destinasi yang ada pada judul. Pengunggah dapat memasukkan beberapa data yang dapat

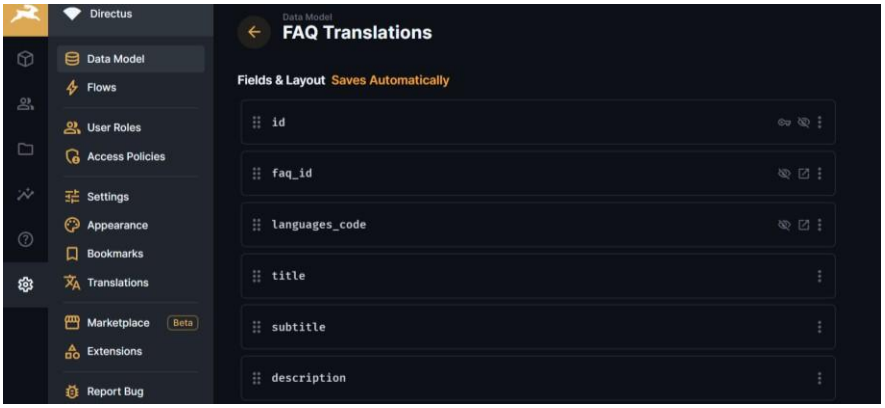
ditampilkan pada halaman utama *website*, dan nantinya data ini akan ditampilkan secara bergantian di halaman utama *website*.



Gambar 3.31 Tampilan Koleksi *Hero Card*

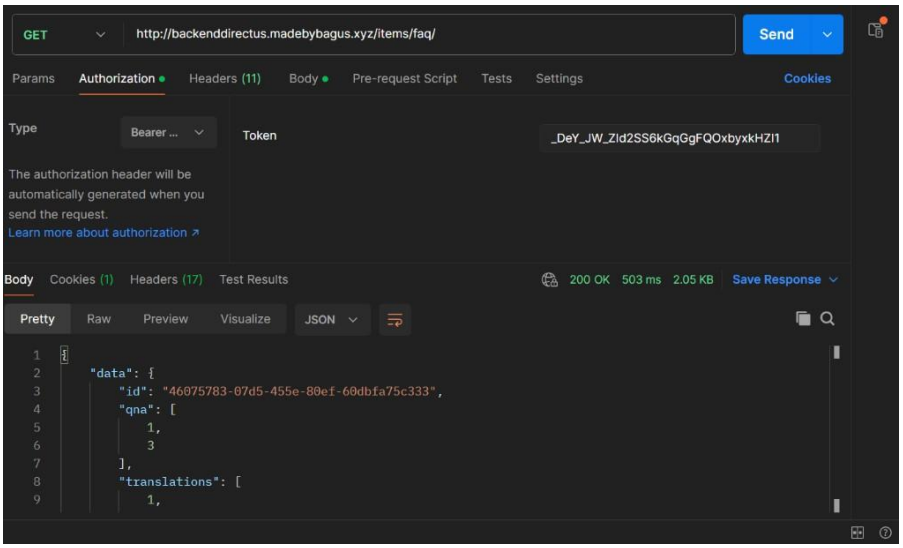
2. Frequently Asked Questions (FAQ)

Koleksi ini memuat daftar pertanyaan yang sering diajukan oleh pengunjung *website* terkait dengan Keraton Yogyakarta, beserta dengan jawabannya. Tujuannya adalah untuk memberikan kemudahan bagi pengunjung dalam menemukan informasi yang dibutuhkan tanpa harus menghubungi pihak Keraton Yogyakarta secara langsung. Dengan adanya koleksi ini, pengunjung dapat memperoleh pemahaman yang lebih baik mengenai berbagai aspek Keraton Yogyakarta, seperti sejarah, tata cara kunjungan, serta kegiatan budaya yang diselenggarakan.



Gambar 3.32 Tampilan Koleksi *FAQ*

Koleksi *FAQ* memiliki relasi dengan satu koleksi lain yang bernama koleksi *Qna*, dimana koleksi FAQ berperan sebagai penghubung antara koleksi *FAQ* sendiri dengan koleksi *Qna*. Relasi antar dua koleksi ini bertipe *one to many* dimana satu *FAQ* bisa memiliki banyak pertanyaan beserta dengan jawabannya.

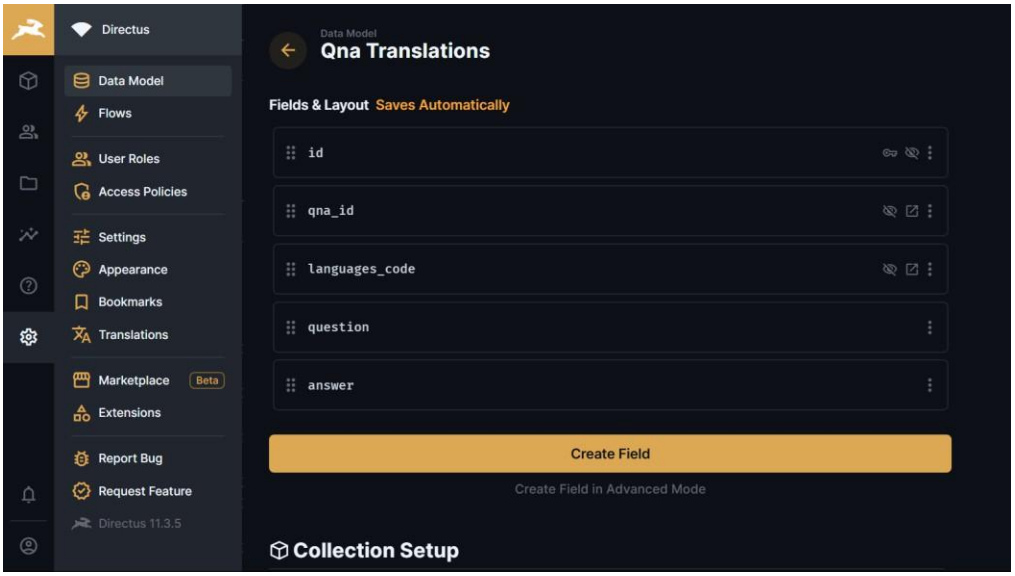


Gambar 3. 33 Pengujian API Koleksi *FAQ*

Gambar ini menampilkan hasil pengujian API GET pada koleksi *FAQ*. Respon yang dihasilkan menunjukkan bahwa daftar pertanyaan dan jawaban dapat diakses dengan benar melalui *endpoint* yang otomatis yang diberikan oleh Directus. Pengujian ini memastikan bahwa koleksi *FAQ* siap digunakan sebagai sumber informasi dalam aplikasi.

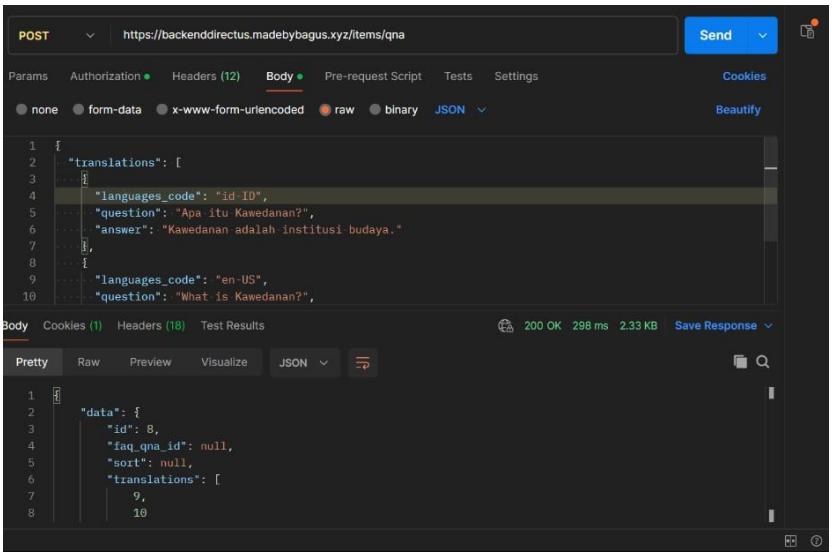
3. Question and Answer (QnA)

Koleksi *QnA* digunakan untuk menyimpan data pertanyaan dan jawaban yang akan ditampilkan pada halaman *FAQ*. Setiap item di dalam koleksi ini berisi pertanyaan, beserta dengan jawaban. Dengan demikian, *QnA* berfungsi sebagai sumber data terpusat untuk kebutuhan informasi tanya jawab dalam sistem.



Gambar 3.34 Tampilan Koleksi *QnA*

Pada koleksi *QnA* (*Questions and Answers*), admin memiliki peran dalam mengelola berbagai pertanyaan dan jawaban yang berkaitan dengan informasi yang akan ditampilkan pada *website*. Koleksi ini berfungsi sebagai wadah penyimpanan data berbentuk tanya jawab yang biasanya digunakan pada halaman *FAQ*, di mana pengguna dapat menemukan berbagai informasi umum atau penjelasan mengenai layanan, produk, maupun sistem yang disediakan oleh perusahaan. Melalui koleksi ini, pengelolaan konten tanya jawab menjadi lebih sistematis, terstruktur, dan mudah diperbarui kapan pun diperlukan.

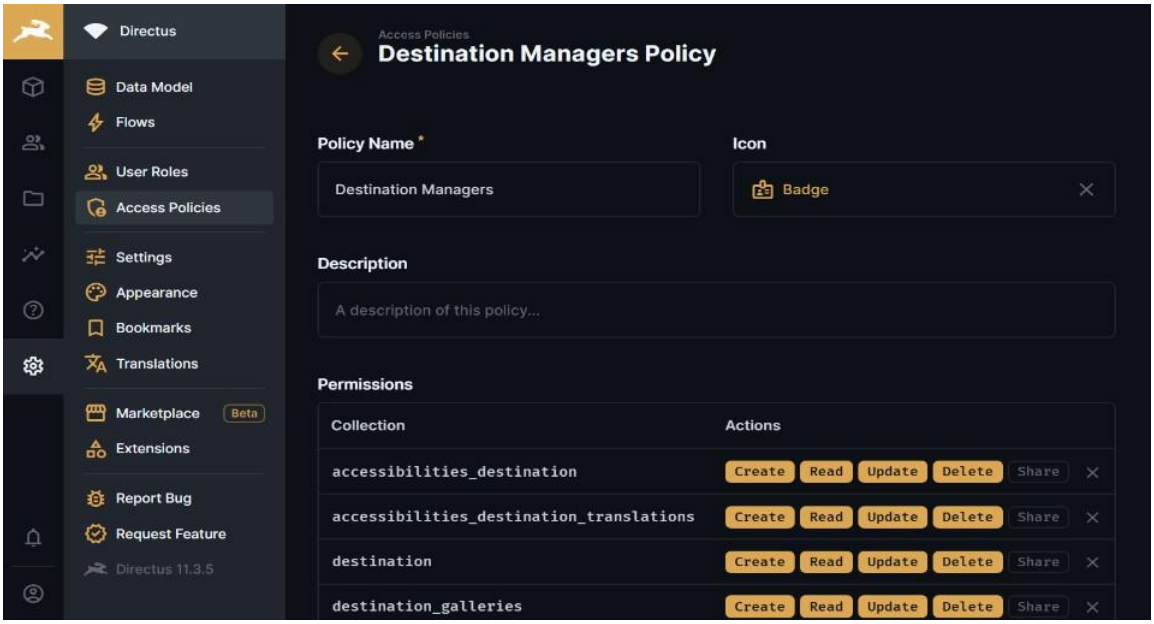


Gambar 3. 35 Pengujian API Koleksi *QnA* (POST)

Pengujian API metode POST pada koleksi *QnA* berhasil dilakukan dengan respons 200 OK, yang menandakan bahwa data pertanyaan dan jawaban dalam dua bahasa berhasil ditambahkan ke sistem. Nilai yang dikirim melalui *field translations* diterima sepenuhnya oleh Directus dan langsung tersimpan sesuai struktur multilingual yang digunakan. Hasil ini menunjukkan bahwa API pada koleksi *QnA* berfungsi dengan baik dalam memproses dan menyimpan data berbasis translasi.

4. Access Policies

Pengaturan *access policies* bertujuan untuk mengelola dan membatasi hak akses setiap akun pengguna sesuai dengan peran atau tanggung jawabnya di dalam sistem. Fitur ini berfungsi untuk menjaga keamanan data serta memastikan bahwa setiap pengguna hanya dapat mengakses bagian tertentu dari sistem yang relevan dengan tugasnya.



Gambar 3.36 Tampilan *Access Policies*

Sebagai contoh, untuk peran *destination managers*, akun dengan peran ini hanya diberikan izin untuk mengakses koleksi yang berhubungan dengan destinasi, seperti data lokasi, deskripsi, dan gambar pendukung destinasi. Mereka tidak memiliki akses untuk melihat, mengubah, atau menghapus data dari koleksi lain yang tidak berkaitan. Dengan demikian, pengaturan *access policies* membantu mencegah terjadinya kesalahan akses data, melindungi informasi sensitif, dan memastikan bahwa sistem berjalan secara efisien dan aman sesuai dengan prinsip pembagian peran dalam manajemen data di Directus.

3.5 Evaluasi dan Penutupan Proyek

Fase ini merupakan tahap terakhir dari seluruh rangkaian proses yang telah dilaksanakan dalam pengembangan *website*, sekaligus menjadi tahap penutup yang menandai selesainya seluruh kegiatan inti dalam proyek tersebut. Evaluasi ini menjadi langkah krusial untuk memastikan bahwa sistem yang dikembangkan benar-benar mampu berjalan dengan baik, stabil, dan dapat digunakan sesuai ekspektasi baik oleh pihak perusahaan maupun pengguna akhir. Tahap ini tidak hanya menekankan pada aspek teknis semata, tetapi juga mencakup kegiatan konsultasi dan refleksi bersama pihak pembimbing, supervisor, atau tim teknis dari perusahaan. Dalam sesi konsultasi ini, penulis memaparkan hasil capaian yang telah diperoleh, termasuk kemajuan proyek, tantangan yang dihadapi selama proses pengembangan, serta solusi yang telah diterapkan untuk mengatasi kendala tersebut. Diskusi ini bersifat terbuka dan konstruktif, di mana pembimbing maupun pihak perusahaan memberikan umpan balik dan

rekomendasi yang bertujuan untuk meningkatkan kualitas sistem dan menyempurnakan hasil kerja yang telah dilakukan.

Selain berfungsi sebagai bentuk pertanggungjawaban terhadap hasil magang dan proyek yang dikerjakan, fase ini juga menjadi sarana untuk melakukan refleksi diri terhadap seluruh proses yang telah dijalani. Hasil dari seluruh kegiatan evaluasi dan konsultasi ini kemudian dijadikan dasar dalam penyusunan laporan akhir magang maupun laporan penelitian, yang memuat analisis menyeluruh mengenai pencapaian tujuan, kendala teknis maupun non-teknis, serta rekomendasi untuk pengembangan sistem selanjutnya. Dengan demikian, fase ini tidak hanya menjadi akhir dari proses pengembangan *website*, tetapi juga menjadi tahap pembelajaran penting yang memberikan wawasan baru, pengalaman, dan pemahaman mendalam mengenai penerapan ilmu di dunia kerja nyata.

BAB IV
REFLEKSI PELAKSANAAN MAGANG

4.1 Relevansi Akademik

Pengembangan basis data *website* Radyakartiyasa merupakan *website* yang dibuat dengan tujuan untuk mempromosikan destinasi di wilayah Keraton, serta *event* kebudayaan yang diselenggarakan oleh pihak Keraton. *Website* Radyakartiyasa menyediakan berbagai fitur untuk memfasilitasi layanan dan mempermudah para pengunjung yang tertarik untuk mengunjungi atau menghadiri *event* yang ada. Perancangan basis data merupakan salah satu tahapan yang diperlukan dalam proses pengembangan *website* ini.

Dalam pengerjaan pengembangan basis data *website* ini memiliki relevansi akademik yang kuat karena berkaitan langsung dengan teori yang dibahas dalam tinjauan pustaka, terutama mengenai konsep *Headless CMS* yang mendasari penggunaan Directus. Teori-teori tersebut memberikan pemahaman mengenai pengelolaan data yang terstruktur dan fleksibel, sehingga sesuai dengan kebutuhan pembangunan basis data *website* Radyakartiyasa. Dengan demikian, praktik yang dilakukan selama magang menjadi penerapan nyata dari konsep yang telah dipelajari dalam kajian literatur.

Pengalaman magang ini juga memiliki relevansi dengan materi yang telah dipelajari selama perkuliahan, terutama pada mata kuliah Pengembangan Aplikasi Berbasis Website (PABW) dan Manajemen Proyek Teknologi Informasi (MPTI). Konsep yang dipelajari menggunakan phpMyAdmin di PABW, seperti pengelolaan data, struktur konten, dapat diterapkan secara langsung melalui penggunaan Directus sebagai *backend*. Sementara itu, prinsip perencanaan, pembagian tugas, dan alur kerja proyek yang diajarkan dalam MPTI sangat membantu penulis dalam memahami proses kerja tim di lingkungan perusahaan. Melalui magang ini, penulis dapat melihat bagaimana teori yang dipelajari di perkuliahan digunakan dalam penyusunan dan pengembangan sistem.

Selain itu, penggunaan Docker untuk menjalankan Directus memberikan kesempatan bagi penulis untuk mengaplikasikan kembali konsep dasar yang pernah dipelajari pada mata kuliah Sistem Jaringan Komputer (SJK). Penulis dapat memahami bagaimana layanan saling berkomunikasi, serta bagaimana *port* digunakan. Secara keseluruhan, magang ini menjadi pengalaman yang memperkuat hubungan antara teori akademik dengan praktik yang diterapkan dalam dunia kerja.

Selama pelaksanaan magang, penulis bertanggung jawab untuk membuat basis data yang dibutuhkan dalam pengembangan *website* Radyakartiyasa. Pembuatan basis data dilakukan berdasarkan rancangan desain UI/UX yang sebelumnya telah disusun oleh tim desain. Selama pengerjaan, penulis juga aktif berdiskusi dengan pihak supervisor terkait dengan penyesuaian kebutuhan yang diinginkan perusahaan.

.

4.2 Pembelajaran Magang

4.2.1 Manfaat

Manfaat yang dirasakan penulis selama melaksanakan magang di PT Lumintu Logic Indonesia adalah penulis mendapatkan kesempatan untuk mempelajari dan berpartisipasi dalam proses pengembangan basis data *website* Radyakartiyasa. Penulis dapat mengimplementasikan ilmu yang belum pernah penulis dapatkan di tempat lain sebelumnya.

Kegiatan magang memberikan kontribusi signifikan terhadap peningkatan kompetensi profesional penulis. Penulis menjadi terampil dalam bekerja secara mandiri dibawah batas waktu yang ditentukan, serta mematuhi standar yang berlaku di lingkungan kerja. Kemampuan dalam menentukan prioritas tugas dan melakukan pemecahan masalah (*troubleshooting*) terhadap isu teknis juga meningkat secara signifikan.

Pelaksanaan magang ini juga memfasilitasi pengembangan kemampuan komunikasi profesional, yang dipraktikkan melalui sesi diskusi dan presentasi hasil kerja dengan supervisor. Kemampuan untuk menerima dan mengimplementasikan masukan serta memastikan bahwa hasil pekerjaan dapat memenuhi tujuan tim pengembangan. Secara keseluruhan, program ini berperan sebagai jembatan antara wawasan akademik dan dunia industri, yang memperkuat kesiapan penulis dalam menghadapi karir profesional.

4.2.2 Kendala, Hambatan, dan Tantangan

Selama melaksanakan kegiatan magang sebagai *backend developer* di PT Lumintu Logic Indonesia, penulis menghadapi berbagai tantangan dan hambatan teknis maupun non-teknis yang muncul selama proses pengembangan basis data *website* Radyakartiyasa. Salah satu tantangan terbesar yang dirasakan adalah keterbatasan wawasan dan pengalaman praktis dalam penggunaan teknologi yang diterapkan oleh perusahaan, khususnya dalam mengoperasikan *Headless CMS* Directus, yang merupakan *platform* utama dalam pengelolaan data dan integrasi sistem backend. Sebelumnya, penulis belum memiliki pengalaman mendalam dalam bidang *backend development*, karena sebagian besar pengalaman akademik penulis lebih berfokus

pada sisi desain. Oleh karena itu, penulis memerlukan waktu untuk beradaptasi dengan lingkungan kerja baru, alur pengembangan sistem, serta teknologi yang digunakan.

Selain kendala tersebut, penulis juga menghadapi sejumlah kesulitan teknis selama proses penyusunan struktur data di Directus. Beberapa konfigurasi awal yang dibuat ternyata belum sesuai dengan standar yang diterapkan perusahaan, seperti penggunaan *interface* yang kurang tepat, penentuan tipe koleksi yang belum sesuai, serta relasi antar koleksi yang perlu diperbaiki agar data dapat saling terhubung dengan benar. Pada beberapa bagian, terdapat *field* yang tidak diperlukan sehingga harus dihapus atau disesuaikan kembali, sementara beberapa koleksi lainnya perlu diubah agar lebih rapi dan konsisten dengan kebutuhan sistem. Proses revisi ini menuntut ketelitian serta pemahaman yang lebih dalam terhadap alur data yang digunakan oleh tim *frontend*, sehingga menjadi tantangan tambahan bagi penulis dalam menyelesaikan tugas dengan baik.

Keterbatasan tersebut menjadi kendala yang paling menonjol, terutama pada tahap awal pelaksanaan magang, ketika penulis harus mempelajari konsep-konsep baru seperti pengelolaan API, manajemen data dalam Directus, konfigurasi *database*, serta pengaturan koneksi antara *backend* dan *frontend*. Proses adaptasi ini menuntut penulis untuk memiliki kemampuan belajar mandiri yang tinggi dan mampu memahami struktur kerja sistem dengan cepat. Selain itu, penulis juga menghadapi tantangan dalam memahami alur kerja tim pengembang yang sudah memiliki ritme dan standar kerja tersendiri, sehingga diperlukan penyesuaian terhadap cara berkomunikasi, pembagian tugas, serta manajemen waktu agar dapat berkontribusi secara efektif dalam tim.

Untuk mengatasi kendala tersebut, penulis berupaya aktif melakukan berbagai langkah strategis. Salah satunya dengan berkonsultasi langsung kepada supervisor ketika menghadapi kesulitan teknis atau ketidakpahaman dalam menjalankan instruksi. Supervisor memberikan arahan yang sangat membantu dalam memahami konsep-konsep dasar *backend* serta cara menerapkannya pada proyek yang sedang dikerjakan. Selain itu, penulis juga secara mandiri mempelajari dokumentasi resmi Directus yang tersedia secara *online*, guna memahami fitur, fungsi, serta konfigurasi yang dapat dioptimalkan dalam proyek.

BAB V
PENUTUP

5.1 Kesimpulan

Penelitian ini dilakukan untuk merancang dan mengembangkan basis data pada *website* Radyakartiyasa sebagai media digital yang mendukung promosi dan penyebaran informasi kebudayaan. Proses pengembangan dilakukan dengan memanfaatkan *Headless Content Management System* (CMS) Directus agar pengelolaan konten dapat dilakukan secara lebih efisien, terstruktur, dan mudah dikembangkan. Sistem yang dibangun diharapkan mampu menjadi fondasi yang kuat dalam pengelolaan data serta mendukung keberlanjutan pengembangan *website* di masa mendatang.

Proyek ini dikembangkan dengan menggunakan metode *Agile* yang diterapkan melalui kerangka kerja *Scrum*. Pendekatan ini menekankan kolaborasi tim, pelaksanaan iterasi secara bertahap, serta kemampuan beradaptasi terhadap perubahan kebutuhan pengguna. Setiap tahap dalam *Scrum* seperti *sprint planning, daily scrum,* dan *sprint review* membantu tim untuk memantau perkembangan proyek, mengevaluasi hasil sementara, dan melakukan perbaikan berkelanjutan agar hasil akhir sesuai dengan target yang telah ditetapkan.

Hasil implementasi menunjukkan bahwa penggunaan Directus mampu mewujudkan *platform* digital yang informatif sekaligus dinamis bagi Kawedanan Radyakartiyasa. Koleksi data yang telah dirancang seperti *navigation, destination index, event index, hero banner,* dan *FAQ* membentuk struktur basis data yang mendukung pengelolaan konten secara sistematis. Dengan sistem yang fleksibel dan mudah dioperasikan, proses pembaruan informasi dapat dilakukan lebih cepat.

Berdasarkan hasil evaluasi, seluruh tujuan penelitian yang telah ditetapkan dapat dicapai dengan baik. Struktur basis data yang dibuat telah sesuai dengan kebutuhan pengelolaan konten, penggunaan antarmuka grafis Directus terbukti mempermudah proses pengembangan tanpa memerlukan penulisan *query* SQL secara manual, dan hasil pengujian API menunjukkan bahwa sistem yang dibangun telah berfungsi dengan baik untuk memenuhi kebutuhan *frontend*. Dengan demikian, penerapan *Headless CMS* Directus menjadi solusi yang tepat dan efisien dalam mendukung pengelolaan konten digital.

Dari sisi akademik, penelitian ini memberikan temuan penting berupa penerapan teknologi *Headless CMS* yang masih jarang dibahas dalam pembelajaran di tingkat universitas. Hal tersebut memperkuat relevansi antara teori yang dipelajari di perkuliahan dengan praktik

nyata di dunia industri. Sementara dari sisi non-akademik, pengalaman dalam beradaptasi dengan lingkungan kerja profesional, bekerja sama lintas divisi, serta membangun komunikasi yang efektif menjadi pengalaman yang paling berkesan dan memberikan nilai tambahan bagi penulis selama menjalani program magang.

5.2 Saran

Berdasarkan hasil temuan dan analisis yang didokumentasikan, terdapat beberapa saran yang dapat dipertimbangkan untuk penelitian selanjutnya yaitu:

a. Untuk proyek itu sendiri, langkah utama selanjutnya yang disarankan adalah melakukan integrasi penuh antara API yang telah dikembangkan dengan antarmuka *frontend*, serta mengimplementasikan fungsionalitas multibahasa yang basis datanya telah disiapkan. Disarankan pula untuk mengeksplorasi fitur-fitur lanjutan dari *platform Headless CMS* yang digunakan, seperti otomatisasi alur kerja dan manajemen hak akses pengguna yang lebih terperinci, serta menyusun panduan pengguna untuk memudahkan tim pengelola konten.
b. Dari sisi *backend*, untuk *website* disarankan untuk menambahkan fitur *sorting* dan *filtering* pada data *destination* dan *event index* agar pengolahan serta penyajian data menjadi lebih optimal. Fitur *sorting* dapat digunakan untuk mengurutkan destinasi atau *event* berdasarkan jumlah kunjungan atau jumlah pembelian tiket terbanyak, sehingga *backend* dapat menyediakan data yang lebih terstruktur untuk kebutuhan analisis. Selain itu, penambahan *filter* seperti kategori, harga, dan lokasi akan memudahkan proses pengambilan data spesifik sesuai kebutuhan sistem. Dengan adanya fitur ini, *backend* dapat mendukung perusahaan dalam melihat tren destinasi atau event *yang* paling diminati dan mempermudah proses pengembangan fitur lanjutan.
c. Bagi pihak perusahaan, penerapan metode kerja berbasis *Agile* dan penggunaan teknologi *Headless CMS* dapat dijadikan acuan dalam mengembangkan proyek serupa di masa depan. Pelatihan tambahan mengenai manajemen konten digital juga dapat diberikan kepada peserta magang agar proses adaptasi terhadap teknologi baru menjadi lebih cepat.
d. Terakhir, bagi pihak akademik, hasil penelitian ini dapat menjadi bahan pertimbangan dalam penyusunan kurikulum yang lebih relevan dengan perkembangan teknologi industri

DAFTAR PUSTAKA

Adhitya, M. R., & Suartana, I. M. (2025). Pengaruh Implementasi Headless Content Management System Terhadap Performa dan Kinerja E-Learning. *Journal of Informatics and Computer Science*, *07*.

Afriyanto, T., & Noviandi. (2023). IMPLEMENTASI SCRUM DALAM PENGEMBANGAN SISTEM INFORMASI PENCATATAN LAPORAN TUBERCULOSIS. *JIK: Jurnal Ilmu Komputer*, *08*(01), 2527–9653. https://doi.org/10.47007/komp.v%vi%i.6582

Amarta, A. A. F., & Anugrah, I. G. (2021). Implementasi Agile Scrum Dengan Menggunakan Trello Sebagai Manajemen Proyek Di PT Andromedia. *Jurnal Nasional Komputasi Dan Teknologi Informasi*, *4*(6).

Ariesta, A., Novita Dewi, Y., Ayu Sariasih, F., & Wahyuhening Fibriany, F. (2021). *PENERAPAN METODE AGILE DALAM PENGEMBANGAN APPLICATION PROGRAMMING INTERFACE SYSTEM PADA PT XYZ*.

Arifiany, I., & Kusuma, G. P. (2025). LEVERAGING HEADLESS CONTENT MANAGEMENT SYSTEM AS A SERVICE IN A SERVICE-BASED ARCHITECTURE: ENHANCING USER EXPERIENCE AND OVERCOMING RESOURCE LIMITATIONS FOR START-UPS. *Journal of Theoretical and Applied Information Technology*, *15*(7). www.jatit.org

Arnomo, S. A., & Kurniawan, D. E. (2024). *Metode Agile Scrum Dalam Pengembangan Sistem Pengendali Stok Barang*. http://journal.aptikomkepri.org/index.php/JDDAT

Arsa, D., Weni, I., & Dafian, M. F. (2024). METODE SCRUM PADA RANCANG BANGUN SISTEM INFORMASI PELAPORAN HARIAN PEGAWAI BPS PROVINSI JAMBI. *JSiI (Jurnal Sistem Informasi)*, *11*(1), 40–46. https://doi.org/10.30656/jsii.v11i1.8212

Ayurira, C. L. A., & Fajri, I. N. (2024). IMPLEMENTASI METODE SCRUM DALAM PENGEMBANGAN WEBSITE E-COMMERCE PADA TWINS PETSHOP. *Jurnal Innovation and Future Technology (IFTECH)*, *6*(2).

Berndtsson, A., & Fagerberg, E. (2025). *Integrating Social Media Functionality into a Headless CMS: A Unified Approach to Content Management and Publishing Workflows*.

Chasanah, N. I., Manik, J. S. R., Meliala, R. J., Pasya, T. M., Wicaksono, A., & Mindara, G. P. (2025). Pengembangan Website Nyuci sebagai Sistem Layanan Laundry Responsif Menggunakan Metode Scrum. *Jurnal Teknologi Dan Manajemen Industri Terapan (JTMIT)*, *4*(2), 367–377.

COMPANY PROFILE PT LUMINTU LOGIC. (2024).

Dzaky, F. A., & Kurniawan, D. (2023). Implementasi Metode Agile Framework Scrum dalam Pengembangan Sistem Informasi Manajemen Aset Terpadu Universitas Diponegoro Modul Inventarisasi. *Jurnal Masyarakat Informatika*, *14*(1), 2777–0648.

Eassa, A. M. (2025). *International Journal of Computers and Information (IJCI) Optimizing Web Application Development: A Proposed Architecture Integrating Headless CMS and Serverless Computing* (Vol. 12, Issue 1). https://ijci.journals.ekb.eg/

Emadamerho-Atori, N. (2023, December 1). *Headless CMS: A Complete Introduction*. Prismic.

Gutama, R., & Dirgahayu, T. (2021). *Implementasi Scrum Pada Manajemen Proyek Pengembangan Aplikasi Sistem Monitoring dan Evaluasi Pembangunan (SMEP)*.

Hartadi, T. B., & Rahayu, N. W. (2025). PENGEMBANGAN BASIS DATA MULTILINGUAL MENGGUNAKAN HEADLESS CMS DIRECTUS PADA WEBSITE RADYA KARTIYASA. *Sains Dan Teknologi*, *12*(3), 2025–1415. https://doi.org/10.47668/edusaintek.v12i3.1779

Hietala, V. (2025). *USING HEADLESS SOLUTION FOR E-COMMERCE WEBSITE IMPLEMENTATION*.

Maulana, A., Tiawan, Wahyudi, Purba, A. B., Indra Grahana, C., Hilman, A., Setiyani, L., Najib Mutawally, A., & Fatkhu Rohman, A. (2024). REKAYASA SISTEM INFORMASI BOOKING SPORT DENGAN AGILE SCRUM DAN UML. *Jurnal Informatika Teknologi Dan Sains (JINTEKS)*, *6*(2), 247–251. https://doi.org/http://dx.doi.org/10.51401/jinteks.v6i2.4128

Miftahuljannah, V., & Suharso, A. (2023). PENGIMPLEMENTASIAN BERBAGAI WEB BERDASARKAN KEBUTUHAN PENGGUNA DENGAN MENGGUNAKAN METODE SYSTEMATIC LITERATURE REVIEW. *INFOTECH Journal*, *9*(2), 401–405. https://doi.org/10.31949/infotech.v9i2.6341

Mohammed, B. (2024, March 15). *Seamless Content Management with Directus*. Vpsserver.

Nana, J., Lamabelawa, M. I. J., & Payong, Y. (2022). *SISTEM INFORMASI PROMOSI PARIWISATA BERBASIS WEBSITE DI KABUPATEN MALAKA*. *13*(1).

Pritandari, Y. R., Ratnaningtyas, Y. A., & Putra, T. P. (2024). *Roles And Functions Of Kawedanan Radya Kartiyasa In Managing Kraton Yogyakarta Museum*. *2*(1). https://doi.org/https://doi.org/10.24821/tumata.v2i1.13254

Renata, A., Zahrani, R., & Melandri, B. (2022). ANALISIS EFISIENSI TAHAPAN PENGGUNAAN METODE MANAJEMEN PROYEK STI (AGILE SCRUM DAN

WATERFALL) PADA SUATU PERUSAHAAN ATAU ORGANISASI. *SINTESIA: Jurnal Sistem Dan Teknologi Informasi Indonesia*, *1*(2). https://teknosi.fti.unand.ac.id/index.php/tek

Sari, A. P., & Suhendi. (2020). RANCANG BANGUN SISTEM INFORMASI PENGELOLAAN TALENT FILM BERBASIS APLIKASI WEB. *Jurnal Informatika Terpadu*, *6*(1), 29–37. https://journal.nurulfikri.ac.id/index.php/JIT

Schwaber, K., & Sutherland, J. (2020). *Panduan Scrum Panduan Definitif untuk Scrum: Aturan Permainan*.

Singh, T. (2023). Exploring WebAssembly, Progressive Web Apps, and Headless CMS for Cutting-Edge Web Development. *Proceedings - 2023 3rd International Conference on Innovative Sustainable Computational Technologies, CISCT 2023*. https://doi.org/10.1109/CISCT57197.2023.10351343

Sobri, N. A. N., Abas, M. A. H., Yassin, A. I. M., Ali, M. S. A. M., Tahir, N. M., Zabidi, A., & Rizman, Z. I. (2022). *Comparison between Headless CMS and Backend-as-a-Service Products for E-Suripreneur Backend*. *71*(3). http://philstat.org.ph

Surentu, Y. Z., Warouw, D. M. D., & Rembang, M. (2020). *PENTINGNYA WEBSITE SEBAGAI MEDIA INFORMASI DESTINASI WISATA DI DINAS KEBUDAYAAN DAN PARIWISATA KABUPATEN MINAHASA*. *2*(4).

Tandra, R., Siallagan, D. Y., & Prestoroika, E. (2025). *INTEROPERABILITAS SISTEM INFORMASI PROFIL JURUSAN BERBASISKAN HEADLESS CONTENT MANAGEMENT SYSTEM WEBSITE JURUSAN*. *10*.

Usman, M. F., Wibawa, A. P., Zaeni, I. A. E., & Budiarti, A. (2021). Implementasi SCRUM Pada Pengembangan Aplikasi Sistem ADI STETSA SMAN 4 Malang: Bidang TATA Usaha. *Jurnal Inovasi Teknologi Dan Edukasi Teknik*, *1*(7), 498–507. https://doi.org/10.17977/um068v1i72021p498-507

Yadav, D. (2021). Enhancing E-commerce Footwear Business: A Headless Implementation Approach for Website and App Development. *International Journal of Innovative Research and Creative Technology (Www.Ijirct.Org)*, *7*(1). www.ijirct.org

LAMPIRAN

MATERI TRAINING



JADWAL KELAS

LUMINTU

Hari, Tanggal	Waktu	Pertemuan	Topik	Materi
Kamis, 19 Sept 2024	09:00 - 12:00 WIB	Orientasi 1	Non-Technical	Induction
Kamis, 19 Sept 2024	13:00 - 14:00 WIB	Orientasi 2	Technical	Project Presentation
Senin, 23 Sept 2024	10:00 - 12:00 WIB	Pertemuan 1	Technical	Key Definition: SQL Database vs Directus
Senin, 23 Sept 2024	13:00 - 15:00 WIB	Pertemuan 2	Technical	Installing Directus: npm method
Kamis, 26 Sept 2024	10:00 - 12:00 WIB	Pertemuan 3	Technical	Creating collections and relationships
Kamis, 26 Sept 2024	13:00 - 15:00 WIB	Pertemuan 4	Technical	Accessing API: Basics of collections APIs
Senin, 30 Sept 2024	10:00 - 12:00 WIB	Pertemuan 5	Technical	Accessing API: Global paramaters using REST
Senin, 30 Sept 2024	13:00 - 15:00 WIB	Pertemuan 6	Technical	Accessing API: User authentication flow and Access Control
kamis. 03 Okt 2024	10:00 - 12:00 WIB	Pertemuan 7	Non-Technical	English for IT Placement Test
kamis. 03 Okt 2024	13:00 - 15:00 WIB	Pertemuan 8	Technical	Directus Flows: 3rd party system integration example with Midtrans payment gateway
Kamis, 10 Okt 2024	10:00 - 12:00 WIB	Pertemuan 9	Technical	Advanced configuration: Email SMTP integration and Redis Integration
Kamis, 10 Okt 2024	13:00 - 15:00	Pertemuan 10	Technical	Websocket APIs
Senin, 14 Okt 2024	10:00 - 12:00	Pertemuan 11	Non-Technical	English for IT
Senin, 14 Okt 2024	13:00 - 15:00	Pertemuan 12	Technical	Starting AstroJS Project: Pages and Components

LUMINTU

Hari, Tanggal	Waktu	Pertemuan	Topik	Materi
Kamis, 17 Okt 2024	10:00 - 12:00 WIB	Pertemuan 13	Technical	Static Site Generation: Frontmatter API fetching from Directus
Kamis, 17 Okt 2024	13:00 - 15:00 WIB	Pertemuan 14	Technical	Static Site Generation: Permalink rendering
Senin, 21 Okt 2024	10:00 - 12:00 WIB	Pertemuan 15	Non-Technical	English for IT
Senin, 21 Okt 2024	13:00 - 15:00 WIB	Pertemuan 16	Technical	Static Site Generation: Client-side rendering and form submission
Senin, 28 Okt 2024	10:00 - 12:00 WIB	Pertemuan 17	Technical	Static Site Generation: Jamstack static assets procurement
Senin, 28 Okt 2024	13:00 - 15:00 WIB	Pertemuan 18	Technical	Server Side Rendering: Server entrypoint and Middleware
Kamis, 31 Okt 2024	10:00 - 12:00 WIB	Pertemuan 19	Non-Technical	English for IT
Kamis, 31 Okt 2024	13:00 - 15:00 WIB	Pertemuan 20	Technical	Server Side Rendering: Page's frontmatter and dynamic querying
Senin, 04 Nov 2024	10:00 - 12:00 WIB	Pertemuan 21	Technical	Server Side Rendering: Creating custom API endpoints
Senin, 04 Nov 2024	13:00 - 15:00 WIB	Pertemuan 22	Technical	Server Side Rendering: User authentication flow and cookies management II
Kamis, 07 Nov 2024	10:00 - 12:00 WIB	Pertemuan 23	Non-Technical	English for IT
Kamis, 07 Nov 2024	13:00 - 15:00 WIB	Pertemuan 24	Technical	CI/CD flow: Lumintu Logic's secret recipe
Kamis, 14 Nov 2024	10:00 - 12:00 WIB	Pertemuan 25	Technical	Linux: Installation and basic commands

LUMINTU

Hari, Tanggal	Waktu	Pertemuan	Topik	Materi
Kamis, 14 Nov 2024	13:00 - 15:00 WIB	Pertemuan 26	Technical	Docker and Docker-Compose: Creating a container using example from Code-Server
Senin, 18 Nov 2024	10:00 - 12:00 WIB	Pertemuan 27	Non-Technical	English for IT
Senin, 18 Nov 2024	13:00 - 15:00 WIB	Pertemuan 28	Technical	Docker and Docker-Compose: Install Directus
Kamis, 21 Nov 2024	10:00 - 12:00 WIB	Pertemuan 29	Technical	Cloudflare: connecting a domain, creating a tunnel, and creating public hostnames.
Kamis, 21 Nov 2024	13:00 - 15:00 WIB	Pertemuan 30	Non-Technical	Law in IT (IP Rights)
Senin, 25 Nov 2024	10:00 - 12:00 WIB	Pertemuan 31	Non-Technical	English for IT
Senin, 25 Nov 2024	13:00 - 15:00 WIB	Pertemuan 32	Technical	Git: Basic Git commands and managing repositories for CI/CD flow
Kamis, 28 Nov 2024	10:00 - 12:00 WIB	Pertemuan 33	Technical	Cloudflare: completing CI/CD for Astro Project
Kamis, 28 Nov 2024	13:00 - 15:00 WIB	Pertemuan 34	Technical	Cloudflare: Project staging and deployment
Kamis, 05 Des 2024	10:00 - 12:00 WIB	Pertemuan 35	Non-Technical	English for IT
Kamis, 05 Des 2024	13:00 - 15:00 WIB	Pertemuan 36	Technical	Cloudflare: Using cloudflare featured services using Turnstile as example
Senin, 09 Des 2024	10:00 - 12:00 WIB	Pertemuan 37	Non-Technical	Public Speaking & Presentation Workshop (final presentation briefing)
Senin, 09 Des 2024	13:00 - 15:00 WIB	Pertemuan 38	Technical	System Integration: 3rd party webhook using example from Midtrans

LUMINTU

Hari, Tanggal	Waktu	Pertemuan	Topik	Materi
Kamis, 12 Des 2024	10:00 - 12:00 WIB	Pertemuan 39	Non-Technical	English for IT
Kamis, 12 Des 2024	13:00 - 15:00 WIB	Pertemuan 40	Technical	Documenting Directus' API using Swagger
Senin, 16 Des 2024	10:00 - 12:00 WIB	Pertemuan 41	Non-Technical	Law in IT (Software Development Contract)
Senin, 16 Des 2024	13:00 - 15:00 WIB	Pertemuan 42	Non-Technical	Career Preparation (Application letter, CV & interview workshop)
Kamis, 19 Des 2024	10:00 - 12:00 WIB	Pertemuan 43	Non-Technical	Career Preparation (CV & interview simulation & feedback)
Kamis, 19 Des 2024	13:00 - 14:00 WIB	Pertemuan 44	Non-Technical	Students presentation & Enter Interview
Kamis, 19 Des 2024	14:00 - 15:00 WIB	Pertemuan 45	Non-Technical	Students presentation & Enter Interview

PELAKSANAAN TRAINING MATERI ASTRO JS



PRESENTASI SAAT EXIT INTERVIEW



SERTIFIKAT TRAINING



Lembaga Pelatihan Kerja
Vocational Training Institution
PT. LUMINTU LOGIC INDONESIA
Jl. Tegal Arum No.08 Ringroad Barat, Gamping Kidul, Ambarketawang, Kec. Gamping, Kab. Sleman, Daerah Istimewa Yogyakarta 55294
NIB: 0209220056565

SERTIFIKAT PELATIHAN KERJA
VOCATIONAL TRAINING CERTIFICATE

No. 01/LPK/XII/2024

Lembaga Pelatihan Kerja **Lumintu Logic Indonesia**
menyatakan bahwa:
*The Vocational Training Institution **Lumintu Logic Indonesia** certifies that:*

MUTIARA IRDINA

Lahir di Balikpapan tanggal 08 April 2003 telah menyelesaikan Pelatihan Kerja bidang Teknologi Informasi dan Komunikasi dengan predikat:
Born in Balikpapan on 08 April 2003 has successfully completed the Vocational Training Program in the field of Information and Communication Technology with the grade:

Baik
Good

Sertifikat ini ditetapkan di **Yogyakarta**, berlaku tanggal **23 Desember 2024** sampai **23 Desember 2026**
*This certificate is issued in **Yogyakarta** and is valid from **23 December 2024** to **23 December 2026***

Yogyakarta, 23 Desember 2024

Happy Hartana
Direktur PT. Lumintu Logic Indonesia
Director of PT. Lumintu Logic Indonesia

REVIEW MILESTONE

